

Bruna Rossetto Delazeri

**Reconhecimento de expressões faciais- Uma  
abordagem com aprendizado não supervisionado  
de *pool* de representações diversas**

CURITIBA

2024

Bruna Rossetto Delazeri

**Reconhecimento de expressões faciais- Uma abordagem  
com aprendizado não supervisionado de *pool* de  
representações diversas**

Tese apresentada ao Programa de Pós  
Graduação em Informática da Pontifícia  
Universidade Católica do Paraná como  
requisito parcial para obtenção do título de  
Doutor em Informática.

Área de Concentração: Ciência da Computação

Orientador: Prof. Dr. Alceu de Souza  
Britto Junior

Co-orientador: Prof. Dr. Alessandro Lameiras  
Koerich

CURITIBA  
2024



Pontifícia Universidade Católica do Paraná  
Escola Politécnica  
Programa de Pós-Graduação em Informática

Curitiba, 27 de fevereiro de 2024.

11-2024

## DECLARAÇÃO

Declaro para os devidos fins, que **Bruna Rossetto Delazeri** defendeu a tese de Doutorado intitulada “**Reconhecimento de expressões faciais - Uma abordagem com aprendizado não supervisionado de pool de representações diversas**”, na área de concentração Ciência da Computação no dia 07 de dezembro de 2023, no qual foi aprovada.

Declaro ainda, que foram feitas todas as alterações solicitadas pela Banca Examinadora, cumprindo todas as normas de formatação definidas pelo Programa.

Por ser verdade firmo a presente declaração.

---

Prof. Dr. Emerson Cabrera Paraiso  
Coordenador do Programa de Pós-Graduação em Informática

# Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

Em primeiro lugar, agradeço a Deus pela saúde e força que me permitiram concluir este projeto.

Ao meu marido, Guido, pelo carinho, incentivo e apoio incondicional para a realização deste trabalho.

Ao meu orientador, Professor Dr. Alceu de Souza Britto Jr., pela sua orientação, suporte e incentivo.

Ao Professor Dr. Alessandro L. Koerich, pela coorientação, ensinamentos e sugestões que me ajudaram a desenvolver o trabalho da melhor maneira possível.

Aos Professores Dr. Jean Paul Barddal e Dr. André Gustavo Hochuli, pelas sugestões fornecidas que contribuíram significativamente para o desenvolvimento deste trabalho.

Aos Professores do PPGIA.

Aos colegas de laboratório.

À CAPES pela concessão da Bolsa de Doutorado.

A todos que contribuíram, direta ou indiretamente, para a realização deste trabalho.

# Resumo

O reconhecimento automático de expressões faciais tem sido um campo de pesquisa ativo na área da visão computacional há décadas. Isso se deve ao fato de que essa tecnologia permite a criação de diversas novas aplicações em áreas como interação homem-computador, análise de dados e até mesmo medicina. Nos últimos anos, foram propostos vários métodos para o reconhecimento automático de expressões faciais, sendo que a maioria deles considera sete emoções básicas como universais entre culturas e subgrupos: neutro, feliz, surpreso, medo, zangado, triste e com nojo. Para lidar com os desafios do reconhecimento de expressões faciais, têm sido cada vez mais empregadas técnicas de aprendizado profundo. Essas técnicas visam extrair características e reduzir o sobreajuste causado pela escassez de dados de treinamento disponíveis. Em particular, a técnica de aprendizagem autodidata destaca-se por ser menos restritiva no desenvolvimento de sistemas que enfrentam a falta de dados. Ela utiliza dados não rotulados provenientes de diferentes distribuições para extrair características e, assim, aprimorar o desempenho de forma mais flexível. Este trabalho apresenta o algoritmo de aprendizagem por conjunto de representações, uma nova abordagem para gerar diversas representações não supervisionadas fundamentadas nos princípios da aprendizagem autodidata. O conjunto compreende autoencoders convolucionais (AECs) aprendidos de maneira não supervisionada, promovendo a diversidade por meio de uma função de custo projetada para penalizar representações latentes de AECs semelhantes. Empregamos máquinas de vetores de suporte, *bagging* e *random forest* como métodos de classificação primários para a etapa final de classificação. Além disso, incorporamos o KnoraU, uma técnica bem estabelecida usada para selecionar dinamicamente apenas classificadores competentes de uma amostra de teste. Analisamos várias estratégias de fusão, incluindo soma, produto e empilhamento, para avaliar de forma abrangente o desempenho do conjunto. Um protocolo experimental robusto, considerando o problema de reconhecimento de expressões faciais, mostra que a abordagem proposta baseada na aprendizagem autodidata supera a acurácia dos modelos de redes neurais convolucionais ajustados. Em termos de acurácia, o método proposto apresenta vantagens de até 9,9 e 6,3 pontos percentuais em relação aos conjuntos de dados JAFFE e CK+, respectivamente, em comparação com os modelos ajustados.

**Palavras-chave:** Aprendizagem autodidata; reconhecimento de expressão facial; aprendizagem profunda; rede neural convolucional; autoencoder.

# Abstract

Automatic facial expression recognition has been an active research field in computer vision for decades. This is because this technology enables the creation of several new applications in areas such as human-computer interaction, data analytics, and even medicine. In recent years, several methods have been proposed for automatic facial expression recognition, with most of them considering seven basic expressions as universal among cultures and subgroups: neutral, happy, surprised, fear, angry, sad, and disgusted. Deep learning techniques have been increasingly employed to deal with the challenges of facial expression recognition, aiming to extract features and reduce overfitting caused by the scarcity of available training data. Self-taught learning is a less restrictive technique in the development of systems that suffer from a lack of data, using unlabeled data from different distributions to extract features and improve performance. To address the challenges of facial expression recognition, deep learning techniques have increasingly been employed. These methods aim to extract features and alleviate overfitting caused by the scarcity of available training data. Particularly, self-supervised learning stands out for being less restrictive in the development of systems dealing with data scarcity. It utilizes unlabeled data from different distributions to extract features and thereby enhance performance in a more flexible manner. This work introduces the representation ensemble learning algorithm, a novel approach for generating diverse unsupervised representations based on the principles of self-taught learning. The ensemble comprises convolutional autoencoders (CAEs) learned in an unsupervised manner, fostering diversity through a loss function designed to penalize similar CAEs' latent representations. We employ support vector machines, bagging, and random forest as primary classification methods for the final classification step. Additionally, we incorporate KnoraU, a well-established technique used to select only competent classifiers from a test sample dynamically. We evaluate various fusion strategies, including sum, product, and stacking, to comprehensively assess the ensemble's performance. A robust experimental protocol considering the facial expression recognition problem shows that the proposed approach based on self-taught learning surpasses the accuracy of fine-tuned convolutional neural network models. In terms of accuracy, the proposed method presents advantages of up to 9,9 and 6,3 percentage points in relation to the JAFFE and CK+ datasets, respectively, compared to the adjusted models.

**Keywords:** Self-taught learning; facial expression recognition; deep learning; convolutional neural network; autoencoder.

# Lista de ilustrações

Figura 1 – Fluxo computacional aplicado para solucionar problemas de reconhecimento de emoções faciais. . . . .	44
Figura 2 – Características <i>Haar</i> utilizadas pelo método proposto por Viola e Jones (2004). . . . .	46
Figura 3 – (a) Retângulo de características Haar. (b) Soma de pixels dentro do retângulo D. . . . .	47
Figura 4 – Estrutura de cascata de classificadores de características <i>Haar</i> . . . . .	48
Figura 5 – Arquitetura do <i>Autoencoder</i> . . . . .	55
Figura 6 – Camada convolucional com o campo receptivo da imagem. . . . .	57
Figura 7 – Exemplo de um filtro de tamanho 2x2. . . . .	58
Figura 8 – Operação da camada de convolução. . . . .	58
Figura 9 – Operação máxima da camada de <i>pooling</i> . . . . .	60
Figura 10 – SVM. . . . .	61
Figura 11 – <i>Estrutura geral do Bagging</i> . . . . .	64
Figura 12 – Etapas de aprendizagem autodidata: (1) Aprendizagem de representação não supervisionada; (2) Extração de características; (3) Aprendizagem supervisionada . . . . .	89
Figura 13 – Gerador automático de estratégia não supervisionada para representação de problemas. (a) Diferentes estratégias geradoras de representações. (b) AEC para aprendizagem de representações de alto nível. (c) Representações geradas com base nas estratégias: sementes (S), arquitetura AEC (A), representação latente (L), sementes + arquitetura AEC (SA), sementes + representação latente (SL), representação latente + arquitetura AEC (LA) e sementes + representação latente + arquitetura AEC (SLA). . . . .	91
Figura 14 – Arquitetura AEC padrão. . . . .	93
Figura 15 – Amostras de imagens da base de dados Kyoto. . . . .	102
Figura 16 – Amostras de imagens da base de dados LabelMe. . . . .	102
Figura 17 – Amostras de imagens da base de dados LFW. . . . .	103
Figura 18 – Amostras de imagens da base de dados STL-10. . . . .	104
Figura 19 – Amostras da base JAFFE de raiva, nojo, medo, felicidade, neutro, tristeza, e expressões de surpresa. . . . .	105
Figura 20 – Amostras da base CK+ de raiva, desprezo, desgosto, medo, feliz, tristeza, surpresa. . . . .	105

Figura 21 – Exemplo de uma imagem original, detecção de face e extração dos pontos de referências do conjunto de dados JAFFE (a) e conjunto de dados CK+ (b). . . . .	106
Figura 22 – Curvas de perda de treinamento de três autoencoders distintos. . . . .	114
Figura 23 – O impacto da variação do tamanho do conjunto na melhor configuração alcançada para o conjunto de dados JAFFE ao usar conjuntos de dados Kyoto, LabelMe e LFW como auxiliares. . . . .	119
Figura 24 – Visualização dos mapas de características extraídas de uma imagem da base JAFFE nas diferentes camadas do encoder. . . . .	120
Figura 25 – O impacto da variação do tamanho do conjunto na melhor configuração alcançada para o conjunto de dados CK+ ao usar conjuntos de dados Kyoto, LabelMe e LFW como auxiliares. . . . .	125
Figura 26 – Visualização dos mapas de características extraídas de uma imagem da base CK+ nas diferentes camadas do encoder. . . . .	126
Figura 27 – Comparação do método proposto baseado em AAD (em vermelho) com arquiteturas CNNs (em azul) em termos de acurácia (%), tempo de classificação (em segundos) e espaço em disco (em megabytes) como o tamanho do círculo. Os resultados foram calculados usando o conjunto de dados JAFFE e o protocolo LOSO. O método baseado em AAD considera o classificador final RF com seleção dinâmica KnoraU. . . . .	130
Figura 28 – Comparação do método proposto baseado em AAD (em vermelho) com arquiteturas CNNs (em azul) em termos de acurácia (%), tempo de classificação (em segundos) e espaço em disco (em megabytes) como o tamanho do círculo. Os resultados foram calculados usando o conjunto de dados CK+ e o protocolo LOSO. O método baseado em AAD considera como classificador final um conjunto de SVMs. . . . .	130

# Lista de tabelas

Tabela 1 – Principais características dos tipos de aprendizagem. . . . .	49
Tabela 2 – Características dos Métodos de Aprendizagem. . . . .	51
Tabela 3 – Principais características de estudos que utilizaram fusão em sistemas REF. . . . .	79
Tabela 4 – Principais características de estudos que utilizam AAD com bases de imagens faciais . . . . .	82
Tabela 5 – Valores de parâmetros padrão para o modelo AEC . . . . .	94
Tabela 6 – Configurações de parâmetros dos algoritmos usados no Passo 3. Classificadores: único (SVM), conjuntos (BG: <i>bagging</i> com árvore de decisão como classificador base, RF: <i>random forest</i> ), KnoraU (DT: conjunto de árvores; RF: <i>random forest</i> ). . . . .	98
Tabela 7 – Acurácia da variação de sementes aleatórias (10 representações) na base de dados JAFFE. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão (soma, produto e empilhamento). . . . .	107
Tabela 8 – Acurácia da variação de sementes aleatórias (10 representações) na base de dados CK+. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão (soma, produto e empilhamento). . . . .	108
Tabela 9 – Acurácia da variação dos conjuntos de dados não rotulados (4 representações) na base de dados JAFFE. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão de tudo (soma, produto e empilhamento). . . . .	108
Tabela 10 – Acurácia da variação dos conjuntos de dados não rotulados (4 representações) na base de dados CK+. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão de tudo (soma, produto e empilhamento). . . . .	109
Tabela 11 – Acurácia da variação das arquiteturas de redes (5 representações) na base de dados JAFFE. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão de tudo (soma, produto e empilhamento). . . . .	109

Tabela 12 – Acurácia da variação das arquiteturas de redes (5 representações) na base de dados CK+. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão de tudo (soma, produto e empilhamento). . . . .	110
Tabela 13 – Acurácia da variação do tamanho do vetor latente (10 representações) na base de dados JAFFE. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão de tudo (soma, produto e empilhamento). . . . .	110
Tabela 14 – Acurácia da variação do tamanho do vetor latente (10 representações) na base de dados CK+. Classificadores: único (SVM), <i>Ensembles</i> (BG: <i>Bagging</i> , RF: <i>Random Forest</i> ), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: <i>Random Forest</i> ). Fusão de tudo (soma, produto e empilhamento). . . . .	111
Tabela 15 – Acurácia de <i>Ensembles</i> combinando todas as estratégias (as diferentes representações) na base de dados JAFFE. . . . .	111
Tabela 16 – Acurácia de <i>Ensembles</i> combinando todas as estratégias (as diferentes representações) na base de dados CK+. . . . .	112
Tabela 17 – Comparação com trabalhos relacionados com base na aprendizagem autodidata . . . . .	112
Tabela 18 – Comparação com trabalho relacionado baseado em AAD na base de dados CK+ (área sob a curva ROC). . . . .	112
Tabela 19 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da acurácia do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	116

Tabela 20 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da acurácia do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	117
Tabela 21 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da acurácia do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	118
Tabela 22 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da acurácia do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	122

Tabela 23 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 30 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da acurácia do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	123
Tabela 24 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da acurácia do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	124
Tabela 25 – Experimento <i>ablation</i> do parâmetro $\beta$ considerando a melhor configuração do método proposto para os conjuntos de dados JAFFE e CK+. . . . .	127
Tabela 26 – Arquiteturas CNNs treinadas com o protocolo LOSO e duas estratégias de ajuste fino (1, 2). . . . .	129
Tabela 27 – Comparação de referência no conjunto de dados JAFFE usando o protocolo LOSO. . . . .	131
Tabela 28 – Comparação de referência no conjunto de dados CK+ usando o protocolo LOSO. . . . .	131
Tabela 29 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	138

Tabela 30 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	139
Tabela 31 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 15 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	140
Tabela 32 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 20 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	141

Tabela 33 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 30 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	142
Tabela 34 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 40 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	143
Tabela 35 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	144

Tabela 36 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 70 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	145
Tabela 37 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 100 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	146
Tabela 38 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 150 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	147

Tabela 39 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	148
Tabela 40 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	149
Tabela 41 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 15 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	150

Tabela 42 –	Acurácia considerando o protocolo LOSO e um <i>pool</i> com 20 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	151
Tabela 43 –	Acurácia considerando o protocolo LOSO e um <i>pool</i> com 30 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	152
Tabela 44 –	Acurácia considerando o protocolo LOSO e um <i>pool</i> com 40 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	153
Tabela 45 –	. . . . .	153

Tabela 46 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	154
Tabela 47 – . . . . .	154
Tabela 48 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 70 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	155
Tabela 49 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 100 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	156

Tabela 50 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 150 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	157
Tabela 51 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	158
Tabela 52 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	159

- Tabela 53 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 160
- Tabela 54 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 161
- Tabela 55 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 162

Tabela 56 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 40 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	163
Tabela 57 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	164
Tabela 58 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 70 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	165

Tabela 59 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 100 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	166
Tabela 60 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 150 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	167
Tabela 61 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	168

Tabela 62 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	169
Tabela 63 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 15 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	170
Tabela 64 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 20 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	171

Tabela 65 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 30 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	172
Tabela 66 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 40 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	173
Tabela 67 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	174

Tabela 68 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 70 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	175
Tabela 69 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 100 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	176
Tabela 70 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 150 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	177

Tabela 71 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	178
Tabela 72 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	179
Tabela 73 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 15 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	180

<p>Tabela 74 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 20 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i>. Os melhores resultados estão em negrito. . . . .</p>	181
<p>Tabela 75 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 30 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i>. Os melhores resultados estão em negrito. . . . .</p>	182
<p>Tabela 76 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 40 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i>. Os melhores resultados estão em negrito. . . . .</p>	183

Tabela 77 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 50 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	184
Tabela 78 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 70 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	185
Tabela 79 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 100 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	186

Tabela 80 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 150 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	187
Tabela 81 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 5 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	188
Tabela 82 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 10 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	189
Tabela 83 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 15 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	190

Tabela 84 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 191

Tabela 85 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 192

Tabela 86 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 193

Tabela 87 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito. . . . . 194

Tabela 88 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 70 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	195
Tabela 89 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 100 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	196
Tabela 90 – Acurácia considerando o protocolo LOSO e um <i>pool</i> com 150 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (*) é o desvio padrão da precisão do <i>pool</i> . Os melhores resultados estão em negrito. . . . .	197

# Lista de abreviaturas e siglas

A	Arquitetura.
AAD	Aprendizagem Autodidata.
AAM	<i>Active Appearance Model.</i>
AEC	<i>Autoencoder</i> Convolutacional.
AER	Aprendizado de <i>Ensemble</i> de Representações.
ADFES	<i>Amsterdam Dynamic Facial Expression Set.</i>
AE	<i>Autoencoder.</i>
AFM	<i>Adaptive Feature Mapping.</i>
AR	<i>Alex e Robert.</i>
AUC	<i>Area Under The ROC Curve.</i>
BG	<i>Bagging.</i>
CER	<i>Collaborative Expression Representation.</i>
CNN	Rede Neural Convolutacional, do inglês <i>Convolutional Neural Network.</i>
RBM	Máquina Restrita de Boltzmann, do inglês <i>Restricted Boltzman Machine.</i>
CRBM	Máquina Restrita de Boltzmann Convolutacional, do inglês <i>Convolutional Restricted Boltzmann Machines.</i>
CK	<i>Extended Cohn-Kanade.</i>
DBN	<i>Deep Belief Network.</i>
DFSN	Rede Sequencial Facial Profunda.
DGFN	Rede de Fusão Geométrica Diferencial.
DiCA	<i>Domain Independent Component Analysis.</i>
DCNN	<i>Deep Convolutional Neural Network.</i>
DNN	<i>Deep Neural Network.</i>
DT	Árvores de Decisão.

EC-ESN	<i>Ensemble Convolutional Echo State Network.</i>
EEG	Eletroencefalograma.
EER	<i>Equal Error Rate.</i>
ELM	<i>Extreme Learning Machine.</i>
ESN	<i>Echo State Network.</i>
GDA	<i>Generalized Discriminant Analysis.</i>
GLCM	<i>Gray Level Co-occurrence Matrix.</i>
GPU	<i>Graphical Processing Unit.</i>
HOG	<i>Histogram of Oriented Gradient.</i>
IA	Inteligência Artificial.
ICA	Análise de Componentes Independentes, do inglês <i>Independent Component Analysis.</i>
IOCS	<i>Iterative Optimization Classification Strategy.</i>
ISED	<i>Indian spontaneous expression database.</i>
IVRF	<i>Intra-class Variation Reduced Features.</i>
JAFFE	<i>Japanese Female Facial Expression.</i>
KDE	<i>Kernel Dependency Estimation.</i>
KDEF	<i>Karolinska Directed Emotional Faces.</i>
k-NN	<i>k-Nearest-Neighbours.</i>
KNORA	K Oráculos mais próximos.
KNORA-E	KNORA <i>Eliminate.</i>
KNORA-U	KNORA <i>Union.</i>
KU	Seleção Dinâmica baseada em <i>Knora Union.</i>
L	Vetor Latente.
LA	Vetor Latente + Arquitetura da Rede.
LBP	<i>Local Binary Pattern.</i>

LFW	<i>Labeled Faces in the Wild.</i>
LGC-HD	<i>Local Gradient Coding Based on Horizontal and Diagonal.</i>
LOSO	<i>Leave-one-subject-out.</i>
LR	<i>Logistic Regression.</i>
lr	<i>Learning Rate.</i>
LSTM	<i>Long Short-Term Memory.</i>
MLP	<i>Multilayer Perceptron.</i>
MRDPL	<i>Manifold Regularization Dictionary Pair Learning.</i>
MSE	<i>Mean Square Error.</i>
MUG	<i>Multimedia Understanding Group.</i>
NMF	<i>Non-negative Matrix Factorization.</i>
NVIE	<i>Natural Visible and Infrared Facial Expression.</i>
PCA	Análises dos Componentes Principais, do inglês <i>Principal Component Analysis.</i>
RADAP	<i>Regional Adaptive Affinitive Patterns.</i>
RaFD	<i>Radboud Faces Database.</i>
RBM	<i>Restricted Boltzmann Machines.</i>
REF	Reconhecimento de Expressões Faciais.
REL	<i>Representation Ensemble Learning.</i>
ReLU	<i>Rectifier Linear Unit.</i>
RF	<i>Random Forest.</i>
RL	Regressão Linear.
RMP	<i>Radial Mesh Pattern.</i>
RNA	Redes Neurais Artificiais.
S	Inicialização Aleatória, do inglês <i>Seed.</i>
SA	Inicialização Aleatória + Arquitetura da Rede.

SAE	<i>Autoencoder Esparso.</i>
SC	<i>Sparse Coding.</i>
SD	Seleção Dinâmica.
SEFW	<i>Static Facial Expression in the Wild.</i>
SGD	<i>Stochastic Gradient Descent.</i>
SIFT	<i>Scale Invariant Feature Transforms.</i>
SISC	<i>Shift-Invariant Sparse Coding.</i>
SL	Inicialização Aleatória + Vetor Latente.
SLA	Inicialização Aleatória + Vetor Latente + Arquitetura da Rede.
SMF	<i>Standard Median Filter.</i>
STL	Aprendizagem Autodidata, do inglês <i>Self-Taught Learning.</i>
STL-10	Base de Dados <i>Self-Taught Learning.</i>
SURF	<i>Speeded up Robust Features.</i>
SVM	Máquina de Vetor de Suporte, do inglês <i>Support Vector Machine.</i>
TEA	Transtorno do Espectro do Autismo.
TFEID	<i>Taiwanese Facial Expression Image Database.</i>
TL	Transfer Learning.

# Lista de símbolos

$Acc$	Acurácia.
$b$	Viés.
$A^u$	Matriz de ativação de dados não rotulados.
$A^l$	Matriz de ativação de dados rotulados.
$B^u$	Vetor de base (dicionário) de dados não rotulados.
$C$	Parâmetro de ajuste.
$C_c$	Conjunto de classificadores.
$D$	Distribuição.
$D_T$	Domínio alvo.
$D_o$	Domínio de origem.
$EoC^*$	Conjunto de classificadores reduzidos.
$f$	Classificador.
$f_T$	Função alvo preditiva.
$f_e$	Função <i>encoder</i>
$f_d$	Função <i>decoder</i> .
$g$	Valor de cinza.
$h$	Camada oculta.
$ii$	Imagem integral.
$i$	Imagem original.
$I$	Função indicadora.
$k$	Número de classes.
$L$	Conjunto de dados rotulados.
$mg$	Função de margem.
$p_i$	Filtros.

$PE$	Erro de generalização.
$p$	Valor de confiança.
$r_i^l$	Rótulo ou classe.
$R^d$	Espaço vetorial de dimensão $d$ .
$r^n$	Pontuação combinada da classe.
$T_T$	Tarefa de aprendizagem alvo.
$T_o$	Tarefa de aprendizagem de origem.
$U$	Conjunto de amostras não rotuladas.
$\delta$	Região de competência do classificador.
$vf$	Meta espaço de características.
$x$	Instância/exemplo do problema.
$x^l$	Dados rotulados.
$x^u$	Dados não rotulados.
$x^*$	Conjunto de teste.
$\hat{x}$	Dado reconstruído.
$y$	Variável de saída/classe.
$w$	Pesos da matriz.
$*$	Convolução 2D.
$\sigma$	Desvio padrão.
$\mu$	Média.
$\epsilon$	Erro.
$\theta$	Vetor de parâmetros.

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>39</b>
<b>1.1</b>	<b>Objetivos</b>	<b>41</b>
1.1.1	Objetivos Específicos	41
<b>1.2</b>	<b>Questões de Pesquisa</b>	<b>41</b>
<b>1.3</b>	<b>Contribuições</b>	<b>42</b>
<b>1.4</b>	<b>Visão Geral do Documento</b>	<b>43</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>44</b>
<b>2.1</b>	<b>Reconhecimento de Emoções Faciais</b>	<b>44</b>
2.1.1	Aplicações	44
2.1.2	Pré-processamento	45
<b>2.2</b>	<b>Aprendizagem de Máquina</b>	<b>48</b>
2.2.1	Tipos de Aprendizagens	49
2.2.2	Aprendizagem Autodidata	52
2.2.3	Aprendizagem Profunda Não Supervisionada	54
2.2.3.1	<i>Autoencoder</i>	55
2.2.3.2	<i>Autoencoder Convolutacional</i>	57
2.2.4	Classificação	60
2.2.5	Seleção Dinâmica de Classificadores	64
<b>2.3</b>	<b>Metodologias de Fusão</b>	<b>65</b>
2.3.1	Regra da Soma	66
2.3.2	Regra do Produto	67
2.3.3	Empilhamento	67
<b>2.4</b>	<b>Considerações Finais</b>	<b>68</b>
<b>3</b>	<b>ESTADO DA ARTE</b>	<b>70</b>
<b>3.1</b>	<b>Sistemas de Reconhecimento de Expressão Facial</b>	<b>70</b>
<b>3.2</b>	<b>Combinação em Sistemas REF</b>	<b>76</b>
<b>3.3</b>	<b>Aplicações da Aprendizagem Autodidata</b>	<b>79</b>
3.3.1	Aprendizagem Autodidata em Imagens Faciais	80
3.3.2	Aprendizagem Autodidata em Outras Aplicações	82
<b>3.4</b>	<b>Considerações Finais</b>	<b>87</b>
<b>4</b>	<b>MÉTODO PROPOSTO</b>	<b>88</b>
<b>4.1</b>	<b>Aprendizagem de Representação (Passo 1)</b>	<b>88</b>
4.1.1	Estratégias para Gerar Diversidade	88

4.1.2	Função de Custo Personalizada . . . . .	91
4.1.3	Método Aprendizado de <i>Ensemble</i> de Representações (AER) . . . . .	93
4.2	<b>Extração de Características (Passo 2)</b> . . . . .	<b>96</b>
4.3	<b>Treinando um Classificador (Passo 3)</b> . . . . .	<b>97</b>
4.4	<b>Considerações finais</b> . . . . .	<b>99</b>
<b>5</b>	<b>RESULTADOS EXPERIMENTAIS</b> . . . . .	<b>101</b>
<b>5.1</b>	<b>Protocolo Experimental</b> . . . . .	<b>101</b>
<b>5.2</b>	<b>Resultados Para o Uso de Estratégias</b> . . . . .	<b>106</b>
5.2.1	Experimento 1 - Variando as inicializações (sementes) aleatórias . . . . .	106
5.2.2	Experimento 2 - Variando os conjuntos de dados de treinamento . . . . .	107
5.2.3	Experimento 3 - Variando a Arquitetura AEC . . . . .	109
5.2.4	Experimento 4 - Variando o Tamanho do Vetor Latente . . . . .	110
5.2.5	Combinando Todas as Estratégias . . . . .	111
5.2.6	Discussões . . . . .	113
<b>5.3</b>	<b>Resultados Para o Uso da Função de Custo Customizada Combinada com as Estratégias</b> . . . . .	<b>113</b>
5.3.1	JAFFE como Base Alvo . . . . .	115
5.3.2	CK+ como Base Alvo . . . . .	121
<b>5.4</b>	<b><i>Ablation</i></b> . . . . .	<b>125</b>
<b>5.5</b>	<b>Comparação com soluções baseadas em CNN e trabalhos relacionados</b>	<b>127</b>
<b>5.6</b>	<b>Discussões</b> . . . . .	<b>132</b>
<b>6</b>	<b>CONCLUSÃO</b> . . . . .	<b>135</b>
<b>A</b>	<b>APÊNDICE</b> . . . . .	<b>137</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>198</b>

# 1 Introdução

Engenheiros, matemáticos e cientistas da computação estão explorando diferentes maneiras para reproduzir abordagens capazes de implementar algoritmos eficientes para sistemas de reconhecimento de expressões faciais (REF) a partir de imagens. A natureza invariante do comportamento das expressões humanas entre indivíduos diferentes, bem como a influência de fatores biológicos e sociais na comunicação ao longo do tempo, são questões significativas que estão sendo estudadas e modeladas computacionalmente (CANAL et al., 2022).

O estudo do reconhecimento de expressões faciais está em andamento há mais de duas décadas, pois é uma forma natural e significativa de transmitir estados emocionais durante a comunicação humana que desempenha um papel essencial na sociedade. Além disso, o reconhecimento de emoções tem diversas aplicações, algumas delas com impacto social substancial, como a identificação do espectro do autismo (YEUNG, 2022) e de transtornos de depressão crônica (VEHLEN et al., 2023), ou o monitoramento da fadiga do motorista em sistemas de carros de segurança (QIAO et al., 2016).

Os principais desafios da tarefa REF (Reconhecimento de Expressão Facial) estão relacionados com as diferentes condições ambientais, como variações de iluminação, formatos de cabeça não frontais, expressão facial de baixa intensidade e diferenças nas expressões faciais entre gênero, cultura e grupos etários (LI; DENG, 2020). Além disso, a escassez em termos de quantidade e qualidade dos dados de treinamento disponíveis (amostras de emoções faciais) para algumas aplicações específicas de REF torna cara uma abordagem totalmente supervisionada (RATHOUR et al., 2022). Em outras palavras, uma limitação significativa dos sistemas puramente supervisionados é a sua dependência de grandes volumes de dados rotulados. A rotulagem é, em algumas aplicações REF, um processo caro e demorado. Os métodos baseados em aprendizagem profunda disseminados com modelos de redes neurais convolucionais, do inglês *convolutional neural network* (CNN), tornaram tal limitação ainda mais evidente (GANAIE et al., 2022).

Uma alternativa emergente para mitigar o problema relacionado à falta de dados rotulados é a aprendizagem por transferência (do inglês, *Transfer Learning* - TL). Um caso particular de TL, denominado aprendizagem autodidata (AAD), do inglês *self-taught learning* (STL) (RAINA et al., 2007), consiste em aprender uma representação (extrator de características) a partir de dados não rotulados, posteriormente aplicada a um domínio alvo que é potencialmente de uma distribuição diferente. A ideia principal por trás da AAD baseia-se na aprendizagem humana natural onde os dados não rotulados são considerados uma base essencial para a aprendizagem de alto nível, sendo responsáveis por fornecer um poder de discriminação mais significativo (RAINA et al., 2007). Mesmo com dados rotulados suficientes, a AAD pode melhorar o processo de aprendizagem, como observado

nos experimentos realizados em [Bastien et al. \(2010\)](#).

Os principais desafios no reconhecimento eficiente de expressões faciais são a seleção de técnicas eficientes de extração e classificação de características. Recursos artesanais, como *Local Binary Patterns* (LBP) ([OJALA; PIETIKÄINEN; HARWOOD, 1996](#); [ZAVASCHI et al., 2013](#)), *Scale Invariant Feature Transforms* (SIFT) ([LOWE, 2004](#)), *Gray Level Co-occurrence Matrix* (GLCM) ([HARALICK; SHANMUGAM; DINSTEN, 1973](#)), *Speeded up Robust Features* (SURF) ([BAY et al., 2006](#)) e *Histograms of Oriented Gradient* (HOG) ([DALAL; TRIGGS, 2005](#)), alcançaram um avanço em vários campos. Embora seja possível usar diferentes descritores para extrair recursos artesanais, atualmente é comum optar por aprender a representação, retirando do desenvolvedor a tarefa de definir os recursos.

O potencial das abordagens de aprendizagem profunda para gerar representações de dados altamente eficazes tem sido amplamente reconhecido. Em pesquisas FER, soluções baseadas em CNNs ganharam destaque ([CANAL et al., 2022](#)). Da mesma forma, o *autoencoder* convolucional (AEC) aparece como uma estratégia promissora para viabilizar soluções baseadas em aprendizagens não supervisionadas ([FENG; YU; DUARTE, 2020](#); [ALLOGNON; BRITTO; KOERICH, 2020](#)).

Este trabalho introduz o Algoritmo de Aprendizado de *Ensemble* de Representações (AER) como uma abordagem para gerar um conjunto diversificado de representações, utilizando princípios de Aprendizado Não Supervisionado baseado na Aprendizagem Autodidata. O AER baseia-se na ideia central de explorar múltiplas estratégias para aprender representações (extratores de características) de maneira não supervisionada. Além disso, incorpora uma função de custo personalizada, elaborada especificamente para promover a diversidade entre as representações geradas, ou seja, representações distintas e complementares, ampliando a abrangência e a riqueza das informações capturadas pelo conjunto de representações resultante. A justificativa é investigar como as representações aprendidas de maneira não supervisionadas podem contribuir para o desempenho da solução FER. Além disso, no contexto do problema FER, avaliamos quão competitivos são os resultados do AER em comparação com arquiteturas baseadas em CNN treinadas em uma abordagem supervisionada e também comparamos nosso método com o estado da arte atual, utilizando o mesmo protocolo experimental.

Acreditamos que além de apresentar um novo método baseado em AAD para gerar um conjunto de representações complementares, é importante avaliar seu desempenho em relação a soluções baseadas em supervisão. Um protocolo experimental robusto considerando quatro conjuntos de dados auxiliares (Kyoto, LFW, LabelMe e STL-10) necessários para a geração não supervisionada de representações propostas aqui e dois conjuntos de dados REF bem conhecidos (JAFFE e CK+) mostrou que o método proposto pode superar a precisão de soluções baseadas em CNNs, incluindo duas estratégias de ajuste fino aplicadas a dez arquiteturas CNNs diferentes e o treinamento de duas CNNs do zero. Além disso, a estrutura REF proposta demonstrou superioridade sobre vários estudos publicados

recentemente em periódicos de alto impacto, abrangendo abordagens de aprendizagem profunda e estruturas artesanais e seguindo o protocolo de validação cruzada de deixar um sujeito de fora (*Leave-one-subject-out* - LOSO).

## 1.1 Objetivos

Este trabalho tem como objetivo principal desenvolver um algoritmo de aprendizado de *ensemble* de representações que introduza uma nova abordagem na geração de representações não supervisionadas, com foco na promoção da diversidade, baseada no conceito de aprendizagem autodidata.

### 1.1.1 Objetivos Específicos

Os objetivos específicos são:

- Aplicar a técnica autodidata para a aprendizagem de representações, utilizando uma abordagem não supervisionada.
- Selecionar conjuntos de dados de contextos diversos para a aprendizagem de representações e extrair diferentes características dos conjuntos de dados alvo.
- Desenvolver um algoritmo para criação de um *pool* de representações não supervisionadas, incorporando técnicas para promover diversidade e aplicando o conceito AAD.
- Propor uma função de custo que leve em consideração a dissimilaridade entre as representações geradas.
- Avaliar o uso de estratégias para diversificar a geração do conjunto de representações não supervisionadas.
- Comparar o desempenho da solução proposta com estratégias supervisionadas.

## 1.2 Questões de Pesquisa

Quatro questões de pesquisa orientam este trabalho:

Questão 1: Quais estratégias podem ser adotadas para assegurar a diversidade na criação do conjunto de representações não supervisionadas?

Questão 2: O uso de um conjunto de representações não supervisionadas geradas pelo algoritmo AER contribui para o desempenho das soluções REF?

Questão 3: Uma função de custo planejada de maneira adequada pode equilibrar efetivamente o aprendizado e a diversificação das representações?

Questão 4: Como o novo algoritmo AER se compara às arquiteturas supervisionadas baseadas em CNN em relação ao desempenho de reconhecimento de expressões faciais?

A primeira questão diz respeito às estratégias utilizadas para promover a diversidade no conjunto de representações. A segunda questão está relacionada à ideia de explorar a geração de um conjunto diverso de representações não supervisionadas geradas pelo algoritmo proposto. A terceira questão se refere à possibilidade de introduzir uma função de custo personalizada, visando promover a diversificação das representações criadas. Na quarta questão, acreditamos que mais importante do que fornecer um novo método baseado em AAD para gerar um conjunto de representações complementares é mostrar o quão distante seu desempenho está de soluções baseadas em supervisão.

### 1.3 Contribuições

Ao responder a essas questões de pesquisa, este trabalho fornece as contribuições listadas abaixo:

- Um novo algoritmo para a geração de representações não supervisionadas, utilizando o conceito de aprendizagem autodidata e enfatizando a diversidade.
- Comparação robusta de desempenho de uma solução baseada em AAD e estratégias supervisionadas sob o mesmo protocolo experimental.
- Proposta de uma nova função de custo que leva em consideração a dissimilaridade entre representações geradas.
- Avaliação de diferentes estratégias para a geração de diversidade durante a geração de um pool de representações treinadas de maneira não supervisionada.
- Avaliação de diferentes estratégias de uso das representações não supervisionadas em bases alvo destinadas ao reconhecimento de expressões faciais.

Trabalho Publicado:

DELAZERI, B. R.; VERA, L. L.; BARDDAL, J. P.; KOERICH, A. L.; BRITTO JR, A. S. *Evaluation of Self-taught Learning-based Representations for Facial Emotion Recognition*. In: 2022 International Joint Conference on Neural Networks (IJCNN). [S.l.: s.n.], 2022. p. 1–8.

Trabalho Submetido:

DELAZERI, B. R.; HOCHULI, A. G.; BARDDAL, J. P.; KOERICH, A. L.; BRITTO JR, A. S.. *Representation Ensemble Learning Applied to Facial Expression Recognition*. In: *Neural Computing and Application*. (Submetido em 01-2024).

## 1.4 Visão Geral do Documento

Após a introdução realizada no Capítulo 1, apresentam-se os conceitos teóricos sobre reconhecimento de expressão facial, a aprendizagem autodidata e os métodos de fusão no Capítulo 2. Os estudos dos trabalhos relacionados encontram-se no Capítulo 3. No Capítulo 4, descreve-se o método proposto. O Capítulo 5 descreve os experimentos e análises dos resultados alcançados. O Capítulo 6 discorre sobre a conclusão e a continuidade do trabalho. Por fim, o Apêndice A apresenta resultados complementares obtidos na realização do trabalho.

## 2 Fundamentação Teórica

Neste capítulo serão abordados os conceitos de reconhecimento de expressão facial, suas principais aplicações e pré-processamentos na Seção 2.1. Posteriormente, serão descritos os métodos de aprendizagem mais utilizados na área de pesquisa, com ênfase na descrição da aprendizagem autodidata na Seção 2.2. Por fim, os métodos de combinação de classificadores são descritos na Seção 2.3 e as considerações finais são discutidas na Seção 2.4. Esses conteúdos são importantes para estabelecer os conceitos aplicados no método proposto, e na validação dos objetivos desse trabalho.

### 2.1 Reconhecimento de Emoções Faciais

Em abordagens convencionais, um sistema de reconhecimento de emoção facial consiste em 5 etapas: dados de entrada, pré-processamento, seleção de características, classificação e análise de resultados e validação, conforme ilustra a Figura 1. Para cada bloco de etapas, diferentes algoritmos são aplicados, dependendo da especificidade do problema a ser abordado.



Figura 1 – Fluxo computacional aplicado para solucionar problemas de reconhecimento de emoções faciais.

Fonte: Adaptado de Canal et al. (2022).

Nesta Seção descrevemos os principais sistemas de reconhecimento de expressão facial e os pré-processamentos utilizados.

#### 2.1.1 Aplicações

O reconhecimento de emoções faciais é uma tarefa interessante e desafiadora para ser computacionalmente explorada. Embora esta seja uma tarefa naturalmente realizada pelo ser humano, encontrar mecanismos computacionais para reproduzir de forma semelhante ainda é um problema não resolvido (BISWAS; SIL, 2015). Soluções algorítmicas capazes de interpretar emoções faciais em rostos humanos têm o potencial de beneficiar diversas

áreas de aplicação. Algumas dessas áreas incluem pesquisa acadêmica e comercial, como descrito abaixo:

- **Transtorno do Espectro do Autismo (TEA):** A capacidade de reconhecimento de emoções faciais é essencial para o desenvolvimento das habilidades sociais, pois transmitem informações importantes sobre o estado emocional. Em contrapartida, a dificuldade de comunicação e interação social é um traço central do transtorno do espectro do autismo. Estudos sugerem que medidas de reconhecimento de emoções faciais podem potencialmente servir como marcador de TEA (YEUNG, 2022).
- **Transtornos de Depressão Crônica:** Estudos realizados nas últimas décadas evidenciam o desempenho do reconhecimento de emoções faciais na detecção da depressão em pacientes. Os transtornos depressivos são caracterizados pelo comprometimento do funcionamento cognitivo social e a atenção visual durante a percepção social é alterada na depressão clínica (VEHLEN et al., 2023).
- **Realidade Virtual:** A análise de expressões faciais é importante para sistemas de realidade virtual para identificar a interação e emoção do usuário e analisar a animação o pelo conteúdo exibido. O desafio desta aplicação está em detectar a expressão já que, quando o usuário está utilizando o equipamento de realidade virtual, metade do rosto está oculto (HICKSON et al., 2019).
- **Motoristas de Veículos Autônomos:** O estado emocional do motorista é um dos principais fatores que influenciam as reações ao dirigir, podendo resultar em acidentes graves. O monitoramento contínuo dos motoristas pode ajudar a avaliar seu comportamento e prevenir acidentes. A tecnologia de detecção de expressão facial pode ser utilizada para reconhecer as expressões de um indivíduo, com o objetivo de monitorar suas emoções, aumentando assim, a segurança no trânsito e promovendo um comportamento de direção mais seguro (JAIN et al., 2023).
- **Monitoramento de Fadiga de Motorista:** Supervisionar a fadiga do motorista é essencial para a prevenção de acidentes. A análise de expressões faciais como piscar de olhos, balançar a cabeça e bocejar juntamente com características de tempo e frequência podem determinar a vigilância do motorista (QIAO et al., 2016).

Quanto mais cedo formos capazes de projetar sistemas de reconhecimento de expressões faciais, melhor podemos ajudar a entender áreas naturais da psicologia, neurociência, cognição humana e aprendizagem (PUTHANIDAM; MOH, 2018).

### 2.1.2 Pré-processamento

A etapa de pré-processamento de imagem é fundamental para o reconhecimento de expressões faciais. Seu objetivo é gerar imagens normalizadas em termos de intensidade,

tamanho e forma, que evidenciem apenas um rosto que esteja expressando uma determinada emoção. Para alcançar esse objetivo, essa etapa pode ser subdividida em várias etapas menores (CANEDO; NEVES, 2019).

## DETECÇÃO FACIAL

A detecção e o processamento de faces evoluíram como uma área de pesquisa eminente com o advento da tecnologia de vídeo e da computação de alto desempenho. Extrair rostos para fins de identificação humana é um requisito importante em aplicações como a de vigilância. O objetivo da detecção facial é encontrar e localizar rostos humanos em uma determinada imagem (MUTNEJA; SINGH, 2019)

Viola-Jones (VIOLA; JONES, 2004) é um método de detecção de face amplamente utilizado em trabalhos de reconhecimento de emoção facial que utilizam aprendizagem profunda para extração de características (JAGADEESH; BARANIDHARAN, 2022; ALSHAREKH, 2022; ULLAH et al., 2022; BENDJILLALI et al., 2019). Este método é dividido em quatro estágios: seleção de características *Haar-like* retangulares, criação de imagem integral, treino de classificador baseado em *AdaBoost* e uso de classificadores em cascata.

As características *Haar-like* referem-se a padrões retangulares que podem ser compostos por um ou mais retângulos adjacentes. Esses padrões são empregados para detectar regiões em uma imagem que exibem diferentes intensidades de pixel. Essas características podem ser aplicadas em qualquer área da imagem por meio de uma janela de varredura. O cálculo dessas características é baseado na diferença entre a soma dos pixels em uma região preta e a soma dos pixels em uma região branca, proporcionando uma representação eficiente para detecção de padrões visuais. As características *Haar-like*, conforme introduzidas por Viola e Jones (2004), são ilustradas na Figura 2a. Posteriormente, Lienhart e Maydt (2002) expandiram essa abordagem ao propor características adicionais com rotações, como representado na Figura 2b.

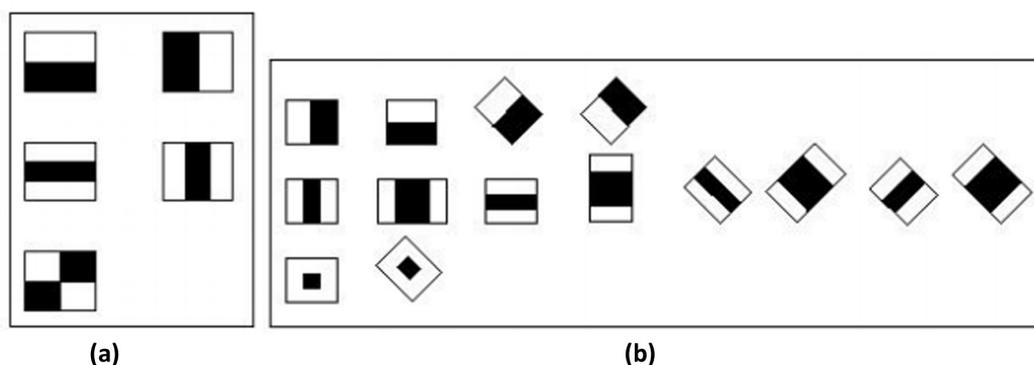


Figura 2 – Características *Haar* utilizadas pelo método proposto por Viola e Jones (2004).

Fonte: Lienhart e Maydt (2002).

Utilizar as características *Haar-like* é computacionalmente mais vantajoso do

que trabalhar com cada pixel da imagem e possibilita o reconhecimento de padrões. Entretanto, pode-se construir milhares de características diferentes considerando todas as variações de tamanho e posição das características contidas na Figura 2. Para aprimorar este processo é utilizada uma representação intermediária da imagem, chamada imagem integral. Desenvolvido por Crow (1984) o algoritmo da imagem integral é utilizado para calcular rapidamente os valores de um subconjunto retangular de uma matriz. A imagem integral  $ii$  é uma tabela que possui o mesmo tamanho  $(x, y)$  da imagem original  $i$  e cada um dos seus elementos equivalem à soma de todos os pixels à esquerda e acima de cada pixel, podendo ser descrita na equação 2.1.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (2.1)$$

Ao invés de somar todos os elementos que estão na área do retângulo de características *Haar*, utilizando a imagem integral gerada a partir da imagem original, a soma dos pixels de qualquer área retangular pode ser calculada com quatro referências de *array* em valores da imagem integral, na Figura 3 (b) representados por:  $4 + 1 - (2 + 3)$ .

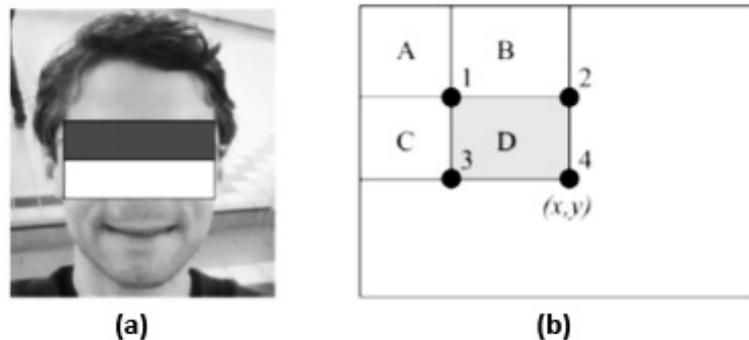


Figura 3 – (a) Retângulo de características Haar. (b) Soma de pixels dentro do retângulo D.

Fonte: Irgens et al. (2017).

Apesar de utilizar a técnica de imagem integral para agilizar o cálculo de características, calcular todo o conjunto é inviável. Viola e Jones (VIOLA; JONES, 2004) utilizaram um variante do algoritmo *AdaBoost* (FREUND; SCHAPIRE, 1997) com classificadores fracos para melhor selecionar características que possuem exemplos positivos (faces) e negativos (não faces). Com a proposta do *AdaBoost*, é gerada uma cascata de classificadores para realizar a detecção da face, conforme ilustrado na Figura 4.

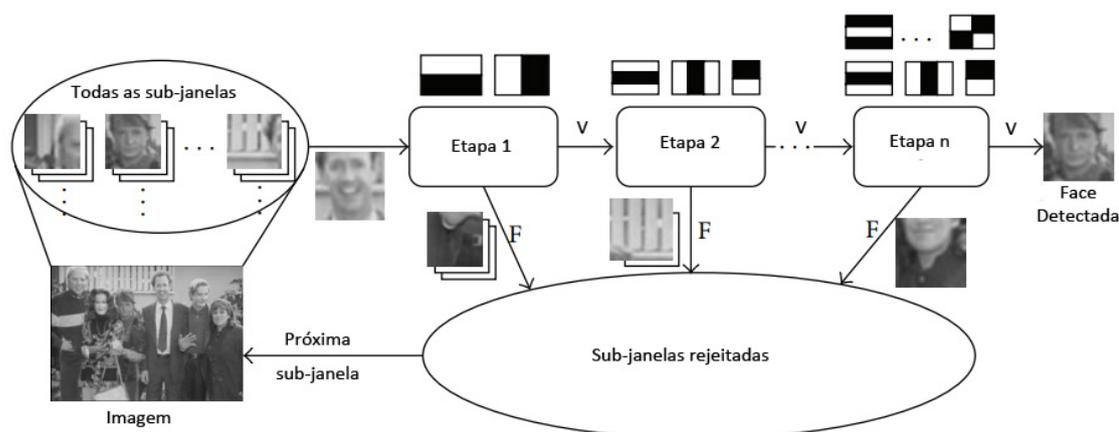


Figura 4 – Estrutura de cascata de classificadores de características *Haar*.

Fonte: adaptado de [Kim, Lee e Kim \(2015\)](#).

## PONTO DE REFERÊNCIA E ALINHAMENTO

Além da detecção, o alinhamento de faces utilizando os pontos de referência (como os olhos, por exemplo), pode melhorar substancialmente o desempenho do sistema de reconhecimento de expressão facial, pois pode reduzir a variação na escala da face e na rotação do plano ([LI; DENG, 2020](#)).

A detecção de ponto de referência é um componente projetado que fornece um meio eficaz para alinhar fotos facilitando o processamento da imagem. Utilizamos, no trabalho, a implementação de código aberto desenvolvido por [Kazemi e Sullivan \(2014\)](#), disponibilizada na Dlib ([KING, 2009](#)), como etapa do pré-processamento para o alinhamento das imagens.

## 2.2 Aprendizagem de Máquina

Aprendizagem de máquina é uma área de pesquisa em constante evolução e os algoritmos desenvolvidos podem ser categorizados de diversas formas. Na literatura, as categorias mais comuns são: aprendizagem supervisionada, não supervisionada, semi-supervisionada e por transferência. Proposto por [Raina et al. \(2007\)](#), o conceito de aprendizagem autodidata destaca-se como uma abordagem de transferência de aprendizado que se distingue por impor restrições mais flexíveis e adaptáveis durante o treinamento, ampliando assim as possibilidades de aplicação e adaptabilidade nos cenários de aprendizado de máquina. Nesta seção, descrevemos brevemente os tipos de aprendizagem mais utilizados na área de pesquisa, detalhamos a aprendizagem autodidata e comparamos as principais características dos tipos de aprendizagem apresentados. Apresentamos os principais algoritmos de aprendizagem profunda não supervisionada, detalhando o *autoencoder* convolucional, por fim, apresentamos os métodos de classificação e de seleção dinâmica de classificadores utilizados no trabalho.

Tabela 1 – Principais características dos tipos de aprendizagem.

Método	Dados Rotulados	Dados Não Rotulados	Distribuição dos Dados	Tarefa
Aprendizagem Supervisionada	Abundante	—	Mesmo	Classificação e Regressão
Aprendizagem Não Supervisionada	—	Abundante	Mesmo	Agrupamento, Redução de Dimensionalidade, Extração de Características e Geração de Novas Amostras
Aprendizagem Semi-Supervisionada	Escasso	Abundante	Mesmo	Classificação, Regressão, Agrupamento e Redução de Dimensionalidade
Aprendizagem por Transferência	Escasso	Abundante	Diferente, porém relacionado	Classificação, Regressão, Agrupamento e Redução de Dimensionalidade
Aprendizagem Autodidata	Escasso	Abundante	Diferente	Classificação, Regressão e Agrupamento

### 2.2.1 Tipos de Aprendizagens

Utilizando como exemplo a tarefa de classificação de dígitos manuscritos (tarefa alvo), os métodos de aprendizagens são executados de maneiras diferentes:

- **Aprendizagem supervisionada:** Requer um amplo conjunto de dados rotulados da tarefa alvo.
- **Aprendizagem não supervisionada:** Utiliza diversos dados não rotulados da tarefa alvo.
- **Aprendizagem semi-supervisionada:** Permite aproveitar grandes quantidades de dados não rotulados, disponíveis em muitos casos, em combinação com conjuntos tipicamente menores de dados rotulados (ENGELEN; HOOS, 2020).
- **Aprendizagem por transferência:** Utiliza dados rotulados de outra distribuição para extrair características e usa as mesmas características para treinar um modelo supervisionado em uma distribuição com poucos dados rotulados.
- **Aprendizagem autodidata:** Utiliza dados não rotulados de diferente distribuição para aprender uma representação de alto nível dos dados e, utiliza esta representação para extrair as características dos dados rotulados da tarefa alvo para, posteriormente, aplicar em uma tarefa de classificação tradicional.

A Tabela 1 apresenta as características das aprendizagens, destacando as principais diferenças em relação à forma de utilização dos dados em cada uma.

O aprendizado supervisionado possui variáveis de entrada  $x$ , uma variável de saída  $y$  e utiliza um algoritmo para aprender a função de mapeamento da entrada para uma saída  $y = f(x)$ . O objetivo é aproximar a função de mapeamento de modo que quando

novos dados de entrada  $x$  forem apresentados, o algoritmo seja capaz de prever a saída correspondente  $y$ . Esse cenário é conhecido como supervisionado, pois o conjunto de dados de treinamento é rotulado e o processo de aprendizado é supervisionado por um professor que conhece as respostas corretas e corrige as previsões do algoritmo. As tarefas de aprendizado supervisionado podem incluir classificação e regressão. (BONACCORSO, 2017) (BROWNLEE, 2016a).

A tarefa de classificação consiste em atribuir a uma variável de saída uma categoria, como, por exemplo, na classificação binária entre cachorro e gato, ou na classificação de múltiplas classes para determinar a raça de um cão. Já na tarefa de regressão, o valor de saída é um número real, como na previsão de preços futuros de ações com base em dados históricos e tendências atuais do mercado. Os algoritmos mais populares para tarefas supervisionadas incluem: Regressão Linear (RL) (FERNÁNDEZ-DELGADO et al., 2019) para regressão, *Random Forest* (RF) (ANTONIADIS; LAMBERT-LACROIX; POGGI, 2021) para ambos e *Support Vector Machine* (SVM) (CERVANTES et al., 2020) para classificação.

Com o aumento da disponibilidade de dados não rotulados, a aprendizagem não supervisionada tem despertado interesse entre pesquisadores e profissionais. Nesse tipo de aprendizagem, os dados não possuem rótulos e são mais abundantes e fáceis de obter, em comparação com os dados rotulados.

A aprendizagem não supervisionada é empregada para identificar padrões interessantes e úteis entre os dados (FRIEDMAN; HASTIE; TIBSHIRANI, 2001). Aplicações como o reconhecimento de padrões, mineração na web, análise de redes sociais, sistemas de recomendação e detecção de fraudes são frequentemente associadas ao uso da aprendizagem não supervisionada (CELEBI; AYDIN, 2016). Embora algumas dessas aplicações possam envolver tarefas que, em sua essência, exigem alguma forma de supervisão, como o treinamento inicial de modelos ou a validação de resultados, é importante destacar que a aprendizagem não supervisionada desempenha um papel crucial em várias etapas desses processos.

Uma tarefa típica dessa aprendizagem é o agrupamento, por exemplo, utilizando o *K-means*. No entanto, outras tarefas, como estimação de densidade, redução de dimensionalidade, extração de características e geração de novas amostras, também são aplicadas (SIMEONE, 2018). Nesta abordagem, o conjunto de  $N$  observações  $(x_1, x_2, \dots, x_N)$  não possui as variáveis de resposta associadas. O objetivo é descobrir estruturas ocultas nos dados (FRIEDMAN; HASTIE; TIBSHIRANI, 2001).

Conceitualmente situado entre o aprendizado supervisionado e não supervisionado, a aprendizagem semi-supervisionada utiliza dados rotulados escassos e dados não rotulados em grande quantidade, geralmente extraídos da mesma distribuição de dados subjacente aos dados rotulados, para executar determinadas tarefas (ENGELEN; HOOS, 2020). Além das amostras rotuladas  $D_o$ , a aprendizagem semi-supervisionada utiliza um conjunto

de amostras não rotuladas  $U$ . O objetivo é usar as amostras em  $U$  para melhorar o classificador  $f$ , onde  $f$  é construído apenas usando amostras em  $D_o$  (CHEPLYGINA; BRUIJNE; PLUIM, 2019). Com o desempenho dominante das redes neurais nas áreas de pesquisa, a aprendizagem semi-supervisionada profunda se tornou uma área de interesse com uma variedade de aplicações práticas (YANG et al., 2022), além disso, diferentes técnicas foram desenvolvidas e descritas para aplicar a aprendizagem semi-supervisionada (PRAKASH; NITHYA, 2014).

Embora a aprendizagem semi-supervisionada e não supervisionada compartilhem da mesma lógica de aprendizado não supervisionado, a configuração de aprendizado é diferente. Especificamente, o semi-supervisionado assume a disponibilidade de dados rotulados limitados e seu principal desafio é expandir o conjunto rotulado com dados não rotulados abundantes. O não supervisionado não assume dados rotulados para a principal tarefa de aprendizado e seu principal desafio é aprender representações genéricas de tarefas a partir de dados não rotulados (CHEN et al., 2022).

Dado um domínio de origem  $D_o$  e uma tarefa de aprendizagem  $T_o$ , um domínio alvo  $D_T$  e uma tarefa de aprendizagem  $T_T$ , o objetivo da aprendizagem por transferência é melhorar o aprendizado da função alvo preditiva  $f_T(\cdot)$  no  $D_T$  usando o conhecimento aprendido em  $D_o$  e  $T_o$  onde  $D_o \neq D_T$  ou  $T_o \neq T_T$ , porém relacionado. A aprendizagem por transferência possui algumas subcategorias, detalhadas em Pan e Yang (2010) e Zhang, Li e Ogunbona (2017). A Tabela 2 apresenta algumas das principais categorias de aprendizagem por transferência.

Tabela 2 – Características dos Métodos de Aprendizagem.

Fonte: adaptado Pan e Yang (2010).

Categorias Aprendizagem por Transferência	Sub-Categorias	Domínio de origem Rotulado	Domínio Alvo Rotulado	Tarefas	Domínio de origem alvo	Tarefa origem e Alvo
Indutivo	Aprendizagem Multitarefa	Disponível	Disponível	Regressão e classificação	O mesmo	Diferente, porém relacionado
	Aprendizagem Autodidata	Indisponível	Disponível	Regressão, classificação e agrupamento	Diferente	Diferente
Transdutivo	Adaptação de Domínio, Viés de Seleção da Amostra, Mudança Covariável	Disponível	Indisponível	Regressão e classificação	Diferente	Diferente
		Indisponível	Indisponível	Agrupamento e redução de dimensionalidade	Diferente, porém relacionado	Diferente, porém relacionado

No contexto da aprendizagem por transferência transdutiva, as tarefas tanto no domínio de origem quanto no domínio de destino são semelhantes, porém, não idênticos. Nessa configuração, destaca-se a ausência de dados rotulados no domínio de destino, enquanto uma quantidade substancial de dados rotulados está prontamente disponível no

domínio de origem. Nesse cenário, o desafio reside na capacidade de utilizar efetivamente a informação disponível no domínio de origem para melhorar o desempenho da aprendizagem no domínio de destino, onde a escassez de rótulos é uma limitação significativa. O paradigma de aprendizagem por transferência transdutiva surge como uma abordagem estratégica para superar essa lacuna, buscando otimizar a generalização do conhecimento adquirido no domínio de origem para beneficiar a tarefa de destino em um domínio diferente (PAN; YANG, 2010).

A aprendizagem autodidata é uma forma de aprendizagem por transferência que utiliza dados não rotulados de distribuição diferente do problema em questão para aprender uma representação de alto nível por meio de uma abordagem de aprendizagem não supervisionada (RAINA et al., 2007; RAJAT, 2019). Em seguida, as características dos dados rotulados são extraídas a partir da representação aprendida anteriormente. Por fim, um algoritmo de classificação supervisionado é utilizado para classificar os dados rotulados. Devido à sua capacidade de lidar com dados escassos, tanto rotulados quanto não rotulados, a abordagem autodidata tem sido amplamente utilizada em diversas aplicações.

## 2.2.2 Aprendizagem Autodidata

A aprendizagem autodidata (AAD) foi inicialmente proposta em Raina et al. (2007) como uma estratégia de aprendizagem não supervisionada na qual dados não rotulados e rotulados, não necessariamente mostrando a mesma distribuição, podem ser usados para construir um modelo preditivo. A motivação para o uso de dados não rotulados é mitigar a limitação de algoritmos de aprendizagem causada pela falta de instâncias para treinamento de modelos em muitas aplicações. Autores que defendem a AAD explicam que é possível obter uma representação robusta a partir de amostras não rotuladas, como estruturas discriminantes de baixo nível (cantos, curvas e formas), a partir de imagens não rotuladas de paisagens naturais, como sugerido em Lee et al. (2011).

Uma abordagem baseada em AAD geralmente exige três etapas, além de conjuntos de dados auxiliares e de destino (GAN et al., 2014; LEE et al., 2008; HUANG et al., 2009; MARKOV; MATSUI, 2013). Primeiro, um algoritmo não supervisionado aprende a representação do problema (Etapa de aprendizagem de representação não supervisionada) a partir de um conjunto de dados auxiliar composto por amostras não rotuladas. Tal processo caracteriza o aprendizado não supervisionado de características, no qual os dados não rotulados podem pertencer a um domínio diferente daquele de destino. Em segundo lugar, o extrator de características é aplicado ao conjunto de dados alvo para mapear os dados brutos em um novo espaço (Etapa de extração de características). Na última etapa, os vetores de características extraídos que representam o conjunto de dados alvo são usados para treinar (Etapa de aprendizagem supervisionada) o modelo preditivo final (MARKOV; MATSUI, 2013; RAINA; MADHAVAN; NG, 2009). Em resumo, AAD não pode ser considerada uma estratégia de aprendizagem puramente não supervisionada, uma

vez que os dados não rotulados usados podem ser de qualquer classe e não necessariamente corresponder ao conjunto de classes dos dados rotulados alvo (PAPAMARTZIVANOS; MÁRMOL; KAMBOURAKIS, 2019).

Formalmente, uma solução baseada em AAD usa um pequeno conjunto contendo  $M$  amostras rotuladas  $X^l = \{x_i^l, r_i^l\}_{i=1}^M$  de distribuição desconhecida  $D$ , e um conjunto de dados não rotulados  $X^u = \{x_i^u\}_{i=1}^N$  disponíveis, que não necessariamente precisam apresentar a mesma distribuição que os dados rotulados e não precisam estar associados aos rótulos de  $X^l$ . Os dados não supervisionados em  $X^u$  são usados para aprender uma representação de alto nível. Dada uma entrada  $x_i^u \in R^d$ , o método de representação de características não supervisionado fornece padrões ou vetores de base como  $B^u = \{b_i^u\}_{i=1}^k \in R^d \times K$  de modo que cada amostra  $x_i^u$  possa ser representada combinando algumas funções básicas conforme indicado na Equação 2.2.

$$x_i^u = \sum_{k=1}^k a_{i,k}^u b_k^u \quad (2.2)$$

onde  $a_{i,k}^u$  é a combinação dos coeficientes de  $x_i^u$ , chamados ativações e  $b_k^u$  são as funções base, ou seja, as características de alto nível. Para impor a restrição de que apenas alguns vetores de base são usados para uma entrada  $x_i^u$ , apenas alguns valores de gatilho serão diferentes de zero. Considerando todos os dados não rotulados da base de treinamento, a Equação 2.2 pode ser reescrita como uma matriz como na Equação 2.3.

$$X^u = B^u A^u \quad (2.3)$$

onde  $X^u = \{x_i^u\}_{i=1}^N \in R^{d \times N}$  é o produto de duas matrizes  $B^u = \{b_i^u\}_{i=1}^k \in R^{d \times K}$  e  $A^u = \{a_i^u\}_{i=1}^N \in R^{K \times N}$ , cada  $a_i^u$  representa o vetor coeficiente do vetor dado  $x_i^u$ . Esta Equação decompõe a matriz de dados  $X^u$  em duas matrizes  $A^u$ , conhecidas como matriz de ativação e  $B^u$ , conhecido como matriz de dicionário. Tal esquema de representação pode ser obtido usando técnicas como Análise de Componentes Principais, do inglês *Principal Component Analysis* (PCA), Análise de Componentes Independentes, do inglês *Independent Component Analysis* (ICA), Máquina Restrita de Boltzmann, do inglês *Restricted Boltzmann Machine* (RBM) e *Autoencoders* (LU et al., 2020).

De acordo com a abordagem autodidata, uma vez que o dicionário  $B^u$  tenha sido aprendido a partir dos dados de entrada não rotulados  $X^u$ , ele é utilizado para calcular as ativações dos dados rotulados  $X^l$ . Em outras palavras, os vetores rotulados  $x_i^l$  podem ser expressos como combinações de algumas funções básicas, como demonstrado na Equação 2.4.

$$X^l = B^u A^l \quad (2.4)$$

onde  $A^l = \{a_i^l\}_{i=1}^M \in R^{K \times M}$  é a matriz de ativação correspondente aos dados rotulados. Esta pode ser considerada uma nova forma de representar  $X^l$ , onde é possível atribuir à

classe original  $y_i$  cada vetor de ativação  $a_i^l$ , e então obter uma nova representação para a base rotulada alvo, que pode ser usada para construir algum classificador de forma supervisionada (um SVM por exemplo) (MARKOV; MATSUI, 2013).

Um estudo interessante é apresentado em Bengio et al. (2011). Os autores mostram que o aprendizado profundo se beneficia mais ao usar exemplos fora da distribuição. Com isso em mente, os pesquisadores usaram AAD como uma alternativa para redes de pré-treinamento para obter um bom ponto de partida para seus parâmetros (ZHANG; LI; OGUNBONA, 2017). Uma vez aprendida a representação, o banco de dados alvo pode ajustá-la, introduzindo uma função softmax na última camada da rede, como feito em Zhang et al. (2019), Kamath, Liu e Whitaker (2019). Esse pré-treinamento baseado em AAD fornece boa inicialização e captura dependências complexas entre parâmetros. O pré-treinamento não supervisionado pode ser uma forma de regularização cujo efeito não desaparece com mais dados em comparação com as técnicas de regularização L1/L2 (ERHAN et al., 2010). Do ponto de vista da regularização, a estrutura de representação aprendida sem supervisão pode ser transferida do domínio auxiliar e adaptada ao domínio alvo usando um parâmetro de ajuste, como feito em Liu, Yang e Tao (2017).

### 2.2.3 Aprendizagem Profunda Não Supervisionada

O uso de Redes Neurais Convolucionais (CNNs) tem aumentado a eficácia do aprendizado profundo em visão computacional. O desenvolvimento de novas arquiteturas de alto desempenho e o aumento no poder de processamento computacional forneceram novas ferramentas para abordar o problema (TURABZADEH et al., 2017). As CNNs têm sido eficazes na tarefa de reconhecimento de expressões faciais porque são capazes de aprender recursos diretamente das imagens, dispensando a necessidade de recursos manuais, embora o treinamento desses modelos seja computacionalmente caro (LUTTRELL et al., 2018). Na literatura, há diversos trabalhos dedicados ao desenvolvimento de sistemas REF com abordagens de CNNs nos últimos anos (CANAL et al., 2022).

Os algoritmos de aprendizado não supervisionados como *Autoencoder* (AE) (CHARTE et al., 2018), *Sparse Coding* (SC) (OLSHAUSEN; FIELD, 1996), *Restricted Boltzman Machine* (RBM) (SMOLENSKY, 1986), *Deep Belief Network* (DBN) (HINTON; OSINDERO; TEH, 2006), *Nonnegative Matrix Factorization* (NMF) (LEE; SEUNG, 1999), *Independent Component Analysis* (ICA) (COMON, 1994) e *Principal Component Analysis* (PCA) (JOLLIFFE; CADIMA, 2016) têm sido utilizados na aprendizagem autodidata na etapa não supervisionada de representação.

As arquiteturas de aprendizado profundo buscam criar uma hierarquia de representações úteis dos dados em vários níveis. Essa hierarquia facilita o desenvolvimento de representações adaptadas a tarefas específicas e capazes de emprestar força estatística de tarefas relacionadas. A aprendizagem de representação também pode resultar em

características de nível superior que são mais robustas diante das variações presentes em dados reais (BENGIO et al., 2011).

A utilização do pré-treinamento não supervisionado da rede é apropriada quando há uma quantidade significativa de exemplos não rotulados disponíveis para a inicialização do modelo (BROWNEE, 2018). Neste trabalho, foi empregado o *autoencoder* convolucional como extrator de características não supervisionado para aplicar a técnica de aprendizagem autodidata.

### 2.2.3.1 Autoencoder

*Autoencoder* é uma rede neural artificial que utiliza a aprendizagem não supervisionada para aprender um mapeamento para reconstruir os dados de entrada na sua camada de saída. O *autoencoder* possui dois processos principais: *encoder* e *decoder*. O *encoder* mapeia os dados de entrada em uma camada oculta e o *decoder* reconstrói a camada oculta e produz uma saída aproximada à entrada (NG et al., 2011; GOODFELLOW; BENGIO; COURVILLE, 2016). A arquitetura do *autoencoder* é ilustrada pela Figura 5.

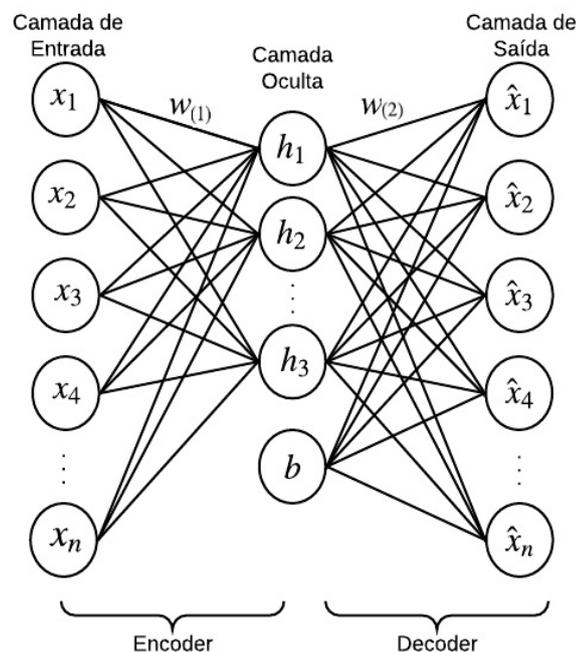


Figura 5 – Arquitetura do *Autoencoder*.

Na Figura 5,  $x_k, k \in 1, \dots, n$  representa a entrada do *autoencoder*,  $h_j, j \in 1, \dots, m$  é o valor das unidades da camada oculta,  $\hat{x}_k, k \in 1, \dots, n$  é a saída aproximada,  $w_{(i)}, i \in 1, 2$  indicam os pesos da matriz e  $b$  é o termo do viés (QU et al., 2017; WANG, 2016).

Dado um vetor de entrada  $x$ , a função encoder é dada por  $h = f_e(w^1x + b)$ , onde  $w$  representa a matriz de pesos entre cada neurônio da camada de entrada e da camada

oculta,  $f_e$  é a função de ativação não linear usada para suavizar a saída da camada oculta. A função decoder é dada por  $\hat{x} = f_d[w_2 f_e(w_1 x + b)]$ , onde  $\hat{x}$  representa o dado de saída reconstruído pela camada oculta,  $w_2$  a matriz de peso entre a camada oculta e a camada de saída e  $f_d$  a função decoder de ativação para suavizar o resultado da camada de saída (QU et al., 2017; WANG, 2016).

O objetivo do treinamento de um *autoencoder* é obter o conjunto de pesos da codificação  $w^1$  e decodificação  $w^2$  cujo erro entre os dados de entrada  $x$  e saída  $\hat{x}$  seja minimizado. O objetivo do aprendizado pode ser escrito como:

$$\arg \min_{w^1, w^2} \|x - \hat{x}\|_2^2 \quad (2.5)$$

Em geral, a saída do *decoder* não é a principal área de interesse; o treinamento do *autoencoder* está centrado na habilidade de replicar a entrada, buscando assim extrair características significativas presentes nas camadas ocultas. Há duas maneiras de aprender representações úteis (GOODFELLOW; BENGIO; COURVILLE, 2016). Uma abordagem inicial é restringir o tamanho da camada oculta  $h$  para uma dimensão menor do que a entrada  $x$ . Neste *autoencoder*, denominado como incompleto, a restrição impõe a necessidade de aprender características úteis dos dados de treinamento. Contudo, é essencial monitorar a capacidade do *encoder* e *decoder*, pois, se apresentarem muita capacidade, o *autoencoder* pode incorrer em *overfitting*, concentrando-se mais na tarefa de cópia em vez de extrair características úteis da distribuição dos dados. Um desafio semelhante surge quando a camada oculta tem o mesmo tamanho ou é maior (supercompleto) que a entrada, como discutido por (GOODFELLOW; BENGIO; COURVILLE, 2016).

Outra abordagem consiste na seleção das dimensões do código, bem como na capacidade do codificador e do decodificador, levando em consideração a complexidade da distribuição a ser modelada. Os autoencoders regularizados demonstram uma capacidade de empregar funções de custo que não apenas incentivam a replicação dos dados, mas também promovem a captura de outras propriedades essenciais, como a dispersão da representação (*Sparse Autoencoder*) e a robustez ao ruído (*Denoising Autoencoder*) (GOODFELLOW; BENGIO; COURVILLE, 2016). Este enfoque não apenas amplia a capacidade do modelo de aprendizado, mas também introduz a capacidade de adaptação a diversas complexidades presentes nos dados.

Estes autoencoders especializados têm a habilidade de aprender codificações de alta capacidade, pois são treinados para maximizar a probabilidade dos dados, indo além da mera replicação. Essa característica torna-os ferramentas valiosas na aprendizagem não supervisionada, permitindo a extração de características úteis de conjuntos de dados não rotulados. Essas características podem, então, ser aplicadas de maneira eficaz em distribuições de dados distintas (FAYYAZ et al., 2015), conferindo aos autoencoders um papel significativo na descoberta e utilização eficiente de padrões em diversos contextos.

### 2.2.3.2 Autoencoder Convolucional

As redes neurais convolucionais (*Convolutional Neural Networks* - CNN) foram primeiramente propostas por [LeCun et al. \(1989\)](#) e se tornaram conhecidas devido ao seu surpreendente desempenho em tarefas de classificação de imagens ([KRIZHEVSKY; SUTSKEVER; HINTON, 2012](#)). As CNNs são redes neurais que usam a convolução no lugar da multiplicação geral da matriz em pelo menos uma de suas camadas e são normalmente aplicadas a entradas bidimensionais.

Camadas totalmente conectadas não consideram a estrutura bidimensional ao processar dados da imagem. Estas estruturas inserem redundância nos parâmetros, forçando as características a serem globais. A tendência da visão computacional é descobrir padrões localizados que se repetem por toda a entrada. As CNNs são treinadas para reconhecer padrões em imagens. A saída da última camada convolucional é normalmente achatada (transformada em um vetor 1D) e usada como entrada para uma ou mais camadas totalmente conectadas para traduzir as características obtidas pelas camadas anteriores para a fase de saída. As CNNs possuem dois tipos de conexões: convolucionais e de *pooling* ([LENG et al., 2015](#)).

Em uma CNN, a camada mais importante é a de convolução, onde os neurônios não são conectados à cada pixel da imagem de entrada, mas somente nos pixels do campo receptivo (também chamado de filtro ou núcleo convolucional). A Figura 6 ilustra a conexão do campo receptivo com as camadas convolucionais. Cada neurônio da segunda camada convolucional é conectado somente aos pixels localizados em um pequeno retângulo da primeira camada. O funcionamento desta arquitetura permite que a rede se concentre em pequenas características de baixo nível na primeira camada para agrupá-los na próxima camada oculta.

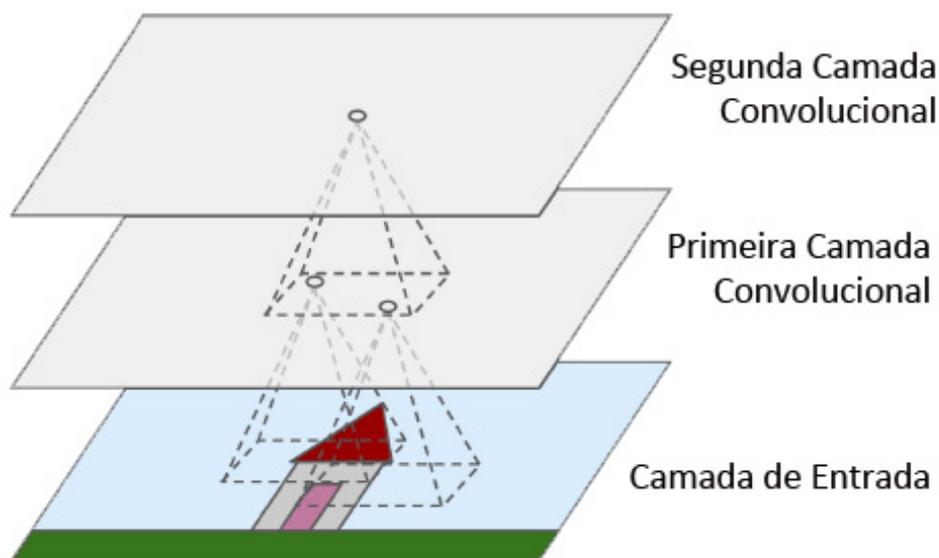


Figura 6 – Camada convolucional com o campo receptivo da imagem.

Fonte: Adaptado de [Géron \(2019\)](#).

Cada filtro em uma camada convolucional é uma grade de números discretos. Os pesos de cada filtro são iniciados aleatoriamente e aprendidos durante o treinamento da CNN. A Figura 7 ilustra o exemplo de um filtro de tamanho 2x2.

2	0
-1	3

Figura 7 – Exemplo de um filtro de tamanho 2x2.

Fonte: Khan et al. (2018).

Uma imagem de entrada é representada por uma matriz composta de uma ou mais camadas de pixels, conhecidas como canais, que utilizam valores variando de 0 a 255. Por exemplo, uma imagem RGB possui canais distintos para as cores vermelha, verde e azul. Para otimizar as operações de soma e multiplicação executadas nas camadas de convolução, a entrada passa por uma transformação simples para reescalar os valores dos pixels, definindo-os em uma escala que varia de 0 a 1. A convolução é realizada entre os filtros e a camada de entrada. A Figura 8 ilustra o processo de convolução de uma matriz de entrada 4 x 4 e o filtro de convolução 2 x 2. A camada convolucional multiplica a matriz do filtro pela parcela em destaque da matriz de entrada para gerar um valor de saída no mapa de características e assim desliza por toda a matriz de entrada.

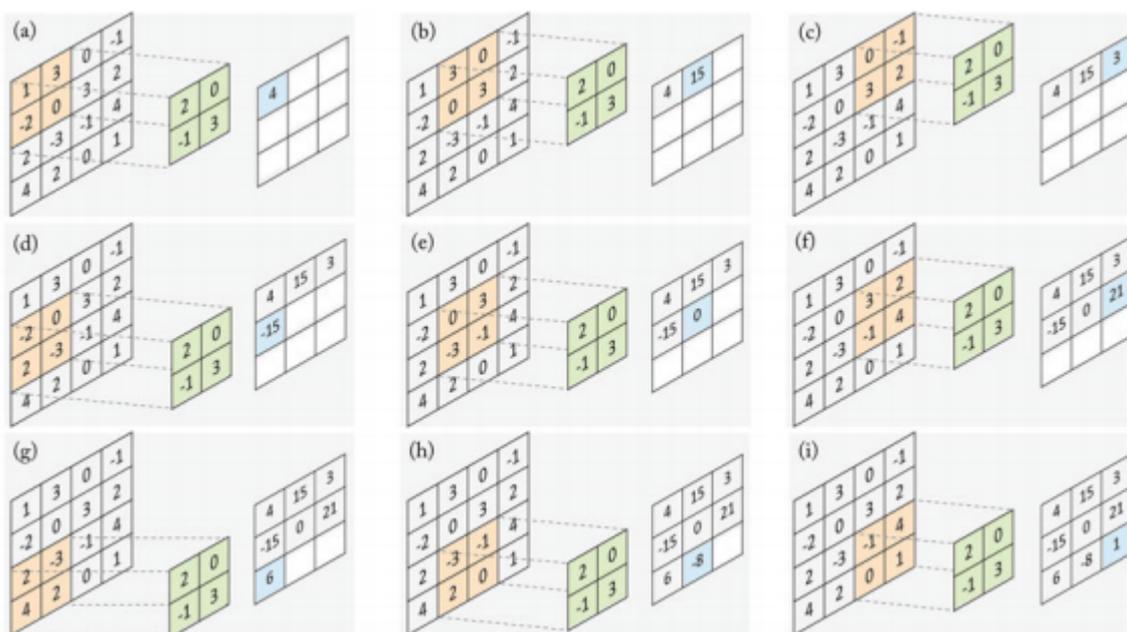


Figura 8 – Operação da camada de convolução.

Fonte: Khan et al. (2018).

No exemplo ilustrado na Figura 8, para calcular o mapa de características de saída, o filtro desliza ao longo das linhas e colunas ao passo de um pixel. Esta etapa é conhecida

como passo do filtro de convolução (*stride*) e pode ser alterado, por exemplo, para passar de 2 ou mais pixels. Conforme aumenta o tamanho do deslização do filtro, menor será o tamanho do mapa de características. Para criar redes mais profundas e obter melhor desempenho, evitar uma redução extrema das dimensões do mapa de características é necessário. Isso pode ser alcançado utilizando o preenchimento zero (*zero padding*) ao redor do mapa, ou seja, preencher com zeros as dimensões horizontais e verticais do mapa de entrada permite aumentar as dimensões da saída.

As camadas com pesos em uma CNN (totalmente conectadas e convolucionais) são frequentemente seguidas por uma função de ativação não linear. É importante utilizar uma função não linear após camadas de peso, pois permite que a rede aprenda mapeamentos não lineares. A função linear desempenha o papel de um mecanismo seletivo dentro da rede, determinando se um neurônio será ativado. Uma função de ativação frequentemente empregada em redes convolucionais é a *Rectifier Linear Unit* (ReLU). Essa função, representada por  $f_{ReLU}(x) = \max(0, x)$ , mapeia a entrada para 0 se for negativa e preserva seu valor original se for positiva (KARIMPOULI; TAHMASEBI, 2019).

A camada de *pooling* é utilizada em redes convolucionais para subamostrar a imagem de entrada e reduzir a carga computacional, o uso de memória e o número de parâmetros. A operação de subamostragem é definida por uma função de agrupamento com função média ou máxima. Assim como na camada convolucional, cada neurônio da camada de *pooling* é conectado às saídas de um número limitado de neurônios na camada anterior, localizado dentro do campo receptivo. Para isso, deve-se especificar o tamanho da região agrupada e a passada (*stride*). A Figura 9 ilustra a camada máxima de *pool* (*max pooling*), que é a mais utilizada (KHAN et al., 2018).

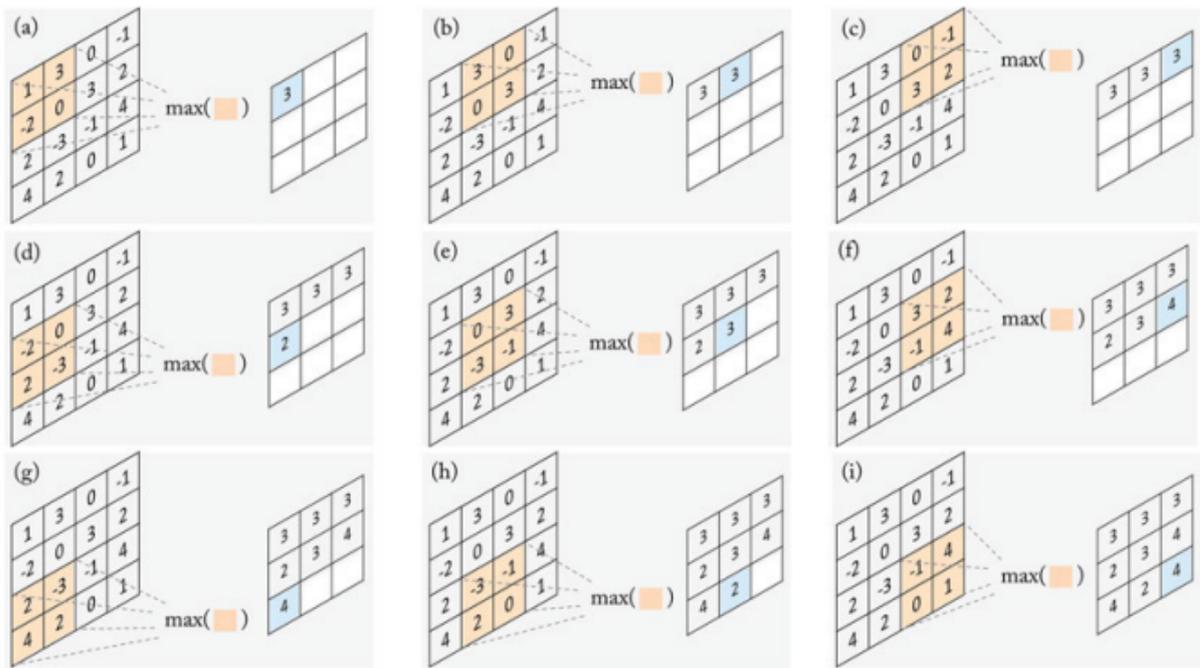


Figura 9 – Operação máxima da camada de *pooling*.

Fonte: Khan et al. (2018).

No contexto do *autoencoder* convolucional (AEC), uma extensão do *autoencoder*, são empregadas camadas convolucionais para aprender representações, seguindo a abordagem das CNNs. Diferentemente dos *autoencoders* com camadas totalmente conectadas, que negligenciam a estrutura das imagens e introduzem redundância nos parâmetros, o AEC tem a capacidade de capturar padrões específicos presentes nas imagens (LENG et al., 2015).

O *autoencoder* convolucional possui também os componentes de codificação e decodificação (CHARTE et al., 2018). O codificador é responsável por mapear os dados de entrada em características de nível superior e o decodificador reconstrói a imagem de entrada a partir do mapeamento aprendido. Para uma entrada mono-canal  $x$ , a representação do mapeamento de características  $j$ -ésimo é dado por (MASCI et al., 2011):

$$h^j = f_e(x * w^j + b^j) \tag{2.6}$$

onde o viés  $b_i$  é compartilhado em todo o mapa,  $f_e$  representa a função de ativação, e o operador  $*$  denota a convolução 2D. Essa expressão encapsula a essência do mapeamento de características, demonstrando como a entrada é processada por meio de pesos  $w^j$  específicos para cada mapeamento  $j$ .

### 2.2.4 Classificação

O treinamento do modelo é o último passo do aprendizado autodidata. empregamos SVM, *Bagging* (BG) e *Random Forest* (RF) como métodos de classificação primários.

Além disso, incorporamos o KnoraU, uma técnica bem estabelecida usada para selecionar dinamicamente apenas classificadores competentes de uma amostra de teste.

Os algoritmos utilizados foram selecionados devido à diversidade e eficácia demonstradas durante a fase de treinamento do modelo supervisionado e pela capacidade de lidar com diferentes complexidades presentes nos dados. A utilização de máquinas de vetores de suporte (SVM) é motivada pela sua habilidade em encontrar hiperplanos ótimos para separar classes em espaços de alta dimensão, enquanto *Random Forest* (RF) e *Bagging* (BG) proporcionam robustez e estabilidade ao agregar a decisão de múltiplas árvores de decisão. A seleção dinâmica baseada na união Knora (KnoraU) é empregada para otimizar o desempenho em contextos específicos, especialmente quando se trata de árvores de decisão e conjuntos de *Random Forest*.

## SVM

Desenvolvido por Vapnik (2013), o SVM pode ser utilizado em problemas de classificação e regressão. Nos últimos anos, este algoritmo de aprendizagem supervisionada ganhou atenção dos pesquisadores devido ao seu bom desempenho em problemas de reconhecimento de imagem. No SVM, cada instância é representada como um ponto em um espaço  $n$ -dimensional (sendo  $n$  o número de características dos dados) e, em seguida, a classificação é realizada encontrando o hiperplano que separa as classes. A Figura 10 ilustra como o SVM resolve o problema de classificação binária.

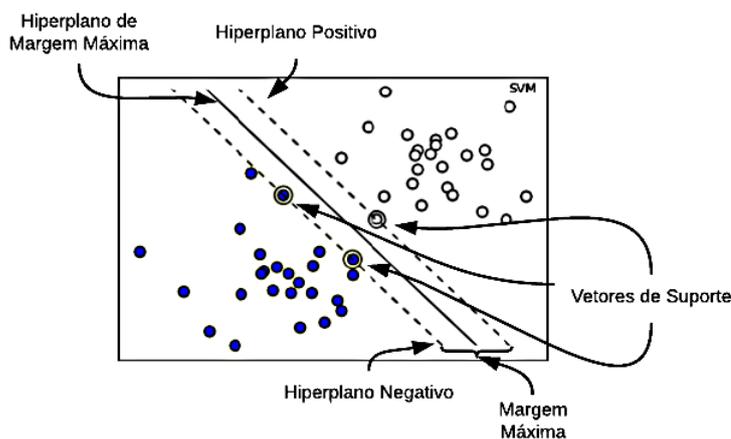


Figura 10 – SVM.

Fonte: Adaptado de Mueller e Massaron (2016).

O classificador de margem máxima é um conceito teórico que ilustra de maneira eficaz o funcionamento do SVM. No SVM, um hiperplano é selecionado para separar os pontos no espaço, visualizável como uma linha em um cenário de duas dimensões. No exemplo da Figura 10 os pontos são completamente separados pelo hiperplano. Na prática, os dados reais geralmente não podem ser separados perfeitamente por um hiperplano.

Assim, a maximização da margem deve ser flexibilizada, permitindo que alguns pontos nos dados de treinamento violem essa linha de separação, ao mesmo tempo que a distância entre os pontos que definem os limites de uma classe e os pontos que definem os limites de outra classe é maximizada. Um parâmetro de ajuste  $C$  é introduzido para controlar a magnitude da tolerância nas dimensões. Quanto maior o valor de  $C$ , mais violações ao hiperplano são permitidas (BROWNLEE, 2016b).

Outro conceito aplicado no SVM é a função de *kernel*. Essa abordagem matemática é empregada para realizar mapeamentos não lineares em dados de dimensões superiores, tornando o SVM capaz de abordar problemas de classificação em espaços de dados de entrada originais com dimensões mais elevadas. Em termos gerais, a função de *kernel* calcula o valor do produto escalar do ponto de dados mapeado no espaço de recursos (KUO et al., 2013). A vantagem do uso destas funções reside no fato de que a complexidade do problema depende exclusivamente da dimensionalidade do espaço de entrada, não da dimensionalidade do espaço de recursos. Diversas funções de *kernel* são amplamente utilizadas, entre elas destacam-se a Linear, Polinomial, de Base Radial e a Sigmóide. Essas funções oferecem flexibilidade e adaptabilidade ao SVM, permitindo lidar eficientemente com desafios em diversos contextos de aprendizado de máquina (CHANDRA; BEDI, 2021).

O SVM foi originalmente proposto para classificação em duas classes distintas, porém, muitas aplicações abrangem o uso de mais de duas classes. Em problemas multiclasse o conjunto de dados é composto por pares  $(x_i, y_i)$  tal que  $y_i \in 1, \dots, k$ , onde  $k > 2$ . Duas abordagens são utilizadas para problemas de classificação multiclasse "um-contra-todos" e "todos-contra-um".

A abordagem "um-contra-todos" gera  $k$  SVMs, onde  $k$  é o número de classes, em cada SVM criado uma classe é fixada como positiva e o restante como negativa. Para prever a classe, basta escolher a saída com valor máximo entre os  $k$  SVMs (WESTON; WATKINS, 1998). A abordagem "todos-contra-um" gera  $k(k - 1)/2$  SVMs, separando cada classe, uma da outra. Cada um dos classificadores fornece uma classe como resultado e o voto dado pela maioria determina o rótulo da classe (FRIEDMAN; FRIEDMAN; FRIEDMAN, 1996). Nesta abordagem, o tamanho dos classificadores gerados é grande, com isso, os resultados levam mais tempo para serem gerados.

## RANDOM FOREST

O algoritmo *Random Forest*, proposto por Breiman (2001), é um método de aprendizado de máquina supervisionado que tem sido amplamente utilizado como um classificador e regressor de uso geral. Ele combina várias árvores de decisão aleatórias por meio de médias, o que tem se mostrado muito eficaz em tarefas onde o número de variáveis é muito maior que o número de observações (BIAU; SCORNET, 2016).

*Random forest* é composto por uma coleção de árvores de decisão estruturadas  $\{h(x, \Theta_k), k = 1, \dots\}$ , onde  $x$  representa o vetor de entrada observado e  $\Theta_k$  são vetores aleatórios independentes e identicamente distribuídos, e cada árvore emite um voto unitário

para a classe mais popular na entrada  $x$ . Esta abordagem de *ensemble* oferece uma maneira eficaz de melhorar a precisão e a robustez dos modelos, tornando o *Random Forest* uma escolha para uma variedade de tarefas de classificação e regressão. (BREIMAN, 2001).

Dado um conjunto de classificadores  $h_1(x), h_2(x), \dots, h_K(x)$  com o conjunto de treinamento aleatoriamente extraído da distribuição do vetor aleatório  $X, Y$ , a função de margem é definida como

$$mg(X, Y) = av_k I(h_k(X) = Y) - \max_{j \neq Y} av_k I(h_k(X) = j) \quad (2.7)$$

Onde  $I(\cdot)$  é a função indicadora. A função de margem mede até que ponto o número médio de votos em  $X, Y$  para a classe certa excede a média de votos para outra classe. Quanto maior a margem, mais confiança na classificação. O erro de generalização é dado por

$$PE^* = P_{X,Y}(mg(X, Y) < 0) \quad (2.8)$$

onde os subscritos  $X, Y$  indicam que a probabilidade está no espaço  $X, Y$ .

No *Random Forest*  $h_k(X) = h(X, \Theta_k)$ . À medida que o número de árvores aumenta as sequências  $\Theta_1 \dots PE^*$  convergem para

$$P_{X,Y}(P_{\Theta}(h(X, \Theta) = Y - \max_{h \neq Y} P_{\Theta}(h(X, \Theta) = j) < 0) \quad (2.9)$$

Esse resultado mostra o motivo que o *Random Forest* não tende a sofrer sobreajuste à medida que mais árvores são adicionadas à coleção, e em vez disso, converge para um valor limite de erro de generalização.

## BAGGING

A técnica conhecida como *Bagging* (*Bootstrap Aggregation*), proposta por Breiman (1996), é frequentemente empregada em algoritmos de classificação e regressão para reduzir a variância associada às previsões e, conseqüentemente, diminuir o erro de previsão dos modelos de aprendizado.

Para se formar um preditor  $\phi(x, L)$  capaz de prever o valor  $y$  a partir de uma entrada  $x$  em um conjunto de dados  $L = \{(x_n, y_n), n = 1, \dots, N\}$  é possível utilizar a técnica de *Bagging* (*Bootstrap Aggregation*). Essa técnica consiste em obter um melhor preditor por meio de uma sequência de conjuntos de dados de treinamento  $\{L_k\}$ , sendo cada um desses conjuntos formado por  $N$  observações independentes selecionadas aleatoriamente da mesma distribuição do conjunto original de treinamento, por meio de amostras de *bootstrap*. Em seguida, é aplicado algum método de previsão a cada conjunto  $\{L_k\}$  e os resultados são combinados, usando votação para classificação ou média para regressão, para obter a previsão geral, reduzindo a variância devido à média. A Figura 11 ilustra a estrutura geral do *Bagging*.

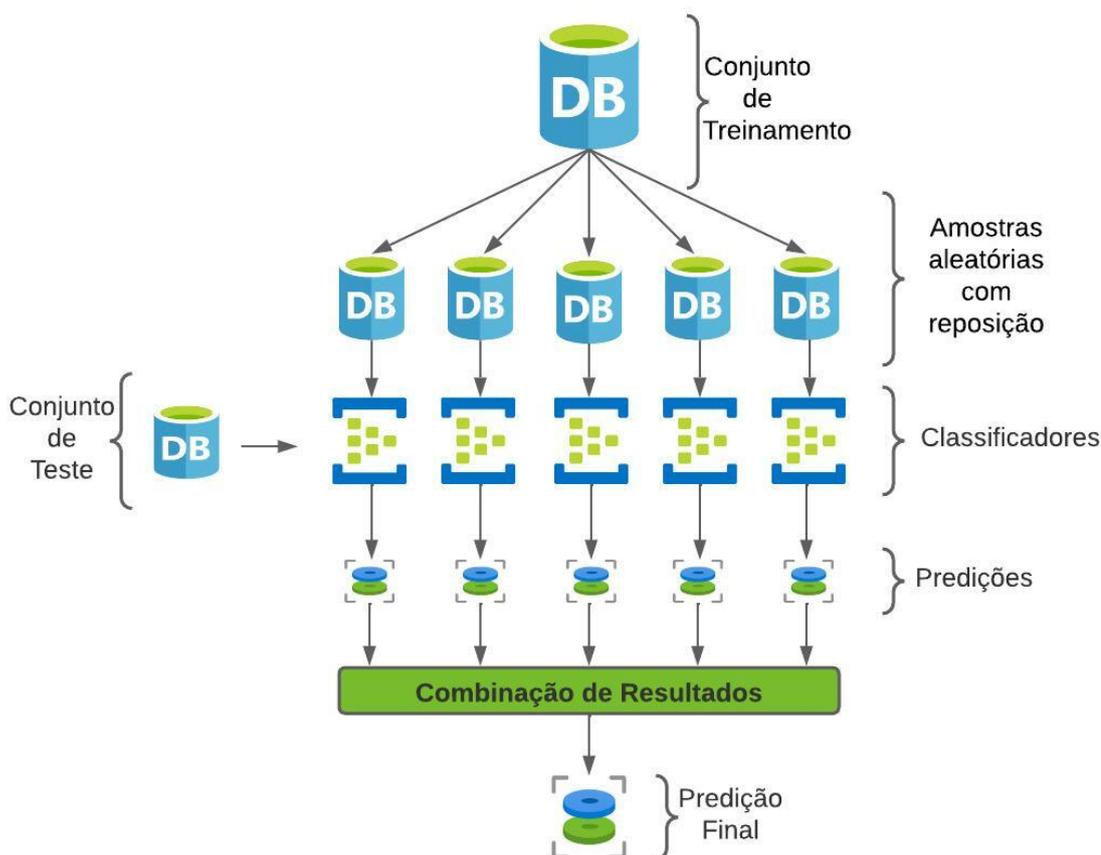


Figura 11 – Estrutura geral do Bagging

Fonte: O Autor

### 2.2.5 Seleção Dinâmica de Classificadores

A técnica conhecida como Seleção Dinâmica (SD) permite a escolha seletiva de um ou mais aprendizes base de um *pool*, em vez de utilizar todos os classificadores disponíveis. A intuição por trás dessa técnica é selecionar apenas os classificadores competentes para a classificação de uma amostra de teste, uma vez que cada classificador base é especialista em uma região local diferente no espaço de características. O objetivo é escolher os melhores classificadores para prever cada amostra de teste (JR; SABOURIN; OLIVEIRA, 2014).

#### KNORA UNION

O método K Oráculos mais próximos (Knora), proposto por Ko, Sabourin e Jr (2008), seleciona um conjunto de classificadores para compor o *ensemble* considerando o resultado da classificação dos  $k$  vizinhos mais próximos da instância de teste no conjunto de validação. Como os classificadores do *pool* que podem reconhecer cada amostra no conjunto de validação são conhecidos, um conjunto de classificadores pode ser selecionado dinamicamente para rotular a amostra de teste fornecida. A combinação padrão desses classificadores é realizada pelo voto majoritário. Knora possui duas principais variações:

Knora *Eliminate* (KnoraE) e Knora *Union* (KnoraU). Neste trabalho, utilizamos a variação KnoraU em nossos experimentos.

No método Knora *Union*, é considerado um conjunto de  $K$  vizinhos mais próximos de uma instância de teste  $t_j$ ,  $1 \leq j \leq K$  e supõe-se que o vizinho mais próximo tenha sido corretamente classificado por um conjunto de classificadores  $C(j)$ ,  $1 \leq j \leq K$ . Cada classificador  $c_i \in C_j$  é solicitado a votar na instância de teste  $t_j$ , podendo um classificador ter mais de um voto se for capaz de classificar corretamente mais de um vizinho. O objetivo é selecionar um conjunto de classificadores competentes para a tarefa de classificação. O Algoritmo 1 apresenta o método KNORA-U.

---

**Algoritmo 1:** Método KnoraU (KO; SABOURIN; JR, 2008)

---

**Entrada:**  $C_c$  - Conjunto de classificadores

$vf$  - meta espaço de características

$x^*$  - Conjunto de teste

**Saída** :  $EoC^*$  - Conjunto de classificadores para amostra  $x^*$

```

1  $\delta$  = exemplos dos  $k$  vizinhos mais próximos da amostra  $x^*$  em  $vf$ 
2 para cada  $\delta_i \in \delta$  faça:
3   para cada  $C_j \in C_c$  faça:
4     se  $C_j$  classifica corretamente  $\delta_i$  então:
5        $EoC^* = EoC^* \uplus C_j$ 
6     fim
7   fim
8 fim
9 retorne  $EoC^*$ 

```

---

## 2.3 Metodologias de Fusão

Os métodos de fusão desempenham são empregados na integração de informações distintas para a realização de tarefas de previsão. O objetivo destes métodos não é confiar exclusivamente em um único esquema de tomada de decisão. Em vez disso, um subconjunto deles, são empregados no processo de tomada de decisões, combinando suas opiniões individuais para chegar a uma decisão consensual. Diversos esquemas de combinação de classificadores foram concebidos, e experimentos demonstraram consistentemente que alguns desses esquemas superam um único classificador considerado o melhor. Essa abordagem de fusão de decisões destaca-se como uma estratégia eficaz para melhorar a robustez e a precisão do processo de tomada de decisões em diversos contextos (KITTLER; HATER; DUIN, 1996).

A aplicação bem-sucedida dessas abordagens tem sido objeto de pesquisa, resultando em uma variedade de estudos. Dois métodos amplamente estudados são a fusão precoce e a tardia. Na fusão precoce, também conhecida como fusão no nível de recurso, os

vetores de características de cada representação são concatenados, formando um vetor expandido. Esse vetor combinado é, então, utilizado para prever a classe por meio de um único classificador. Entretanto, essa técnica pode levar a vetores de características consideravelmente grandes, demandando grandes conjuntos de dados para treinamento e recursos computacionais substanciais. Por outro lado, na fusão tardia, também chamada de fusão no nível de decisão, os classificadores individuais são treinados independentemente para resolver o mesmo problema, como o reconhecimento de expressões faciais neste trabalho. Os resultados individuais de cada classificador são posteriormente combinados (MONEKOSSO; REMAGNINO; KUNO, 2009).

Ao longo das últimas duas décadas, diversos métodos de combinação tardia de classificadores foram propostos na literatura (MOHANDÉS; DERICHE; ALIYU, 2018). Essas técnicas operam com diferentes tipos de saídas dos classificadores, conforme definido por Xu, Krzyzak e Suen (1992):

- **Tipo I (nível abstrato):** a saída do classificador é um rótulo de uma única classe.
- **Tipo II (nível de classificação):** a saída do classificador é uma ordem sequencial de classes candidatas.
- **Tipo III (Nível de medição):** a saída do classificador apresenta valores de confiança atribuídos a cada classe.

Nas combinações de nível abstrato e de classificação, vários esquemas de votação são utilizados na literatura. A vantagem de utilizar as combinações nestes dois níveis é que diferentes características de confiança não tem impacto negativo no resultado, pois a confiança, nestes dois casos, não são utilizadas no processo de decisão. Entretanto, os níveis de confiança atribuídos a uma classe geralmente fornece informações úteis que são desconsideradas na votação. A maioria dos classificadores fornece informações para utilizar o nível de medição, possibilitando a aplicação de métodos que podem explorar os níveis de confiança atribuídos a cada classe, sendo possível utilizar em aplicações práticas (MARINAI; FUJISAWA, 2007).

Nas combinações de nível abstrato e de classificação, a literatura frequentemente emprega esquemas de votação. Embora essas abordagens possam ser eficazes, elas não levam em consideração os níveis de confiança atribuídos a cada classe. Entretanto, as combinações no nível de medição, que incorporam informações de confiança, possibilitam a aplicação de métodos que podem explorar os níveis de confiança atribuídos a cada classe, sendo possível utilizar em aplicações práticas (MARINAI; FUJISAWA, 2007).

### 2.3.1 Regra da Soma

Na fusão por soma, os valores de confiança de cada classificador base são somados e o rótulo é atribuído à classe com a pontuação máxima obtida pela soma. A Equação

2.10 define a combinação pela regra da soma.

$$r^n = \sum_{m=1}^M p_m^n \quad (2.10)$$

onde  $r^n$  é a pontuação combinada da classe,  $p_m^n$  é o valor de confiança obtido da classe  $n$  pelo classificador  $m$  para uma determinada instância.

### 2.3.2 Regra do Produto

Na regra do produto, multiplicam-se os valores de confiança obtidos por cada classificador e atribui o rótulo da classe com a pontuação máxima obtida. A Equação 2.11 define a fusão realizada pela regra do produto.

$$r^n = \prod_{m=1}^M p_m^n \quad (2.11)$$

Esta regra é sensível a valores discrepantes contidos no conjunto de valores de confiança, onde uma pontuação muito baixa ou muito alta tem grande efeito no resultado do produto.

### 2.3.3 Empilhamento

O Empilhamento (do inglês, *Stacking*) é uma técnica para combinar múltiplos classificadores aplicados para um problema específico de classificação (WOLPERT, 1992), e tem sido utilizado com sucesso em diversas aplicações, incluindo os sistemas de melhor desempenho na competição *Netflix* (SILL et al., 2009). O objetivo é melhorar os resultados da classificação sobre um classificador individual. O *Stacking* é dividido em duas partes: o nível-base e o meta-nível. Os classificadores de nível base são treinados com o conjunto de treinamento e geram suas predições, então, o classificador de meta-nível é treinado com os metadados para aprender a mapear as saídas dos classificadores no nível-base para o rótulo da classe real. Durante o processo de classificação de uma nova instância, o classificador de nível-base gera suas predições individuais, estas predições serão consideradas como dados de entrada (metadados) para o classificador de meta-nível gerar a decisão final (AGGARWAL, 2014).

No primeiro passo, ocorre o treinamento de classificadores de nível base utilizando o conjunto original de dados de treinamento. Nesse processo de aprendizado, diversas abordagens podem ser empregadas, como a utilização de classificadores independentes, a aplicação da estratégia de boosting, ou a criação de classificadores homogêneos, ou heterogêneos. No segundo passo é realizado a construção de um novo conjunto de dados, com base na saída dos classificadores base aprendidos na etapa anterior. As previsões de rótulos provenientes desses classificadores base são incorporadas como novas características, preservando os rótulos originais. Adicionalmente, as estimativas de probabilidade dos

**Algoritmo 2:** Algoritmo *Stacking* (WOLPERT, 1992)

---

**Input** : Conjunto de treinamento  $D = \{(x_i, y_i)\}_{i=1}^m$  ( $x_i \in R, y_i \in Y$ )  
**Output** : Um classificador *ensemble*  $H$

- 1 **Passo 1:** Aprender classificador de nível-base
- 2 **para**  $t=1$  to  $T$  **faça:**
- 3 | Aprenda um classificador de nível-base  $h_t$  baseado em  $D$
- 4 **fim**
- 5 **Passo 2:** Construir nova base a partir de  $D$
- 6 **para**  $i=1$  to  $m$  **faça:**
- 7 | Construa uma nova base de dados que contém  $x_i^{new}, y_i$ , onde  
 $x_i^{new} = \{h_1(x_i), h_2(x_i), \dots, h_T(x_i)\}$
- 8 **fim**
- 9 **Passo 3:** Aprender o meta-classificador
- 10 Aprender um novo classificador  $h^{new}$  com base no novo conjunto de dados  
construído
- 11 **retorne**  $H(x) = h^{new}(h_1(x), h_2(x), \dots, h_T(x))$

---

classificadores base podem também ser empregadas como atributos adicionais. Por fim, no último passo, ocorre o treinamento de um classificador de segundo nível, fundamentado no conjunto de dados recém-criado. Qualquer método de aprendizagem pode ser empregado para treinar esse classificador de segundo nível. O empilhamento é a estrutura geral, podendo se conectar com diferentes classificadores e abordagens de aprendizagem para criar características baseadas nos classificadores de primeiro nível, transformando os dados em outro espaço de características (OBAIDAT; ALEXANDROU; SANACORE, 2022).

## 2.4 Considerações Finais

Este capítulo descreve as principais aplicações que se beneficiam do uso de sistemas de reconhecimento de expressão facial e resalta a relevância de cada uma delas. Apresentamos os métodos de aprendizagem usados para desenvolver sistemas de reconhecimento de expressão facial, com ênfase na aprendizagem autodidata. Detalhamos como esse método usa dados de diferentes distribuições para extrair características da base de destino, usando *autoencoders*.

As bases rotuladas, utilizadas para o reconhecimento de expressão facial e as bases não rotuladas, de diferente distribuição, usadas para treinar o algoritmo não supervisionado foram descritas, assim como os métodos de classificação SVM, *bagging*, *random forest* e a técnica de seleção dinâmica KnoraU. Discutimos as técnicas de fusão amplamente empregadas para aprimorar a robustez e fortalecer as decisões em modelos. Apresentamos a fusão precoce e tardia e descrevemos as técnicas de soma, produto e *stacking* que são usadas neste trabalho.

As técnicas descritas neste Capítulo farão parte da composição e avaliação para o

desenvolvimento desta pesquisa. No Capítulo 3 será realizado um estudo do estado da arte em trabalhos relacionados, contribuindo para a análise da efetividade do método proposto.

## 3 Estado da Arte

Vários estudos foram realizados na área do reconhecimento de expressões faciais. Nesta seção, serão discutidos estudos da literatura que se relacionam com os objetivos deste trabalho. Na Seção 3.1, serão discutidos e analisados estudos que empregaram uma variedade de técnicas e abordagens no campo do reconhecimento de expressões faciais. Esses estudos abrangem uma ampla gama de métodos e estratégias, oferecendo uma visão abrangente das abordagens mais recentes e inovadoras utilizadas para capturar e interpretar expressões faciais com precisão e eficiência. Na Seção 3.2, serão explorados estudos que se dedicaram à aplicação de técnicas de fusão de múltiplas fontes de informações em sistemas de reconhecimento de expressões faciais. A análise desses estudos proporcionará uma compreensão mais aprofundada de como a fusão estratégica de informações pode impulsionar o desempenho e a precisão, destacando o papel essencial dessa abordagem no contexto da compreensão e interpretação precisas das expressões faciais humanas. Por fim, na Seção 3.3, serão detalhados estudos que empregaram a técnica de aprendizado autodidata em diversos conjuntos de imagens faciais, além de seu emprego promissor em uma variedade de outras áreas de aplicação. A análise desses estudos proporcionará uma compreensão aprofundada de como o aprendizado autodidata está redefinindo os limites da análise de imagens faciais e revelando seu potencial em inúmeras aplicações práticas, demonstrando assim sua relevância abrangente e crescente em diversos campos.

### 3.1 Sistemas de Reconhecimento de Expressão Facial

Diversos estudos têm sido conduzidos na análise de reconhecimento automático de expressões faciais, devido à importância prática do desenvolvimento destes sistemas. Os pesquisadores estão focados no reconhecimento de expressões faciais para lidar com diferentes condições ambientais, usando a abordagem de aprendizagem profunda em várias aplicações para possibilitar soluções em varreduras de retina, formas de cabeça não frontal, variações de iluminação e reconhecimento de expressões de baixa intensidade. Para identificar as diferentes expressões faciais, é necessário um grande volume de dados para treinar uma arquitetura de rede neural profunda. No entanto, a escassez de dados disponíveis, torna a tarefa desafiadora em termos qualitativos e quantitativos (RATHOUR et al., 2022).

Para conduzir o treinamento de redes neurais, a utilização de uma quantidade substancial de dados é essencial para evitar *overfitting*. Os bancos de dados existentes de expressões faciais não são suficientes para treinar uma rede neural profunda com resultados promissores, como em tarefas de reconhecimento de objetos. As altas variações existentes entre sujeitos como idade, sexo, origem étnica e expressividade, assim como variações de

ambiente, como iluminação e oclusões, são fatores que reforçam a necessidade de redes profundas para lidar com a variabilidade intra-classe existente para aprender representações das expressões faciais (LI; DENG, 2018).

O reconhecimento de expressões faciais é uma tarefa computacionalmente complexa e desafiadora em alcançar altas taxas de reconhecimento usando extração de características convencionais e esquemas de classificação. Jain, Shamsolmoali e Sehdev (2019) propuseram um novo modelo de aprendizado profundo que contém várias camadas de convolução e blocos residuais profundos para reconhecimento de emoções faciais. Este modelo treinou em dois conjuntos de dados, CK (LUCHEY et al., 2011) e JAFFE (LYONS et al., 1998a). Os resultados gerais mostram que o modelo proposto pode superar as recentes abordagens de última geração para reconhecimento de emoções com acurácia de 95,23% para JAFFE e 93,24% para CK+.

Um método de aprendizagem conjunto usando sinais de eletroencefalograma (EEG) é proposto por Li et al. (2022). O método usa uma janela de tempo deslizante para extrair recursos e regularização L1 para selecionar recursos efetivos. Em seguida, aplica um método de seleção de modelos para escolher os melhores submodelos de análise básica e um operador de conjunto para converter os resultados da classificação. Os parâmetros ideais são determinados usando otimização de enxame de partículas de múltiplos objetivos. O método é avaliado em dois conjuntos de dados públicos, DEAP (KOELSTRA et al., 2011) e SEED (ZHENG; ZHU; LU, 2017), usando LOSO no protocolo experimental. As taxas médias de precisão para ativação e valência são 65,70% e 64,22%, respectivamente, no conjunto de dados DEAP, e a precisão média no conjunto de dados SEED é 84,44%.

A TL ganhou popularidade e emergiu como uma área promissora em aprendizado de máquina devido ao seu amplo potencial de aplicações e tem sido extensivamente aplicada em tarefas de visão computacional. Dhankhar (2019) introduziram as arquiteturas ResNet50 e VGG16 para reconhecimento de emoções faciais e propuseram um conjunto que combina os dois modelos. O modelo conjunto superou o SVM (VAPNIK, 2013) de linha de base, bem como as redes individuais ResNet50 e VGG16, alcançando a maior precisão de 75,8%. O SVM teve uma precisão de 37,9%, enquanto o ResNet50 e o VGG16 tiveram taxas de precisão de 73,8% e 71,4%, respectivamente. Também foi explorada a transferência de aprendizado para o reconhecimento de expressões faciais utilizando redes de arquitetura AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2017), VGG e ResNet pré-treinadas. Foi alcançada uma precisão média de 90% nos conjuntos de dados combinados JAFFE (LYONS et al., 1998a) e CK+ (LUCHEY et al., 2011).

Chowdary, Nguyen e Hemanth (2021) discutem o uso de técnicas TL no cenário REF. Redes pré-treinadas como ResNet50 (DHANKHAR, 2019), VGG19 (SIMONYAN; ZISSERMAN, 2014a), Inception V3 (SZEGEDY et al., 2016) e MobileNet (HOWARD et al., 2017) são empregadas. As camadas totalmente conectadas dessas CNNs (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) pré-treinadas são removidas e substituídas por suas pró-

prias camadas totalmente conectadas, adequadas aos requisitos específicos da tarefa. Essas camadas recém-adicionadas são então treinadas para atualizar os pesos. O experimento foi realizado utilizando o conjunto de dados CK+ (LUCEY et al., 2011), resultando em uma precisão média de 96%.

Uma nova estratégia de pipeline que melhora gradualmente a precisão do REF, primeiro treinando a(s) camada(s) densa(s) e depois ajustando cada bloco CNN (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) pré-treinado sucessivamente, foi proposta por Akhand et al. (2021). O sistema REF proposto é testado em oito modelos CNNs pré-treinados usando os conjuntos de dados de imagens faciais KDEF (GOELEVEN et al., 2008) (CALVO; LUNDQVIST, 2008) e JAFFE (LYONS et al., 1998a). O REF é um desafio mesmo para visões frontais e é ainda mais complicado pela diversidade de visões de perfil no conjunto de dados KDEF. O método proposto alcançou precisão notável em ambos os conjuntos de dados com modelos pré-treinados, alcançando uma precisão REF de 96,51% e 99,52% nos conjuntos de dados KDEF e JAFFE, respectivamente, em uma validação cruzada de 10 *folds*.

Uma nova arquitetura (FERRAtt - *Facial Expression Recognition with Attention Net*) que concentra a atenção no rosto, utilizando uma representação espacial gaussiana para o reconhecimento das expressões para sistemas REF foi proposta por Fernandez et al. (2019). A arquitetura proposta é baseada em dois componentes principais: (1) correção e atenção da imagem facial, utilizando uma rede codificador-decodificador e um extrator de recursos convolucional que são multiplicados em pixels para obter um mapa de atenção de recursos e (2) representação incorporada e classificação da expressão facial, propondo uma função de custo que cria uma estrutura gaussiana no espaço de representação. O trabalho realizou experimentos utilizando o protocolo de validação cruzada *Leave-10-subject-out* nas bases BU3DFE (YIN et al., 2006) e CK+ (LUCEY et al., 2011) e obtiveram 82,11% e 90,30% de acurácia respectivamente, mostrando resultados superiores quando comparado com a linha de base PreActResNet18 (HE et al., 2016b).

Liu et al. (2014) desenvolveu um sistema REF com o algoritmo DBN (RANZATO et al., 2011) e utilizou as bases JAFFE (LYONS et al., 1998a) e CK+ (LUCEY et al., 2011) para avaliar o modelo proposto. A estrutura do modelo desenvolvido consiste em dois processos de aprendizado de características: um processo não supervisionado, que aprende representação hierárquica das características dado a imagem de entrada e o processo de fortalecimento supervisionado de características que refina os recursos de maneira supervisionada. O erro da classificação supervisionada é propagado pela rede para contribuir na minimização da função objetivo. Utilizando 8 *folds* e apenas 6 classes da base CK+ (LUCEY et al., 2011), alcançou-se 96,7% de acurácia no modelo. A base de dados JAFFE (LYONS et al., 1998a) foi utilizada para aprimorar o modelo treinado com a base CK+, resultando em 93,0% de acurácia. Através do modelo criado, é possível aprender características mais complexas de imagens faciais, o que conseqüentemente demonstra

resultados promissores.

Para reduzir a quantidade de dados necessários para treinar uma rede neural Hamester, Barros e Wermter (2015) desenvolveram uma arquitetura Multi-canal para o reconhecimento de expressões faciais. O primeiro canal é uma rede neural convolucional, que permite ao modelo extrair e aprimorar os recursos da imagem. O segundo canal da arquitetura proposta é composto de duas camadas convolucionais com uma camada *max pooling* entre elas. A primeira camada do segundo canal é previamente treinada como um *autoencoder* convolucional não supervisionado com a base de dados Kyoto e os pesos desta camada são mantidos durante a fase supervisionada. Os dois canais são conectados por uma camada totalmente conectada e produz um resultado utilizando um classificador de regressão logística. O tamanho da entrada é o mesmo para ambos os canais. A base de dados JAFFE (LYONS et al., 1998a) foi utilizada para avaliar a performance da arquitetura de dois canais. Com o protocolo *leave-one-out* foi obtida uma acurácia de 95,8%. Comparando com o método anteriormente proposto por Barros et al. (2014), que utiliza filtros de *Sobel* aplicados nas imagens, observou-se um aumento de 2,7% ao substituir os filtros pelo segundo canal. A abordagem proposta fornece uma vantagem importante sobre os métodos puramente supervisionados, pois utilizar uma rede pré-treinada na primeira camada de uma arquitetura profunda acelera o tempo geral de treinamento do modelo.

Shao e Qian (2019) propuseram três modelos de CNNs com diferentes arquiteturas. A primeira, uma rede rasa composta por seis módulos de convolução residuais separáveis em profundidade, a segunda é uma CNN de ramificação dupla que extrai recursos LBP (OJALA; PIETIKÄINEN; HARWOOD, 1996; ZAVASCHI et al., 2013) tradicionais e recursos de aprendizado profundo em paralelo. A terceira é uma CNN pré-treinada que é projetada pela técnica de aprendizado de transferência para superar a escassez de amostras de treinamento. Três conjuntos de dados foram utilizados para a avaliação das arquiteturas: CK+ (LUCEY et al., 2011), BU-3DEF (YIN et al., 2006) e FER2013 (GOODFELLOW et al., 2013). O desempenho da CNN pré-treinada foi melhor que as demais arquiteturas comparadas com ganho de 2,43% em CK+, 0,30% em BU-3DEF e 3,14% em FER2013, demonstrando que utilizar redes profundas pré-treinadas tem o potencial de melhorar significativamente o desempenho de sistemas REF e resolver o problema de falta de amostras de treinamento e *overfitting*.

Zeng et al. (2017) utilizaram o *autoencoder* para o reconhecimento de expressão facial. Foi proposta a arquitetura de um *autoencoder* esparso profundo e a classificação na base de dados CK+ (LUCEY et al., 2011) foi avaliada. Foram selecionadas as 4 últimas imagens de cada sequência, mais 327 imagens da expressão neutra, totalizando 1635 imagens selecionadas para avaliação do modelo. As imagens da base foram pré-processadas pelo alinhamento AAM (do inglês, *Active Appearance Model*) (COOTES; EDWARDS; TAYLOR, 2001). Diferentes descritores como HOG (DALAL; TRIGGS,

2005), LBP (OJALA; PIETIKÄINEN; HARWOOD, 1996; ZAVASCHI et al., 2013) e *gray value* foram usados para avaliar o desempenho do modelo proposto e o PCA (JOLLIFFE; CADIMA, 2016) foi aplicado para reduzir a dimensionalidade das características e tornar o treinamento mais prático e eficiente. Finalmente, as características reduzidas pelo PCA são utilizadas como dados de entrada dos *autoencoders*. No treinamento, os dados de entrada são reconstruídos por *autoencoders* esparsos de forma não supervisionada. Posteriormente, de forma supervisionada, as imagens com rótulos são utilizadas para realizar o ajuste fino do modelo. O protocolo LOSO foi utilizado, separando o conjunto de dados em 10 sujeitos, com aproximadamente o mesmo número de imagens, sendo que 9 sujeitos foram usados como conjunto de treinamento e o restante no conjunto de teste. O resultado para a classificação de 7 e 8 classes foi de 95,78% e 86,82% respectivamente, demonstrando que o modelo proposto pode distinguir automaticamente as expressões com alta precisão.

Um método de destaque para a extração de características manuais é a abordagem proposta por Kola e Samayamantula (2021). A estrutura consiste em três etapas e combina duas vantagens: a fusão da transformada *wavelet* e do operador LGC-HD (do inglês, *local gradient coding based on horizontal and diagonal*) (TONG; CHEN; CHENG, 2014) para fornecer características distintivas e robustas, enquanto os valores singulares demonstram uma notável insensibilidade às variações de escala de cinza, decorrentes de interferências de ruído. Posteriormente à extração das características, o método emprega um SVM (VAPNIK, 2013) para identificar e reconhecer expressões faciais nos conjuntos de dados JAFFE (LYONS et al., 1998a) e CK+ (LUCEY et al., 2011). Com o protocolo LOSO, o sistema alcançou taxas de acurácia de 58,2% e 70,6%, respectivamente, para as bases de dados.

Extrair padrões distintos e eficazes de imagens faciais desempenha um papel crucial nos sistemas REF, especialmente considerando a sutil diferença existente entre expressões. Com o intuito de capturar mudanças específicas das expressões, Kartheek, Prasad e Bhukya (2023) introduziram o conceito de *Radial Mesh Pattern* (RMP), uma abordagem resultante da fusão dos métodos *Radial Pattern* e *Mesh Pattern*, proposto para gerar distintos conjuntos de características para diversas regiões da imagem. Para validar a eficácia do método proposto, uma série de experimentos foi conduzida em seis bases de dados de referência, a saber: JAFFE (LYONS et al., 1998a), TFEID (CHEN; YEN, 2007), KDEF (GOELEVELN et al., 2008), CK+ (LUCEY et al., 2011), MUG (AIFANTI; PAPACHRISTOU; DELOPOULOS, 2010) e OULU-CASIA (LYONS et al., 1998b). No decorrer desses experimentos, o protocolo LOSO foi adotado, com exceção da base CK+. Os resultados obtidos revelaram um desempenho notável, com taxas de precisão de 61,64%, 88,22%, 95,00%, 83,80%, 91,67% e 76,18%, respectivamente, para as bases de dados consideradas.

Um novo e robusto descritor de características, denominado *Regional Adaptive Affinitive Patterns* (RADAP), foi proposto por Mandal et al. (2019). O RADAP determina

limiares posicionais adaptativos em sua vizinhança local e codifica características de magnitude multidistância, o que as tornam resilientes às variações intraclasse e flutuações irregulares de iluminação em uma imagem. Além disso, estabelece relações de co-ocorrência entre distâncias através da utilização de operadores lógicos. A eficácia do método foi avaliada em nove conjuntos de dados: CK+ (LUCEY et al., 2011), JAFFE (LYONS et al., 1998a), MUG (AIFANTI; PAPACHRISTOU; DELOPOULOS, 2010), MMI (PANTIC et al., 2005), OULU-CASIA (LYONS et al., 1998b), ISED (HAPPY et al., 2015), DISFA (MAVADATI et al., 2013), AFEW (DHALL et al., 2012), e um banco de dados combinado (CK+, JAFFE, MUG, MMI e GEMEP-FERA). O protocolo LOSO foi adotado em JAFFE, MUG (AIFANTI; PAPACHRISTOU; DELOPOULOS, 2010) e OULU-CASIA (LYONS et al., 1998b), enquanto os demais utilizaram validação cruzada *N-fold*. Os resultados obtidos demonstraram taxas de acurácia de 95,4%, 56,2%, 98,7%, 86,6%, 99,2%, 93,5%, 95,3%, 88,7%, e 94,0% para cada conjunto de dados, respectivamente, evidenciando a eficácia do método proposto quando comparados com outras abordagens de estado da arte.

Um novo descritor de recursos para expressão facial é proposto por Du e Hu (2019) que integra o modelo MRDPL-IVRF (*Manifold Regularization Dictionary Pair Learning* (XIE; HU; YIN, 2018) e *Intra-class Variation Reduced Features* com IOCS (*Iterative Optimization Classification Strategy*). Os recursos em cascata baseados em patches ponderados são primeiramente extraídos usando LBP (OJALA; PIETIKÄINEN; HARWOOD, 1996; ZAVASCHI et al., 2013). O modelo MRDPL-IVRF é então adotado para uma representação robusta e esparsa de imagens faciais e, por fim, com base no IOCS, amostras desnecessárias pertencentes à classe mais improvável são removidas do conjunto de treinamento por iterações, e um conjunto de treinamento mais preciso é obtido para treinar o modelo MRDPL iterativamente. A viabilidade da estrutura proposta foi testada com sucesso nas bases de dados CK (KANADE; COHN; TIAN, 2000), CK+ (LUCEY et al., 2011), CMU-PIE (SIM; BAKER; BSAT, 2001), SEFW (DHALL et al., 2011) e Multi-PIE (GROSS et al., 2010) e obtiveram 91,72%, 86,47%, 98,0%, 41,1% e 89,7% de acurácia respectivamente.

Um novo método de pré-processamento foi proposto por Wu e Lin (2018) para auxiliar o modelo CNN a alcançar uma taxa de precisão mais elevada em aplicações de processamento de imagens faciais, juntamente com três tipos de AFMs (*Adaptive Feature Mapping*) que têm a capacidade de reconfigurar as características de novas amostras que não possuem informações de rótulo. Isso permite corrigir algumas amostras mal classificadas. Em comparação com as arquiteturas concorrentes de aprendizado profundo que utilizaram os mesmos conjuntos de dados de treinamento, a abordagem proposta demonstrou o melhor desempenho ao ser aplicada nas bases de dados CK+ (LUCEY et al., 2011), RaFD (LANGNER et al., 2010) e ADFES (SCHALK et al., 2011). Especificamente, houve um notável aumento na precisão do reconhecimento em aproximadamente 3,01%, 0,49% e 5,33%, respectivamente.

Proposto por [Lee, Baddar e Ro \(2016\)](#), um método para reconhecimento de expressões faciais em vídeos tem a capacidade de selecionar automaticamente a face de expressão de pico em uma sequência de vídeo. O método introduz uma abordagem robusta de extração de características, denominada *Collaborative Expression Representation* (CER), que permite a extração eficaz de características de expressão independentes do sujeito no pico da expressão da face. Isso é alcançado por meio da utilização das diferenças entre o pico da expressão e suas variações intraclasse geradas a partir de imagens de expressões de treinamento. Os resultados obtidos nas bases de dados CK+ ([LUCHEY et al., 2011](#)), MMI ([PANTIC et al., 2005](#)) e NVIE ([WANG et al., 2010](#)) foram de 92,34%, 70,12% e 70,33%, respectivamente. Esses resultados evidenciam a viabilidade prática do método proposto em lidar com vídeos que contenham expressões faciais espontâneas e variações de pose.

Usuários das redes sociais utilizam imagens para expressar suas emoções. O trabalho de [Huang, Wu e Tang \(2019\)](#) utilizou uma CNN profunda para resolver o problema de análise de sentimento de imagem a partir de conteúdo visual. O treinamento da CNN requer um grande número de dados para fornecer um bom desempenho. O trabalho propôs o uso de imagens sintéticas para compor o conjunto de treinamento e aumentar substancialmente o tamanho da base para contornar o problema de dados escassos para o reconhecimento de expressão facial, pois as emoções são subjetivas e várias pessoas precisam fornecer anotações para imagens, isso requer muita mão de obra. Os experimentos conduzidos utilizaram imagens faciais sintéticas, reais e uma mistura de ambas. Os resultados apontaram melhor precisão no uso de imagens faciais reais e mistas, quando comparado ao uso apenas de faces sintéticas.

## 3.2 Combinação em Sistemas REF

Utilizar a abordagem de combinação de classificadores pode reduzir problemas de ajuste excessivo, diminuir o risco de selecionar um único classificador e pode encontrar uma performance melhor ([MOUSAVI; EFTEKHARI, 2015](#)). Segundo [Kittler, Hater e Duin \(1996\)](#), há basicamente dois cenários de combinação de classificadores. No primeiro, todos os classificadores usam o mesmo padrão de representação de entrada, e, no segundo, cada classificador utiliza sua própria representação de entrada, ou seja, a eficácia dos métodos de combinação é conquistada pela diversidade da reamostragem do conjunto de treinamento ou pela variação do conjunto de características.

Um novo método para REF usando um conjunto de CNNs ([KRIZHEVSKY; SUTSKEVER; HINTON, 2012](#)) com fusão baseada em probabilidade foi proposto por [Wen et al. \(2017\)](#). O conjunto é construído variando aleatoriamente parâmetros e arquitetura em torno dos valores ideais para CNN. Cada rede é treinada para gerar uma probabilidade para cada classe, que é então fundida usando fusão baseada em probabilidade. Os experimentos foram conduzidos utilizando 100 CNNs, e a média dos melhores resultados foi alcançada

quando o tamanho do conjunto era 35 para todos os conjuntos de dados. O método foi testado, alcançando 50,70% e 76,05% para conjuntos de dados JAFFE (LYONS et al., 1998a) e CK+ (LUCEY et al., 2011), respectivamente, e superou outros métodos comparados em termos de precisão.

Zavaschi et al. (2013) desenvolveram um novo método de reconhecimento de expressão facial que emprega a combinação de dois conjuntos diferentes de características. Foi criado um conjunto de classificadores SVM (VAPNIK, 2013), utilizando filtro Gabor (LYONS; BUDYNEK; AKAMATSU, 1999; TAN; TRIGGS, 2007) e LBP(OJALA; PIETIKÄINEN; HARWOOD, 1996). Em seguida, um algoritmo genético multiobjetivo foi utilizado para procurar o melhor conjunto de classificadores, usando como função objetivo a minimização da taxa de erro e do tamanho do conjunto. Os experimentos foram conduzidos com as bases de dados JAFFE (LYONS et al., 1998a) e CK+(LUCEY et al., 2011) e os resultados obtidos demonstram que o conjunto de classificadores criado possui melhor desempenho quando comparado a um classificador singular. A melhora nas taxas de reconhecimento foi de 5 a 10% em relação às abordagens com recursos e classificadores únicos.

Ghimire e Lee (2014) utilizaram o *bagging* para desenvolver um trabalho com o algoritmo de máquina de aprendizado extremo (do inglês, *Extreme Learning Machine - ELM*) (HUANG; ZHU; SIEW, 2006), baseado em características HoG (do inglês, *histogram of orientation gradient*) (LOWE, 2004) das bases de dados JAFFE (LYONS et al., 1998a) e CK+ (LUCEY et al., 2011) para desenvolver o sistema REF. Primeiramente, as características HoG foram extraídas das imagens dividindo-as em pequenas células. Posteriormente, o algoritmo *bagging* (BREIMAN, 1996) foi usado para construir *bags* da base de dados de treinamento e cada um foi selecionado para treinar utilizando ELMs separados. Para avaliar o reconhecimento das expressões, características HoG foram fornecidas para cada ELM treinado e os resultados obtidos foram combinados com o voto majoritário. A utilização do *bagging* melhorou significativamente o desempenho do reconhecimento. Na base de dados JAFFE (LYONS et al., 1998a), a acurácia obtida foi de 94,37%, com 7,59% de ganho, quando comparado ao melhor resultado da literatura. Na base Cohn Kanade, a acurácia obtida de 97,30% mostra que possui um resultado promissor no reconhecimento de expressão facial.

Tang, Zhang e Wang (2018) desenvolveram um estudo que combina a fusão de extração de características para o sistema REF. Primeiramente, foi proposta uma rede de fusão geométrica diferencial, do inglês *differential geometric fusion network* (DGFN), que reúne diferentes técnicas de extração de características geométricas das imagens baseadas em psicologia e fisiologia, classificadas por uma rede neural artificial. Posteriormente, foi proposta a utilização de uma rede neural convolucional com 12 camadas para a extração de características, chamada de rede sequencial facial profunda, do inglês *deep-facial-sequential network* (DFSN). As bases de dados utilizadas pelo trabalho foram Cohn Kanade, Oulu-

CASIA (LYONS et al., 1998b) e MMI (VALSTAR; PANTIC, 2010), estas, são compostas de sequências que iniciam com uma expressão neutra e terminam com a expressão facial contida na classe da base. Como entrada para a rede DFSN, é utilizada a primeira, a última e a imagem do meio de cada sequência contida na base. Ao final, as características extraídas de cada uma das imagens são concatenadas e então a classificação é realizada com base na fusão destas características. Por fim, foi feita uma combinação das duas redes anteriormente propostas com o objetivo de aproveitar as vantagens de ambos os modelos e obter um melhor desempenho. Em todas as bases, melhores resultados foram alcançados com a combinação das arquiteturas.

Liu, Zhang e Pan (2016) propuseram um modelo que consiste na fusão de várias sub-redes neurais convolucionais com diferentes estruturas e avaliaram a performance, utilizando o conjunto de dados FER2013 (GOODFELLOW et al., 2013). O melhor resultado de uma sub-rede única alcançado foi de 62,44% e a precisão do modelo de fusão proposto obteve melhor performance, com 65,03%. Wen, Li e Li (2015) também fizeram o uso de redes convolucionais para o sistema REF. O modelo proposto faz o uso do ESN (do inglês, *Echo State Network*), método que possui um reservatório dinâmico com grande número de neurônios esparsamente interconectados com pesos não treináveis, juntamente com uma rede convolucional, chamado de EC-ESN (do inglês, *Ensemble Convolutional Echo State Network*). A rede convolucional é utilizada para extrair as características das imagens e o ESN é utilizado como classificador, ou seja, as características extraídas pela rede neural são utilizadas como entrada para o ESN. A geração de parâmetros aleatórios é realizada para garantir a diversidade no conjunto de classificadores que, no final são classificadas com a utilização simples de voto majoritário. Experimentos realizados com as bases de dados JAFFE (LYONS et al., 1998a) e Cohn Kanade (KANADE; TIAN; COHN, 2000), aplicando o protocolo experimental de pessoa independente, ou seja, aleatoriamente são selecionadas pessoas garantindo que esta não esteja simultaneamente no conjunto de treinamento e teste. Os resultados obtidos mostram um ganho de 9,53% na base JAFFE e 9,89% na Cohn Kanade, mostrando ser promissor o modelo proposto de combinação de classificadores.

Devido à eficácia em tarefas de extração de características e classificação, as CNNs (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) tem sido utilizadas em trabalhos recentes. Explorando a diversidade para criar conjuntos de CNNs, Renda et al. (2019) investigaram diferentes estratégias para induzir diversidade em um conjunto de CNNs aplicadas ao REF. Quatro diferentes estratégias de combinações selecionadas para considerar formas de abordar diversidade na construção de classificadores de base foram consideradas: inicialização aleatória, pré-processamento, pré-treinamento e *bagging* (BREIMAN, 1996). Os resultados no conjunto de dados FER2013 (GOODFELLOW et al., 2013) mostraram que a variação de sementes produziu os melhores resultados de reconhecimento, enquanto as variações no processo de pré-treinamento de suas CNNs alcançaram o melhor desempenho

em tempo de execução.

Na Tabela 3 reunimos as principais características dos trabalhos acima descritos, os quais utilizaram técnicas de fusão em sistemas de reconhecimento facial de imagens.

Tabela 3 – Principais características de estudos que utilizaram fusão em sistemas REF.

Trabalhos	Algoritmo	Bases de Dados	Diversidade	Acurácia	Metodologia de Fusão
Zavaschi et al. (2013)	SVM	JAFFE e CK	Características geométricas e de texturas	96,20% - JAFFE 99,40% - CK	Precoce
Ghimire e Lee (2014)	ELM	JAFFE e CK	<i>Bagging</i>	94,34% - JAFFE 97,30% - CK	Tardia
Tang, Zhang e Wang (2018)	ANN	CK e Oulu-CASIA	Diferentes características geométricas baseadas em psicologia e fisiologia	98,73% - CK+ 87,50% - Oulu CASIA	Precoce
Liu, Zhang e Pan (2016)	CNN	FER2013	Diferentes configurações da rede	65,03%	Precoce
Wen, Li e Li (2015)	CNN	JAFFE e CK+	ESN e arquitetura com parâmetros aleatórios	73,02% - JAFFE 84,62% - CK+	Tardia
Renda et al. (2019)	CNN	FER-2013	Inicialização aleatória, pré-processamento, pré-treinamento e <i>bagging</i>	72,24%	Tardia

### 3.3 Aplicações da Aprendizagem Autodidata

Originalmente, a aprendizagem autodidata utilizava os exemplos fora da distribuição como uma fonte de dados não supervisionados, apresentando efeitos positivos em um cenário onde os dados rotulados eram limitados. Os resultados apresentados por Raina et al. (2007), que utilizou uma codificação superficial esparsa para aprendizagem da representação, apontaram que o ganho relativo da aprendizagem autodidata diminui à medida que o número de exemplos rotulados aumenta.

No entanto, estudos que utilizaram arquiteturas profundas mostraram que tal efeito positivo é alcançado mesmo em um cenário com um grande número de exemplos rotulados. As camadas mais profundas das arquiteturas possuem uma hierarquia de características que podem ser compartilhadas entre tarefas ou variantes da distribuição. Experimentos conduzidos por Bengio et al. (2011) teorizam que tanto a profundidade quanto o pré-treinamento não supervisionado contribuem para o ganho observado.

O grande número de dados rotulados está relacionado às hipóteses estudadas por Erhan et al. (2010). Suas descobertas revelaram que, no aprendizado online em um grande conjunto de dados, a vantagem do viés de aprendizado profundo não desapareceu, e um fenômeno semelhante pode estar ocorrendo na aprendizagem autodidata. O pré-treinamento não supervisionado de uma hierarquia profunda com exemplos fora da distribuição inicializa o modelo na bacia de atração do SGD, o que corresponde a uma melhor generalização. Além disso, as bacias de atração não são descobertas por puro aprendizado supervisionado da inicialização aleatória, e uma quantidade maior de exemplos rotulados não permite que

modelos superficiais ou puramente supervisionados descubram o tipo de melhores bacias associadas a exemplos de aprendizado profundo e fora da distribuição.

Na aprendizagem profunda, a aprendizagem de representação não supervisionada pode ser realizada utilizando diferentes técnicas, como PCA (JOLLIFFE; CADIMA, 2016), ICA (BELL; SEJNOWSKI, 1997), RBM (SMOLENSKY, 1986), e *autoencoders* (LU et al., 2020). Uma vez aprendida a representação, existem várias possibilidades de utilização na rede alvo, como congelar o estado das camadas aprendidas com os dados não rotulados em algum nível da rede e usar apenas o restante das camadas para o ajuste fino, ou usar todas as camadas para o ajuste (KAMATH; LIU; WHITAKER, 2019), incluindo uma camada de ativação *Softmax* no final da rede para transformar as saídas em distribuições de probabilidade (SONG; ROTH, 2017; ZHANG et al., 2019). O pré-treinamento não apenas oferece uma boa inicialização, mas também captura dependências complexas entre os parâmetros. O pré-treinamento não supervisionado pode ser uma forma de regularização cujo efeito não desaparece com mais dados, em comparação com as técnicas de regularização L1/L2.

Recentemente, diversos autores estudam a aprendizagem autodidata em diferentes aplicações. Abaixo descrevemos os trabalhos desenvolvidos que usam bases com imagens faciais e também apresentamos trabalhos que aplicam a técnica autodidata em outras áreas, incluindo áudio (MARKOV; MATSUI, 2012a), texto (HU et al., 2019), imagem (LI; LI; FU, 2017) dados de sensores (HE et al., 2017; RAMAMURTHY et al., 2022) e de tráfego de rede (QURESHI et al., 2020).

### 3.3.1 Aprendizagem Autodidata em Imagens Faciais

Alguns estudos vêm sendo desenvolvidos utilizando a aprendizagem autodidata com imagens faciais. Gan et al. (2014) desenvolveu um estudo de predição de beleza facial utilizando duas bases de dados: a primeira, CAS-PEAL-R1 (GAO et al., 2008), com 210 imagens randomicamente selecionadas, e a segunda, formada por 790 imagens de rostos asiáticos coletadas da internet. Para extrair a representação de características, foi utilizado o método de aprendizagem profunda *Convolutional Restricted Boltzmann Machines* (CRBM) (LEE et al., 2009) com os dados não rotulados da base Kyoto (DOI et al., 2003). Em seguida, as características das bases faciais são extraídas pela rede treinada com os dados não rotulados e, por fim, uma análise de regressão foi realizada com o k-NN (FULLER, 2011), *Multinomial Logistic Regression* (HASTIE; TIBSHIRANI; FRIEDMAN, 2001), *Ridge Regression* (HASTIE; TIBSHIRANI; FRIEDMAN, 2001) e SVM (BRERETON; LLOYD, 2010).

Huang et al. (2012) desenvolveu um trabalho para reconhecimento de imagens faciais utilizando a base de dados LFW. Considerando um par de rostos, determinar se são da mesma pessoa é a aplicação da base LFW (HUANG et al., 2007). Para extrair a representação de características, a CRBM foi utilizada em dois experimentos; um deles

utilizando a base Kyoto (DOI et al., 2003) e outro utilizando a própria base LFW. O SVM (VAPNIK, 2013) linear foi utilizado para classificar as características extraídas. Foi concluído que a utilização da própria base LFW, combinando os escores da primeira e segunda camada, proporciona um resultado melhor.

Long et al. (2012) desenvolveram um trabalho de reconhecimento de expressão facial em vídeo, utilizando a aprendizagem autodidata na base de dados Cohn-Kanade (KANADE; TIAN; COHN, 2000). Para representação de características não supervisionada, foi utilizada a base de dados Hateren's Natural Video (HATEREN; SCHAAF, 1998) e o algoritmo *Independent Component Analysis* (ICA) (BELL; SEJNOWSKI, 1997). Após a extração das características da base CK+ (LUCHEY et al., 2011), o SVM (VAPNIK, 2013) foi empregado para classificar as expressões faciais, resultando em uma média de 80,15% de acurácia.

Bhandari et al. (2018) usaram um *Autoencoder* Esparso para aprender a representação de características, com a base Cifar10, sem a utilização dos rótulos. Posteriormente, transferiu a aprendizagem para um *Autoencoder* Convolutacional, utilizando os pesos aprendidos para inicializar os filtros utilizados na rede convolutacional e extrair as características da base JAFFE (LYONS et al., 1997) e classificar as expressões faciais. A função *Softmax* foi empregada na etapa supervisionada durante o processo de aprendizagem autodidata abordado no estudo. O resultado obtido pela aprendizagem autodidata foi comparado a aprendizagem por transferência, cuja aprendizagem foi realizada diretamente na mesma configuração da rede convolutacional e com a mesma base da autodidata (Cifar10), porém, com a utilização dos rótulos. Os *autoencoders* totalmente conectados ignoram a estrutura das imagens, introduzindo redundância nos parâmetros e forçando as características a serem globais. Neste estudo, foi utilizado o *autoencoder* esparso para aprendizagem de características e os pesos transferidos para uma rede convolutacional, de diferente estrutura. Comparando os resultados obtidos com a aprendizagem por transferência (72,58%), que utilizou a rede convolutacional no treinamento e transferência dos pesos, a aprendizagem autodidata, que obteve 56,45% não provou ser eficiente neste trabalho. Com os experimentos realizados, foi possível concluir que a aprendizagem por transferência tende a superar a aprendizagem autodidata nos casos em que o tamanho dos dados e a rede utilizada são pequenos.

Florea et al. (2016) conduziu um estudo para estimar a intensidade de dor de pacientes. Para isso, utilizou *Spectral Regression* (CAI; HE; HAN, 2007) com a base de dados CK+ (LUCHEY et al., 2011) como base auxiliar não rotulada para aprender a representação das características. Posteriormente, a extração das características da base rotulada UNBC (LUCHEY et al., 2011) foi realizada. Para comparação dos resultados, vários classificadores diferentes foram empregados (RF (BREIMAN, 2001), MLP (TAUD; MAS, 2018) e SVR (AWAD et al., 2015)). A comparação entre a aprendizagem autodidata e os resultados obtidos sem a transferência de aprendizado por dados não rotulados revelou

diferenças pequenas, porém, evidenciou um aumento na capacidade de discriminação ao utilizar a aprendizagem autodidata.

Kanan e Cottrel (2010) conduziu um estudo de classificação de imagens utilizando a aprendizagem autodidata. Dentre as classificações realizadas no estudo, estão objetos, flores e faces. Para a classificação de faces, a base rotulada Aleix e Robert (AR) (MARTINEZ; ROBERT, 1998) foi utilizada, com a extração das características desta base feita por meio da representação extraída do ICA (BELL; SEJNOWSKI, 1997) treinado com a base não rotulada McGill (OLMOS; KINGDOM GARRISON, 2004). A base rotulada inclui classes que, além de classificar algumas expressões faciais, detectam a presença de objetos que as pessoas estão utilizando e a incidência de luzes.

A Tabela 4 reúne as principais características dos estudos que utilizam a aprendizagem autodidata, aplicada em sistemas REF.

Tabela 4 – Principais características de estudos que utilizam AAD com bases de imagens faciais

Estudo	Aplicação	Representação de Características	Classificação/ Regressão	Base Não Rotulada	Base Rotulada
Gan et al. (2014)	Predição de beleza facial	CRBM	KNN, Regressão Logística Multinomial, Regressão de Ridge e SVM.	Kyoto	210 imagens de CAS-PEAL-R1 790 imagens de rostos asiáticos
Huang et al. (2012)	Reconhecimento de imagens faciais	CRBM	SVM- Linear	Kyoto	LFW
Long et al. (2012)	Reconhecimento de expressão facial em vídeo	ICA	SVM	Hateren's	CK+
Bhandari et al. (2018)	Reconhecimento de expressão facial	<i>Autoencoder</i> Esparso	<i>Softmax</i>	Cifar10	JAFFE
Florea et al. (2016)	Sistema de análise facial para predição de dor em pacientes	Spectral Regression	RF, MLP e SVR	CK+	UNBC
Kanan e Cottrel (2010)	Classificação de objetos, flores e faces	ICA	NIMBLE	McGill	AR

### 3.3.2 Aprendizagem Autodidata em Outras Aplicações

Além dos trabalhos que utilizam bases com imagens faciais, diversas outras aplicações foram desenvolvidas utilizando a aprendizagem autodidata.

#### CLASSIFICAÇÃO MUSICAL

Classificação de gênero, identificação do artista, classificação do humor, identificação do cover e extração de melodia são tarefas de recuperação de informações aplicadas em músicas, e há uma grande quantidade deste tipo de dado disponível em rede. Independente da tarefa, para se obter um bom resultado, é essencial a extração de características e a seleção do classificador.

A classificação de gênero musical utilizando a aprendizagem autodidata foi um problema abordado em dois trabalhos, os quais utilizaram a base de dados GTZAN (TZANETAKIS; COOK, 2002) como dados não rotulados e a base ISMIR 2004 (CANO et al., 2006) como dados rotulados. Ambas as bases possuem 10 e 6 classes respectivamente, e apenas um gênero musical é comum entre as bases. Primeiramente, Markov e Matsui (2012a) utilizaram o *Sparse Coding* (OLSHAUSEN; FIELD, 1996) como método de representação não supervisionado e, no trabalho posterior Markov e Matsui (2012b) utilizaram o NMF (LEE; SEUNG, 1999). O classificador utilizado em ambos os estudos foi o SVM (VAPNIK, 2013). O mesmo protocolo experimental foi aplicado nos dois contextos e foi possível concluir que a utilização da aprendizagem autodidata é eficiente quando o conjunto de treinamento é pequeno. À medida que o conjunto de treinamento aumenta, o efeito da aprendizagem autodidata diminui gradualmente, independentemente da quantidade de dados não rotulados utilizados para o treinamento das bases.

Bengio et al. (2011) realizaram experimentos que mostram que em arquiteturas rasas o ganho da aprendizagem autodidata diminui à medida que o número de exemplos rotulados aumenta. Em arquiteturas profundas, os experimentos mostram que, tal efeito positivo é alcançado mesmo em um cenário com grande número de exemplos rotulados. Outra característica relevante da AAD (RAINA et al., 2007) é que as camadas profundas das redes neurais possuem características hierarquicamente distribuídas que podem ser compartilhadas entre tarefas e distribuições de dados.

Grosse et al. (2007) desenvolveram um trabalho de identificação de fala e classificação de gênero musical, utilizando dados rotulados escassos e grande quantidade de dados não rotulados. Para representar as características, foi utilizado o *Shift-Invariant Sparse Coding* (SISC) (MØRUP; SCHMIDT; HANSEN, 2008) na base de dados não rotulada. O estudo menciona que as bases utilizadas para as tarefas não compartilham os mesmos rótulos (TIMIT (ZUE; SENEFF; GLASS, 1990) para fala e pequenos trechos de 6 segundos para os gêneros). Na classificação, o SVM (VAPNIK, 2013), GDA (BAUDAT; ANOUAR, 2000) e *MultiExp* (GROSSE et al., 2007) foram utilizados para comparar os resultados.

### CLASSIFICAÇÃO DE TEXTO

Dependendo da tarefa alvo, a classificação de texto pode encontrar dificuldade em achar dados rotulados. Entretanto, há uma grande quantidade de dados de texto disponíveis na rede em *web sites*, por exemplo. Neste caso, a aprendizagem autodidata pode ser satisfatoriamente aplicada. Wang et al. (2012) desenvolveram um trabalho para fazer a categorização de registro de pensamento de pacientes que apresentam depressão. Coletar e rotular os dados é caro para treinar um modelo de classificação e, na prática, há uma pequena quantidade de dados tanto rotulados quanto não rotulados. Foi utilizado o *Sparse Coding* para extrair a representação de características da base não rotulada Yahoo (UEDA; SAITO, 2002). Posteriormente, o SVM (VAPNIK, 2013) foi aplicado para avaliar o modelo criado com a extração da base rotulada, a qual contém 36 laudos de pacientes

com depressão. Os resultados demonstram um aumento de 4,8% em relação aos estudos comparativos, destacando a eficácia promissora do método proposto.

Huang et al. (2009) utilizaram a tarefa de categorização de texto *web*, propondo desenvolver uma forma mais apropriada para extrair características de alto nível dos dados não rotulados, sob uma supervisão oferecida pelos dados rotulados escassos, contidos no processo. Também foi utilizado o *Sparse Coding* para extrair a representação de características da base não rotulada composta pelas 1000 primeiras páginas retornadas de uma pesquisa realizada no *Google*, utilizando palavras-chave relacionadas. O SVM (VAPNIK, 2013) foi utilizado para avaliar a performance das bases rotuladas com as características transformadas. Os resultados de 91,21% para a base WebKB (CRAVEN et al., 1998), 73,14% e 68,25% para Reuters (LEWIS, 1987), e 69,38% para Ohsumed (HERSH et al., 1994) indicam o potencial do método proposto para aprimorar o desempenho da classificação.

Lee et al. (2009) desenvolveu um trabalho de classificação de texto em dois problemas: um, utilizando valores binários e outro, valores inteiros (contagem de palavras). Para a extração de características, foi utilizado o *Gaussian Sparse Coding* com a base não rotulada Reuters para extrair as características da base rotulada 20 Newsgroups (LANG, 1995). A performance foi avaliada utilizando três classificadores padrões: SVM (VAPNIK, 2013), GDA (BAUDAT; ANOUAR, 2000) e *Kernel Dependency Estimation* (KDE) (WESTON et al., 2002).

Zhu et al. (2013) realizaram um trabalho onde propõem uma nova abordagem de redução de dimensionalidade autodidata que é capaz de transferir a aprendizagem em dados fora da distribuição, abundantemente disponíveis, para dados alvo. Foram conduzidos experimentos com bases de imagens e texto. Para imagens foram utilizadas as bases USPS (base não rotulada) (HULL, 1994), Letter (base utilizada sem rótulo) (SLATE, 1991) and MNIST (LECUN et al., 1998). No domínio de análise de documentos, foram utilizadas as bases Reuters21578, 20 Newsgroups (LANG, 1995) e TDT2 (CIERI et al., 1999). Foi utilizado o *Sparse Coding* para extrair a representação de características dos dados não rotulados e o *K-means* (MACQUEEN et al., 1967) para avaliar o modelo.

Hasanuzzaman, Dias e Way (2017) desenvolveu um trabalho utilizando a estratégia de aprendizagem autodidata para detectar linguagem racista no Twitter a partir da incorporação de palavras que associam informações demográficas (idade, gênero e localização). A construção da incorporação das palavras demográficas foi realizada a partir de dados não rotulados e a classificação da base rotulada feita por SVM (VAPNIK, 2013) linear. A precisão da classificação obtida no conjunto de dados de  $F1 = 76,3\%$  melhora significativamente o desempenho da classificação dos modelos demográficos.

Hu et al. (2019) desenvolveram um algoritmo de aprendizagem autodidata para detecção de comportamento para o risco do uso de drogas. Primeiramente, foram coletados os *tweets* por meio de um filtro com uma variedade de nomes de drogas, termos de gírias e

termos que indicam abuso de drogas. Foram anotados manualmente 1794 *tweets*, utilizados para treinar os classificadores, em seguida, os classificadores são aplicados a um grande número de *tweets* não rotulados para produzir *tweets* rotulados por máquina que são verificados pela plataforma de *crowdsourcing*, com boa precisão, mas a um custo menor, totalizando 5 mil *tweets* rotulados por máquina. Os *tweets* rotulados manualmente e os rotulados por máquina são combinados para formar um conjunto de dados rotulados suficiente e confiável para a detecção de comportamento de risco de abuso de drogas. Entretanto, ainda há uma grande quantidade de *tweets* não rotulados que podem ser aproveitados para melhorar significativamente o modelo em termos de precisão de classificação. O algoritmo de aprendizagem autodidata proposto apresenta o aumento de dados não rotulados recursivo para rotular novos *tweets* utilizando CNN (KRIZHEVSKY; SUTSKEVER; HINTON, 2012) e LSTM (HOCHREITER; SCHMIDHUBER, 1997). Estes *tweets* são gerados aplicando os modelos de aprendizagem profunda previamente treinados a uma amostra aleatória de um grande número de *tweets* não rotulados (3 milhões), onde o conjunto de novos *tweets* tem uma distribuição diferente dos conjuntos de dados de treinamento e teste originais. Os modelos profundos foram criados utilizando a aprendizagem autodidata. Como baseline, foram utilizados o SVM (VAPNIK, 2013), Random Forest (BREIMAN, 2001) e Naive Bayes (LEWIS, 1998). Os resultados experimentais obtidos com os modelos de aprendizagem autodidata profunda alcançaram 86,53%, 88,6% e 86,63%, superando os modelos tradicionais do estado da arte, indicando que ajudou a corrigir o viés nos classificadores causados pela natureza desequilibrada do conjunto de dados de treinamento.

### IMAGENS EM GERAL

Outros trabalhos foram desenvolvidos utilizando a aprendizagem autodidata na classificação de imagens em diferentes categorias. Shu et al. (2017) desenvolveram um trabalho de classificação de rochas com a aprendizagem autodidata, utilizando o *K-means* (MACQUEEN et al., 1967) para aprender características dos dados não rotulados e para a classificação o SVM (VAPNIK, 2013). Comparando com outro método de aprendizagem de características, a AAD (RAINA et al., 2007) não obteve um resultado significativo. A justificativa reside na quantidade insuficiente de imagens disponíveis utilizadas para aprender a representação e para o treinamento do classificador.

Jingjing (2016) desenvolveu um trabalho de reconhecimento de ação humana, com o *autoencoder* para representação de características baseado em imagens não rotuladas e o *Softmax* foi empregado para transformar as saídas em distribuições de probabilidade usadas na classificação. Comparando o método proposto com o PCA (JOLLIFFE; CADIMA, 2016), obtiveram um aumento no desempenho de 3% com a utilização da aprendizagem autodidata.

### BIOMETRIA

Trabalhos na área biométrica mostram que a utilização da aprendizagem autodidata

tem apresentado sucesso em aplicações do mundo real. [Fayyaz et al. \(2015\)](#) propôs um estudo usando aprendizagem autodidata para autenticação de veias de dedos como sistema biométrico. Para aprender a representação de dados não rotulados, foi utilizado o *autoencoder*, e um classificador gaussiano foi treinado com as características extraídas da representação aprendida. Foi utilizada a base SDUMLA-HMT ([YIN; LIU; SUN, 2011](#)). Para o treinamento do *autoencoder*, foram utilizadas todas as imagens dos dedos capturados, exceto o dedo indicador direito. Na avaliação do modelo foram utilizadas 600 imagens do dedo indicador direito. Os resultados foram comparados usando a métrica *Equal Error Rate*, revelando uma melhora significativa (0.70 comparado ao resultado mais próximo, 0.761).

[Hou, Yang e Wang \(2014\)](#) realizou trabalhos para reconhecimento facial heterogêneo (HFR), um problema emergente de classificação na área biométrica. A proposta envolve a aplicação da estratégia autodidata em Análise de Componentes Independentes de Domínio (DiCA) para a aprendizagem da representação não supervisionada, com a seleção aleatória de 367 sujeitos para utilizar como base auxiliar não rotulada. CUFS ([WANG; TANG, 2008](#)) e CASIA NIR-VIS 2.0 ([LI et al., 2013](#)) foram as bases de dados alvo utilizadas, aplicando o classificador vizinho mais próximo (k-NN) ([FULLER, 2011](#)). Experimentos realizados mostraram que o uso de DiCA permite aprender um dicionário independente de domínio para melhorar a representação de imagens (melhorando o desempenho do modelo em 2,95%), introduzindo recursos adicionais de representação de imagens, aliviando a preocupação de quantidades insuficientes de imagens rotuladas para o treinamento do modelo.

### DADOS DE SENSORES

[Ramamurthy et al. \(2022\)](#) implementaram o conceito de aprendizagem autodidata em um *framework* chamado STAR-Lite para a identificação de atividades humanas. Ao projetar representações de dados rotulados que abrangem um conjunto específico de atividades em um espaço de representações construído a partir de dados não rotulados com atividades para além daquelas limitadas pelos rótulos, eles conseguiram reduzir a dependência de dados rotulados. Isso resultou em um aprimoramento na precisão de um sistema de reconhecimento de atividades em adultos mais velhos, nos quais os padrões de atividade podem ser influenciados pelo envelhecimento, que alcançou um aumento de 36% na informação e na marcação.

[He et al. \(2017\)](#) conduziram um estudo utilizando um nariz eletrônico (E-nose) - um dispositivo composto por sensores para análise de alimentos, diagnóstico de doenças e controle de ambiente. Eles aplicaram o E-nose para distinguir infecções em feridas, uma tarefa que necessita de dados rotulados para treinar um modelo eficaz. Para lidar com essa questão, propuseram o uso da aprendizagem autodidata. Os dados não rotulados foram coletados por meio de sensores de gases, enquanto dados rotulados foram obtidos a partir de vinte ratos machos distribuídos em quatro grupos de infecções. Para aprender

a representação dos dados não rotulados, utilizaram um *autoencoder* esparso. Na etapa de classificação, empregaram o RBF (*Radial Basis Function*) (PARK; SANDBERG, 1991) e o PLSDA (*Partial Least Squares Discriminant Analysis*). Experimentos com diferentes tamanhos da camada oculta indicaram que tamanhos excessivamente grandes ou pequenos levaram a resultados inferiores. A restrição da esparsidade imposta no *autoencoder* assemelha-se ao funcionamento do cérebro humano: quando algo vem à mente, apenas um pequeno número de neurônios é estimulado, enquanto os outros são suprimidos. Essa característica é análoga à forma como lidamos com novos conceitos em diferentes áreas.

### TRÁFEGO DE REDE

A transferência de dados segura através da rede se tornou uma tarefa desafiadora com o aumento da utilização da Internet. Qureshi et al. (2020) desenvolveram um trabalho para a tarefa de detecção de intrusões na rede e exploraram o conceito autodidata utilizando uma rede pré-treinada em tarefas relacionadas à regressão, usada para extrair recursos do conjunto de dados NSL-KDD (TAVALLAEE et al., 2009). O desempenho da abordagem baseada na aprendizagem autodidata é comparado com várias técnicas usando dez execuções independentes. Observou-se experimentalmente que os recursos extraídos baseados na aprendizagem autodidata, quando concatenados com o conjunto original, melhoram o desempenho do *autoencoder* esparso.

## 3.4 Considerações Finais

Com os trabalhos apresentados neste Capítulo, podemos observar que diversos autores tem conduzido trabalhos em sistemas REF devido à importância de suas aplicações. O uso da aprendizagem profunda tem sido amplamente empregado nesta área devido às vantagens e desempenho satisfatório que esta abordagem oferece. Diferentes formas de combinação de representações também foram desenvolvidas nesta área e os estudos levantados mostram que o uso desta técnica proporciona o aprimoramento dos sistemas. A aprendizagem autodidata, técnica que impõe significativamente menos restrição para o treinamento do sistema, é uma técnica que tem sido aplicada em diversas áreas com sucesso, incluindo sistemas que utilizam imagens faciais.

Inspirados pela abordagem de aprendizagem autodidata para lidar com a escassez de amostras de treinamento em aplicações de Reconhecimento Facial de Expressões (FER), e com a utilização de *ensembles* para gerar um conjunto diversificado de representações, apresentaremos o algoritmo AER no próximo capítulo.

## 4 Método Proposto

A novidade na abordagem proposta para resolver o problema REF é um algoritmo para gerar um conjunto de representações inspirado em AAD e conceitos de diversidade. Nossa abordagem é chamada de Aprendizado de *Ensemble* de Representações (AER). O AER possui duas características essenciais para promover a diversidade: i) explora diferentes estratégias para gerar diversas arquiteturas AEC; ii) usa uma função de custo personalizada para treinar diferentes AECs, observando diversas representações latentes.

A Figura 12 apresenta uma visão geral da estratégia AAD completa adotada neste trabalho. Na **aprendizagem de representação não supervisionada (Passo 1)**, uma representação de alto nível é aprendida usando dados não rotulados. Esta é a parte em que a nossa proposta é inovadora. A seguir, na **extração de características (Passo 2)**, os vetores de recursos são extraídos dos dados rotulados do domínio alvo usando a representação aprendida na etapa anterior. Finalmente, na **aprendizagem supervisionada (Passo 3)**, os vetores de características extraídos na Passo 2 são usados para treinar um modelo de classificação REF (usando classificadores monolíticos ou baseados em conjuntos).

### 4.1 Aprendizagem de Representação (Passo 1)

Esta etapa aprende, de forma não supervisionada, representações de alto nível. Para aprender a partir de dados não rotulados, dado uma entrada  $x_i^u \in R^d$  o *autoencoder* convolucional, utilizado como método de aprendizado de representação não supervisionado, tenta encontrar padrões (vetores de base)  $B^u = \{b_1^u, b_2^u, \dots, b_k^u\} \in R^{d \times K}$  de modo que cada entrada individual não rotulada  $x_i^u$  pode ser representada usando uma combinação de algumas funções básicas, conforme indicado nas Equações 2.2 e 2.3.

#### 4.1.1 Estratégias para Gerar Diversidade

Podemos encontrar na literatura evidências robustas de que uma maior diversidade está altamente correlacionada com o aumento na precisão do conjunto supervisionado baseado em CNN (RENDA et al., 2019; LIU et al., 2019). Assim, decidimos explorar a diversidade para gerar nosso conjunto de representações não supervisionadas conforme ilustrado na Figura 13. O algoritmo proposto usa um AEC para aprender automaticamente representações significativas a partir de dados brutos, eliminando a necessidade de engenharia manual de recursos. As diferentes estratégias para variar parâmetros específicos do AEC são as seguintes:

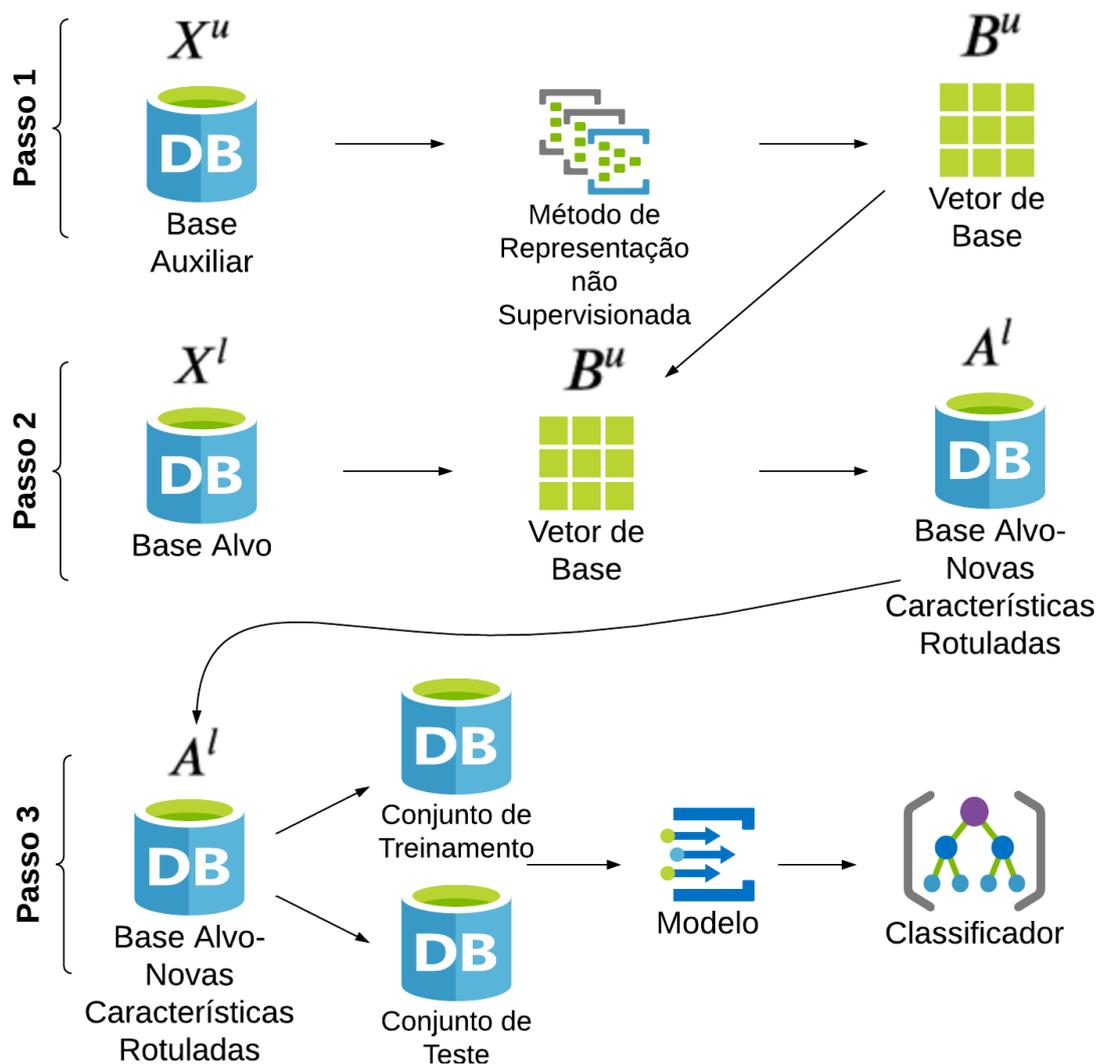


Figura 12 – Etapas de aprendizagem autodidata: (1) Aprendizagem de representação não supervisionada; (2) Extração de características; (3) Aprendizagem supervisionada

Fonte: O autor.

- **Inicialização (semente) Aleatória (S):** aqui, diferentes representações são geradas variando a distribuição de pesos durante o processo de inicialização do AEC. A arquitetura é a mesma de um AEC para outro, mas a semente do processo de inicialização é diferente. A entrada é o número de representações ( $R$ ) geradas com diferentes sementes. O algoritmo seleciona aleatoriamente uma semente em cada AEC gerado no intervalo  $[0, 1000]$ .
- **Arquitetura da Rede (A):** aqui, exploramos a geração de representação usando diferentes arquiteturas AEC. Deve-se definir a profundidade da rede (número de camadas,  $D$ ), os filtros utilizados e a dimensionalidade ( $I$ ) da camada intermediária (latente). O gerador criará a primeira arquitetura com profundidade igual ao número de camadas definidas ( $D$ ), depois a segunda terá ( $D - 1$ ) até a última arquitetura

com profundidade  $D = 1$ , que possui uma camada de entrada, uma intermediária e uma de saída (estrutura básica de um AEC). As representações geradas aqui diferem em profundidade nas arquiteturas AEC, mas usam o mesmo número de neurônios ( $I$ ) na camada latente.

- **Vetor Latente (L)**: nesta estratégia, utilizamos a mesma arquitetura AEC, ou seja, o codificador e o decodificador são os mesmos para todas as representações geradas. A diversidade é obtida variando o número de neurônios na camada intermediária do AEC, denominada camada latente. A entrada é a quantidade de representações ( $R$ ) a serem geradas com dimensionalidade diferente para a camada latente. O algoritmo seleciona aleatoriamente uma dimensionalidade do intervalo  $[150, 2500]$  em cada iteração.

As estratégias mencionadas para gerar diversidade também podem ser combinadas da seguinte forma:

- **SA**: combina a estratégia baseada em sementes (S) e arquitetura AEC (A). As representações geradas por esta combinação possuem variações na inicialização e arquitetura do AEC (número de camadas).
- **SL**: explora a combinação de estratégias de sementes (S) e representação latente (L), onde a variação da inicialização do AEC e da dimensionalidade da camada latente são adotadas para cada representação gerada.
- **LA**: explora o uso da representação latente (L) e da arquitetura AEC (A). Aqui, utilizamos duas abordagens diretamente relacionadas à arquitetura do AEC, variando a dimensionalidade da camada latente e a profundidade do AEC em cada representação gerada.
- **SLA**: na última combinação, consideramos todas as estratégias sementes (S), representação latente (L) e arquitetura AEC (A) para cada representação gerada.

A Figura 13 ilustra a geração de representações por meio do *Autoencoder* Convolutivo (AEC) utilizando as estratégias mencionadas para gerar as representações. Empregadas na inicialização do *autoencoder* convolutivo, as estratégias podem ser utilizadas em uma abordagem única ou na combinação delas.

Além das estratégias S, L e A mencionadas, também propomos a utilização de uma função de custo personalizada. Essa função é introduzida com o objetivo de ampliar ainda mais a diversidade entre as representações geradas.

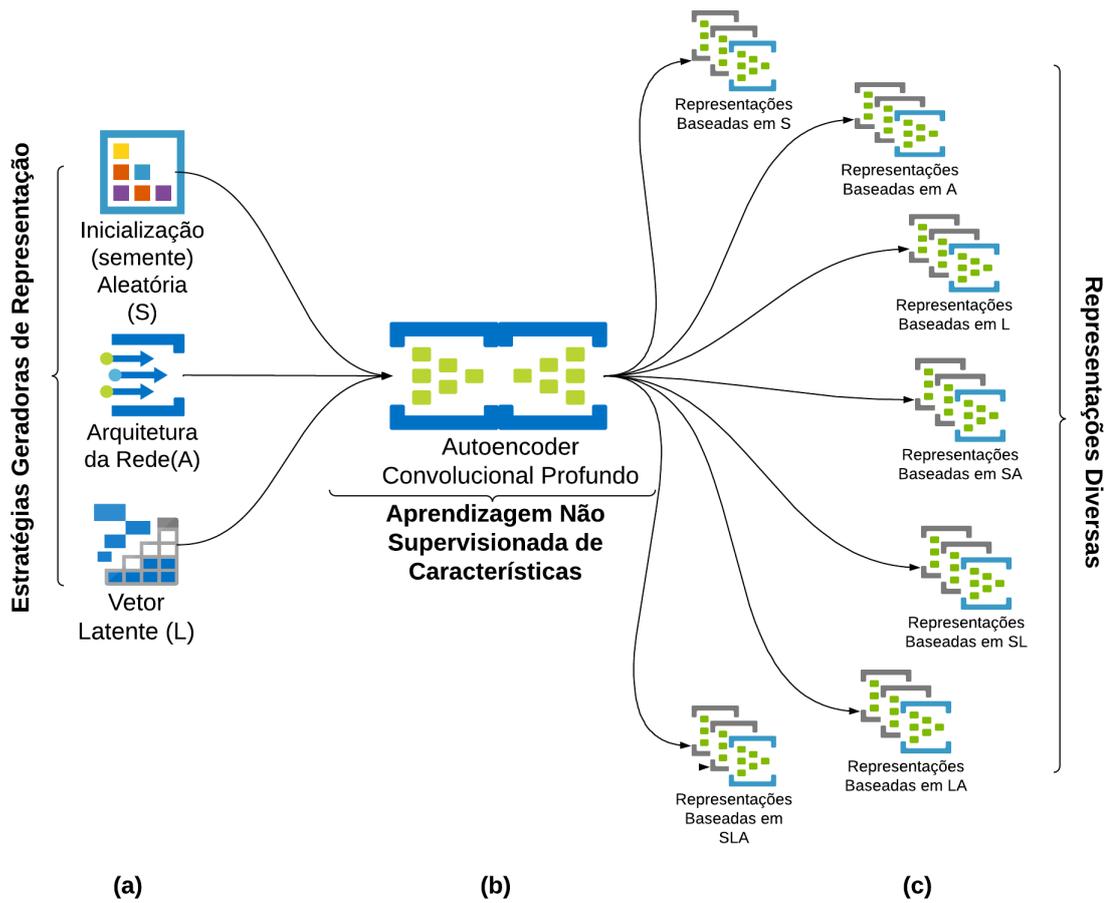


Figura 13 – Gerador automático de estratégia não supervisionada para representação de problemas. (a) Diferentes estratégias geradoras de representações. (b) AEC para aprendizagem de representações de alto nível. (c) Representações geradas com base nas estratégias: sementes (S), arquitetura AEC (A), representação latente (L), sementes + arquitetura AEC (SA), sementes + representação latente (SL), representação latente + arquitetura AEC (LA) e sementes + representação latente + arquitetura AEC (SLA).

#### 4.1.2 Função de Custo Personalizada

A função de custo usada no algoritmo AER contém dois termos: a custo geral (Erro Quadrático Médio, do inglês *Mean Squared Error* -MSE) e um termo de penalidade. O primeiro termo é denotado na Equação 4.1.

$$\mathcal{L}_{\text{MSE}} = \frac{1}{P} \sum_{i=1}^P (X_i - \hat{X}_i)^2 \quad (4.1)$$

onde  $X_i$  são os dados de entrada originais,  $\hat{X}_i$  são os dados de saída reconstruídos pelo AEC e  $P$  é o número total de instâncias de acordo com o tamanho do lote usado para treinar o AEC.

O MSE calcula a média dos quadrados das discrepâncias entre as previsões do modelo ( $\hat{X}_i$ ) e os valores reais ( $X_i$ ) para um conjunto de dados composto por exemplos  $P$ .

Seu objetivo principal é minimizar esse valor, visando tornar as saídas ( $\hat{X}_i$ ) tão semelhantes quanto possíveis às entradas originais ( $X_i$ ). Isto reflete o objetivo central do AEC em aprender uma representação mais concisa e informativa, onde um valor menor indica uma reconstrução mais precisa das entradas.

Porém, o termo MSE por si só não pode garantir a diversidade entre as representações geradas no conjunto. Consequentemente, um termo de penalidade personalizado é adicionado à função de custo após gerar a primeira representação para garantir a diversidade entre as representações. O principal objetivo é minimizar o erro entre as entradas e as saídas do autoencoder, garantindo uma reconstrução precisa. No entanto, além dessa minimização, na função de custo personalizada, introduzimos um componente adicional que busca maximizar a diferença entre as várias representações geradas ao longo do treinamento.

Em termos simples, a função de custo personalizada não apenas se concentra em fazer com que as saídas reconstruídas se aproximem das entradas originais, como é o caso do termo MSE, mas também incorpora uma penalidade que promove a diversidade entre as representações. Esta diversidade evita que o modelo se concentre em soluções redundantes ou similares. Como queremos maximizar a diferença entre as representações latentes de diferentes AECs, o termo de penalidade é adicionado à função de custo como um termo negativo. A Equação 4.2 representa a função de custo personalizada proposta.

$$\mathcal{L}_{\text{MSEP}} = \frac{1}{P} \sum_{i=1}^P (X_i - \hat{X}_i)^2 - \beta \frac{1}{T} \sum_{k=1}^T (\text{PCA}(R_k^{\text{last}}) - \text{PCA}(R_k^{\text{curr}}))^2 \quad (4.2)$$

onde  $\beta$  define o valor da penalidade (a compensação entre ajuste e diversidade),  $R_k^{\text{last}}$  é a saída, ou seja, o vetor latente gerado pelo AEC anterior para a instância  $k_{th}$  de um conjunto de validação composto por instâncias  $T$ . Da mesma forma,  $R_k^{\text{curr}}$  é o vetor latente gerado pelo AEC atual para a mesma instância de validação  $k_{th}$ . A dimensionalidade dos vetores latentes  $R_k^{\text{last}}$  e  $R_k^{\text{curr}}$  pode diferir devido às diferenças nas arquiteturas AEC, de acordo com a estratégia utilizada para gerar diversidade no *pool*. Este é o caso quando variar a dimensionalidade da representação latente é a estratégia utilizada. Assim, aplicamos a análise de componentes principais (PCA) para fazer com que ambos os vetores latentes tenham a mesma dimensionalidade para permitir o cálculo do termo de penalidade na Equação 4.2. Para tanto, consideramos o tamanho do menor vetor latente como o número de componentes do PCA selecionados.

Assim, a função de custo personalizada proposta atua como um mecanismo duplo: minimiza o erro de reconstrução, mantendo a fidelidade nas representações, e simultaneamente maximiza a diferença entre as diferentes representações geradas, promovendo a diversidade desejada. Essa abordagem visa encontrar um equilíbrio entre precisão na reconstrução e variação nas representações, contribuindo para o aprendizado robusto e diversificado do modelo ao longo do treinamento. Quanto maior a similaridade entre os vetores latentes, maior será a penalidade aplicada. A ideia é aumentar a diversidade entre

as representações latentes de um AEC anterior e atual a cada iteração. A necessidade de ajuste do parâmetro  $\beta$  surge da busca por um equilíbrio entre as representações geradas e a diversidade desejada. O ponto ideal é encontrado onde a qualidade das representações produzidas é maximizada e, ao mesmo tempo incentiva a diversidade desejada.

### 4.1.3 Método Aprendizado de *Ensemble* de Representações (AER)

Esta seção apresenta o método AER proposto descrito aqui por três algoritmos. O algoritmo 3 apresenta a função primária responsável por gerar um conjunto de representações não supervisionadas. Os parâmetros de entrada da função AER desempenham um papel fundamental na configuração e execução do método proposto. Em particular, temos  $\psi$  representando o número de representações a serem geradas,  $str$  definindo a estratégia de criação do pool (S, L, A, SL, SA, LA ou SLA),  $X_t^u$  como o conjunto de dados auxiliar de treinamento não supervisionado,  $X_v^u$  como o conjunto de dados de validação não supervisionado adicional, ambos usados para treinar os *autoencoders* criados.

A estratégia de validação durante o treinamento dos *autoencoders* foi aplicada em duas instâncias distintas. Inicialmente, foi empregada para acompanhar o desempenho do modelo em um conjunto de dados não utilizado na prevenção de overfitting e assegurando a generalização apropriada do modelo para dados não observados. Além disso, a abordagem de validação foi incorporada à função de custo para a geração dos vetores latentes tanto da última representação quanto da representação atual, proporcionando um mecanismo de comparação de distinção entre os vetores para direcionar o aprendizado do modelo com o objetivo de gerar diversidade e qualidade entre as representações resultantes.

O parâmetro  $\beta$  está relacionado à função de custo personalizada proposta, definida na Equação 4.2 e usada no Algoritmo 4. É um valor constante que influencia diretamente a magnitude e direção do termo de custo em nossa função com a ideia de penalizar a geração de CAEs semelhantes. Finalmente, a estrutura *params* contém os parâmetros de configuração para uma arquitetura AEC padrão, incluindo o número de camadas, tamanhos de filtro, funções de ativação e outros hiperparâmetros fundamentais. O AEC padrão é usado se *params* for *NULL*.

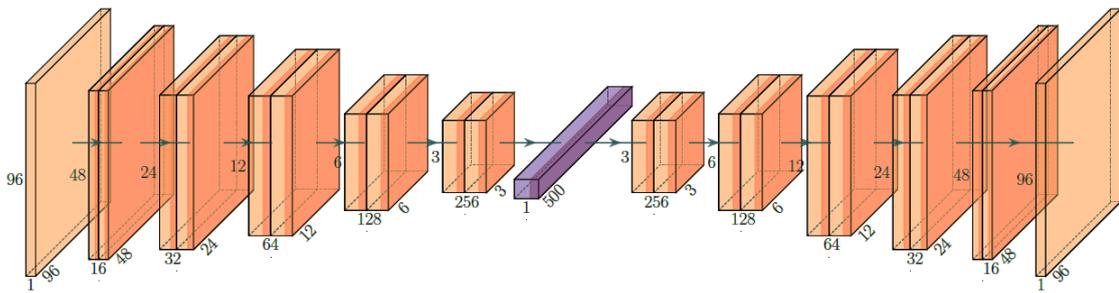


Figura 14 – Arquitetura AEC padrão.

Tabela 5 – Valores de parâmetros padrão para o modelo AEC

Parâmetro	Valor(es)
<i>Stride</i>	2
<i>Activation</i>	RELU
<i>Kernel_Size</i>	3
<i>Filters</i>	[16, 32, 64, 128]
<i>Output_Activation</i>	Linear
<i>Input_Size_Data</i>	[96, 96, 1]
<i>Latent_Size</i>	500
<i>Seed</i>	42
<i>CAE_Depth</i>	5
<i>Custom_Loss</i>	Função de Custo (Eq. 4.2)
<i>Optimizer</i>	SGD
<i>Epoch</i>	20
<i>Batch_Size</i>	60
<i>Padding</i>	Same
<i>Max_random_seed</i>	10000
<i>Min_latent_size</i>	150
<i>Max_latent_size</i>	2500

A Figura 14 mostra a arquitetura AEC padrão usada. Possui camadas de entrada e saída de dimensionalidade  $96 \times 96 \times 1$ , profundidade de 5 e tamanho de vetor latente igual a 500. Porém, dependendo da estratégia usada para gerar o pool, a dimensão do vetor latente e a profundidade AEC pode mudar a cada iteração da função AER, conforme linhas 8-11 e 12-16 do Algoritmo 3, respectivamente. Todos os valores dos parâmetros do AEC padrão são mostrados na Tabela 5. É importante ressaltar que o usuário também pode definir sua arquitetura AEC. Finalmente, o parâmetro de saída  $\rho$  é um conjunto de extratores de recursos treinados em uma abordagem não supervisionada.

A cada iteração do Algoritmo 3, criamos um codificador, um vetor latente e um decodificador para compor um novo AEC (ver linhas 17-22), que é compilado na linha 23 considerando os parâmetros *Optimizer* e *Custom\_Loss* fornecidos. Em seguida, nas linhas 24-27, o AEC criado é treinado. Como pode-se ver na linha 27, a partir da segunda iteração, a previsão de saída ( $R_k^{last}$ ) do AEC anterior estimado no conjunto de dados de validação ( $X_v^u$ ) na linha 28 está disponível para calcular a função de custo personalizada corretamente. Na linha 29, acrescentamos o codificador ao pool como um novo membro, um extrator de recursos.

Detalhamos a função *Custom\_Loss* no Algoritmo 4. Como se pode ver na linha 2 deste algoritmo, na primeira iteração, AER utiliza uma custo regular indicada na Equação 4.1. Porém, para a segunda representação e assim por diante, a função de custo utilizada é a customizada, indicada na Equação 4.2. Ao treinar uma nova representação, inserimos a penalidade com o objetivo de criar representações dissimilares em relação à anterior.

**Algoritmo 3:** Aprendizado de *Ensemble* de Representações (AER)

---

**Entrada:**  $\psi$ , como o número de representações,  
 $X_t^u$ , como base auxiliar de treinamento,  
 $X_v^u$ , como base auxiliar de validação,  
 $params$ , como padrão de parâmetros AEC (Tabela 5),  
 $\beta$ , como uma constante para o termo de penalização,  
 $str$ , como estratégia para variar os parâmetros AEC.

**Saída** :  $\varrho$ , como o *pool* de representações diversas.

1 **Função** AER( $\psi$ ,  $str$ ,  $X_t^u$ ,  $X_v^u$ ,  $\beta$ ,  $params$ ):

2      $max\_depth \leftarrow params.CAE\_Depth$

3      $latent\_size \leftarrow params.Latent\_Size$

4      $seed\_number \leftarrow params.Seed$

5     **para**  $k$  **in**  $[1..\psi]$  **faça:**

6         **se** ( $str$  **in**  $[S, SL, SA, SLA]$ ) **então:**

7              $seed\_number \leftarrow random(range(0 : params.Max\_random\_seed))$

8         **se**  $str$  **in**  $[L, SL, LA, SLA]$  **então:**

9              $ini \leftarrow params.Min\_latent\_size$

10             $end \leftarrow params.Max\_latent\_size$

11             $latent\_size \leftarrow random(range(ini : end))$

12         **se** ( $str$  **in**  $[A, SA, LA, SLA]$ ) **então:**

13             **se** ( $max\_depth = 1$ ) **então:**

14                  $max\_depth \leftarrow params.CAE\_Depth$

15             **senão**

16                  $max\_depth \leftarrow max\_depth - 1$

17         **para**  $i$  **in**  $[1..max\_depth]$  **faça:**

18              $encoder.create\_convolutional\_layer(params)$

19              $encoder.create\_flatten\_layer(params)$

20         **para**  $i$  **in**  $[1..max\_depth]$  **faça:**

21              $decoder.create\_transpose\_layer(params)$

22             AEC.( $encoder, decoder$ ) // Construtor do modelo AEC

23             AEC.compile( $params.Optimizer, params.Custom\_Loss$ )

24         **se**  $k = 1$  **então:**

25              $AEC.fit(params.Epoch, params.Batch\_Size, X_t^u, X_v^u, null)$

26         **senão**

27              $AEC.fit(params.Epoch, params.Batch\_Size, X_t^u, X_v^u, R_k^{last})$

28              $R_k^{last} \leftarrow AEC.encoder.predict(X_v^u)$  // Previsão no conjunto de validação

29              $\varrho.append(AEC.encoder)$

30             delete (AEC)

31     **retorne**  $\varrho$

---

**Algoritmo 4:** Função de Custo Customizada - Equação 4.2

---

**Entrada:**  $\beta$ , como uma constante para o termo de penalização,  
 $X_t^u$ , como base auxiliar de treinamento,  
 $X_v^u$ , como base auxiliar de validação,  
 $R_k^{last}$ , como a saída da camada latente do último AEC,  
 $X_i$ , dados de entrada originais,  
 $\hat{X}_i$ , dados de saída reconstruídos.

**Saída:**  $loss\_value$ , como o valor de custo computado.

```

1 Função Custom_Loss( $\beta$ ,  $X_t^u$ ,  $X_v^u$ ,  $R_k^{last}$ ):
2   se ( $R_k^{last} = null$ ) então:
3     |  $loss\_value \leftarrow \frac{1}{P} \sum_{i=1}^P (X_i - \hat{X}_i)^2$  (Equation 4.1)
4   senão
5     |  $R_k^{curr} \leftarrow AEC.encoder.predict(X_v^u)$ 
6     |  $n\_components \leftarrow \min(\text{length}(R_k^{last}), \text{length}(R_k^{curr}))$ 
7     |  $PCA\_curr \leftarrow PCA(n\_components).fit\_transform(R_k^{last})$ 
8     |  $PCA\_last \leftarrow PCA(n\_components).fit\_transform(R_k^{curr})$ 
9     |  $dif \leftarrow \sum_{k=1}^T (PCA\_last - PCA\_curr)^2$ 
10    |  $penalty\_term \leftarrow \beta dif$ 
11    |  $loss\_value \leftarrow \frac{1}{P} \sum_{i=1}^P (X_i - \hat{X}_i)^2 - penalty\_term$  (Equation 4.2)
12  retorne  $loss\_value$ 

```

---

Na linha 5 do Algoritmo 4, o codificador correspondente à representação atual extrai os recursos do conjunto de validação  $X_v^u$ . Isso é feito usando a função de previsão, e o resultado obtido é armazenado em  $R_k^{curr}$ . Se optarmos pela estratégia do vetor latente para introduzir diversidade nas representações, é fundamental destacar que a dimensionalidade da camada intermediária irá variar entre elas. Portanto, para calcularmos corretamente a diferença entre  $R_k^{curr}$  e  $R_k^{last}$ , é essencial equalizar suas dimensões. Assim, na linha 6, obtém-se o menor valor entre estas dimensões e utiliza-se como o número de componentes a considerar na PCA, conforme indicado nas linhas 7 e 8. Este processo de equalização das dimensões é crucial para garantir a consistência e comparabilidade de representações em todo o algoritmo.

Na linha 9 do Algoritmo 4, calculamos a soma das diferenças quadráticas entre a última representação gerada e a atual. O resultado desse cálculo é então utilizado, na linha 10, como parte do termo de penalidade, multiplicando o valor de  $\beta$ . Este processo é essencial para ajustar o impacto da penalidade na função custo e garantir que a diferença entre as representações seja considerada de acordo com o parâmetro de regularização  $\beta$ . Tal parâmetro de regularização reflete nossa preocupação em gerar representações semelhantes.

## 4.2 Extração de Características (Passo 2)

Após completar o treinamento de todas as representações conforme descrito no Algoritmo 3, passamos para a próxima etapa, chamada "Extração de Características" (Passo

**Algoritmo 5:** Etapa de Extração de Características, Figura 12

---

**Entrada:**  $X^l$  como o banco de dados alvo,  
 $\rho$  como *pool* de extração de características

**Saída :**  $A^l$  como um conjunto de novas características rotuladas do conjunto de dados de destino

1 **Função** `feature_extraction`( $X^l$  ,  $\rho$ ):

2      $A^l \leftarrow []$

3     **para**  $fe$  **in**  $[\rho]$  **faça:**

4          $feature\_set \leftarrow fe.predict(X^l)$

5          $A^l.append(feature\_set)$

6     **retorne**  $A^l$

---

2 da Figura 12), que extrai vetores de características dos dados rotulados  $X^l$  usando os pesos de cada *AEC.encoder* previamente treinado disponível no *pool*  $\rho$ . Detalhamos esse processo no algoritmo 5, que tem o conjunto de dados de destino rotulado  $X^l$  e o conjunto de extratores de recursos  $\rho$  como entrada. A saída é um *pool* de conjuntos de recursos  $A^l$  considerados uma nova maneira de representar  $X^l$ . Essa nova representação do conjunto de dados alvo pode ser posteriormente empregada na construção de classificadores supervisionados.

### 4.3 Treinando um Classificador (Passo 3)

A fase de treinamento do modelo supervisionado representa a etapa final do processo de aprendizagem autodidata. Nesta etapa, avaliamos o desempenho de um conjunto diversificado de classificadores, abrangendo máquinas de vetores de suporte (SVM), *random forest* (RF), *bagging* (BG) com árvore de decisão como classificador base e seleção dinâmica baseada na união Knora (KnoraU) no contexto de árvores de decisão (DT) e conjuntos de RF. Esses classificadores foram examinados para determinar sua eficácia em alavancar as representações geradas anteriormente. As estratégias de fusão avaliadas incluem soma, produto e empilhamento. A configuração dos parâmetros utilizados em nossos experimentos, visando otimizar o desempenho do algoritmo implementado, é apresentada na Tabela 6. Todos os experimentos foram conduzidos com esta mesma configuração, garantindo a reprodutibilidade dos resultados.

/A definição dos parâmetros utilizados nos algoritmos desempenha um papel fundamental para atingir um desempenho eficaz do modelo. No SVM, o *Kernel* Linear foi utilizado para garantir uma abordagem linear na separação entre as classes, considerando que as características contínuas nas expressões faciais podem ser linearmente separáveis. O parâmetro  $C$  foi definido para controlar a regularização, evitando *overfitting* e promovendo uma melhor generalização do modelo. A opção de pesos de classe balanceados (*Class Weight*) é utilizada, pois não há equilíbrio entre a quantidade de imagens disponíveis nas

Tabela 6 – Configurações de parâmetros dos algoritmos usados no Passo 3. Classificadores: único (SVM), conjuntos (BG: *bagging* com árvore de decisão como classificador base, RF: *random forest*), KnoraU (DT: conjunto de árvores; RF: *random forest*).

Algoritmo	Parâmetros	
SVM	<i>Kernel</i>	Linear
	<i>Penalty Parameter C</i>	1e-6
	<i>Class Weight</i>	<i>Balanced</i>
	<i>Probability</i>	<i>True</i>
BG com DT	<i>Max Depth</i>	10
	<i>Tree Max Features</i>	<i>sqrt</i>
	<i>Number of Base Estimators</i>	100
	<i>% of Training Samples</i>	1.0
RF	<i>Max Depth</i>	10
	Número de Árvores	100
	<i>Oob_score</i>	<i>True</i>
KnoraU	Número de Vizinhos	7
	<i>pool_classifiers</i>	<i>Bagging</i> DT RF
Empilhamento	<i>Meta classifier</i>	<i>Logistic Regression</i>
	<i>Solver</i>	<i>lbfgs</i>
	<i>Penalty parameter C</i>	0.1

classes das bases rotuladas, especialmente na CK+. A probabilidade configurada como verdadeira é utilizada para permitir a saída de probabilidades de classe, útil nas estratégias de fusão.

No algoritmo *Bagging*, o parâmetro *Max Depth* foi definido para evitar que as árvores de decisão se tornem muito profundas, o que pode levar ao *overfitting* da rede. *Tree Max Features* foi configurado para permitir uma seleção aleatória de características em cada divisão de nó contribuindo para a diversidade nas árvores. O número de estimadores de base, fixado em 100, foi utilizado para construir um conjunto robusto de árvores de decisão. O parâmetro de porcentagem de amostras de treinamento (*% of Training Samples*) foi fixado em 1.0 para garantir o uso de todas as amostras de treinamento disponíveis, proporcionando assim uma abordagem abrangente durante a construção do modelo.

No *Random Forest*, o parâmetro *Max Depth* limitado em 10 é utilizado para evitar *overfitting* e promover uma generalização eficaz. O número de árvores fixado em 100 é utilizado para construir um conjunto forte e diversificado de árvores de decisão, por fim, *Oob\_score* é configurado como verdadeiro para avaliação da geração do modelo.

Na seleção dinâmica KnoraU, o número de vizinhos é configurado como 7 7 para determinar a quantidade de classificadores que contribuirão para a decisão final. E no pool de classificadores, utilizamos o *bagging* com árvores de decisão (DT) e *random forest* (RF), proporcionando uma combinação de diferentes abordagens.

Na estratégia de fusão de empilhamento, o meta classificador utilizado foi o de

regressão logística para realizar a fusão das previsões dos classificadores base. O parâmetro *Solver* configurado contribui para a convergência rápida durante o treinamento do modelo. Por fim, o parâmetro de regularização  $C$  foi estabelecido em 0.1 reflete uma prioridade na generalização do modelo, visando um desempenho robusto para controlar a regularização do meta classificador.

Além dos parâmetros das configurações dos algoritmos utilizados na classificação, conforme detalhado na Tabela 6, empregamos a técnica de Análise de Componentes Principais (PCA) com 150 componentes. Essa abordagem visa reduzir o tamanho da dimensionalidade dos vetores de características provenientes das diversas representações das bases alvo, as quais foram geradas durante a etapa anterior do processo, ou seja, a extração de características.

Cada representação gerada, para as bases alvo, é classificada e gera uma pontuação de confiança (*score*)  $p$  para cada uma das 7 classes. Os métodos de combinação de classificadores, soma, produto e empilhamento, são utilizados para combinar as probabilidades geradas pelos classificadores das diferentes representações.

Na combinação realizada pela soma, utilizamos a função *decision\_function* da biblioteca *scikit-learn* (PEDREGOSA et al., 2011), publicamente disponível. Para cada instância, o classificador retorna a distância das amostras ao hiperplano de separação. A soma dos valores retornados pelos classificadores é realizada e o maior valor é selecionado como a classe predita pelo classificador.

Na combinação de empilhamento, para cada *fold* de treinamento das bases, os valores retornados de distâncias das amostras ao hiperplano de separação (geradas pela função *decision\_function*) são agrupados, formando novas características de representação da base. Desta forma, a base criada com os valores retornados pelo classificador, mais o rótulo da classe resulta em uma nova forma de representar a base de treinamento. Um classificador de Regressão Logística, então, é utilizado para realizar a aprendizagem supervisionada por meio da nova representação da base, utilizando como conjunto de teste também os valores retornados por cada classificador, porém, descartando o rótulo da classe.

## 4.4 Considerações finais

Com o método apresentado, espera-se criar uma abordagem para gerar múltiplas representações não supervisionadas complementares que possam contribuir para o desempenho de uma solução aplicada ao reconhecimento de expressões faciais.

No capítulo subsequente, será abordado um exame detalhado dos seguintes pontos: (i) Os resultados obtidos ao empregar uma variedade de estratégias para a criação das representações; (ii) A aplicação conjunta das estratégias de geração de diversidade com uma função de custo personalizada; (iii) A apresentação dos resultados provenientes da

---

análise de *Ablation* em relação à função de custo customizada; e (iv) a apresentação de uma análise comparativa entre as soluções baseadas em CNN e os estudos correlatos.

## 5 Resultados Experimentais

Este capítulo descreve o protocolo experimental adotado e apresenta os resultados obtidos durante a condução da pesquisa, seguindo o método proposto no Capítulo 4. Avaliaremos o quão competitivos os resultados obtidos são quando comparados com arquiteturas baseadas em CNNs treinadas em uma abordagem supervisionada e também comparamos nosso método com trabalhos atuais de última geração, que usaram o mesmo protocolo experimental.

Na Seção 5.1, é apresentado o protocolo utilizado para realização dos experimentos apresentados neste trabalho. Na Seção 5.2 são apresentados os experimentos realizados e os resultados obtidos com a utilização das estratégias para gerar diversidade no *pool*. A Seção 5.3 apresenta os resultados dos experimentos realizados adicionando o uso da função de custo personalizada.

### 5.1 Protocolo Experimental

Na etapa de aprendizagem de representação (Passo 1, veja a Seção 4.1), foram utilizadas quatro bases de dados auxiliares distintas: Kyoto (DOI et al., 2003), LFW (HUANG et al., 2007), LabelMe (UETZ; BEHNKE, 2009) e STL-10 (COATES; NG; LEE, 2011). Optamos por explorar o uso de quatro diferentes bases de dados não rotuladas com o objetivo de explorar suas características singulares. Assim, a escolha dessas bases de dados visa enriquecer a análise, explorando a diversidade e complexidade presentes em diferentes domínios, contribuindo para uma compreensão mais aprofundada da aprendizagem de representações. Kyoto é uma base com dados escassos, com imagens que não possuem objetos, mas paisagens naturais. A base LFW não é utilizada para o cenário de expressões faciais, porém, possui imagens inseridas no contexto facial. LabelMe possui maior quantidade de imagens em relação às anteriores, pertencentes às 12 classes de objetos. STL-10 é uma base criada para o uso em aprendizagem autodidata que possui 10 classes e 100 mil imagens não rotuladas.

*Kyoto Natural Images* (DOI et al., 2003) é utilizada no trabalho como base de dados não rotulada de diferente contexto da distribuição alvo para a aprendizagem autodidata. O conjunto de dados contém 62 imagens naturais com resolução 256 x 200 pixels, utilizadas por diversos outros trabalhos que aplicam os conceitos de AAD, devido à sua grande variabilidade. As imagens do nosso protocolo experimental pertencem a um domínio (FER) distante desse conjunto de dados.



Figura 15 – Amostras de imagens da base de dados Kyoto.

Fonte: Doi et al. (2003)

O conjunto de dados LabelMe (UETZ; BEHNKE, 2009) consiste em 50.000 imagens (40.000 para treinamento e 10.000 para teste), e cada imagem tem 256 x 256 pixels de tamanho. As imagens pertencem a uma das 12 classes de objetos: pessoa, carro, edifício, janela, árvore, placa, porta, estante, cadeira, mesa, teclado e cabeça. Para este trabalho, utilizamos apenas a base de testes. A Figura 16 ilustra algumas amostras das imagens contidas na base.



Figura 16 – Amostras de imagens da base de dados LabelMe.

Fonte: Uetz e Behnke (2009).

Para os sistemas de reconhecimento de objetos, esta base de dados é desafiadora, pois as instâncias de cada classe variam muito em aparência, condições de iluminação e ângulos de visão. Além disso, os objetos podem estar parcialmente oclusos ou possuir outros objetos (ou partes deles) na mesma imagem.

*Labeled Faces in the Wild* (LFW) (HUANG et al., 2007) é um conjunto de dados publicamente disponível de fotografias faciais usadas para verificação, também conhecido como correspondência entre pares. O conjunto de dados conta com 13.233 imagens de rostos coletadas na web de 5.749 pessoas. Ao contrário de bases controladas em laboratório, as imagens do LFW capturam uma variedade de condições e contextos da vida cotidiana, refletindo a diversidade e complexidade presentes em ambientes não controlados.

Neste trabalho, utilizamos a base de dados LFW como não rotulada. Diferentemente das bases JAFFE e CK+ que possuem expressões faciais, esta base foi criada para identificar uma determinada pessoa na foto. Apesar de possuir expressões faciais nas imagens de faces, na LFW não há anotação de expressões, desta forma, pode conter diferentes expressões faciais em relação às classes das bases alvo, ou ainda, não possuir imagem de alguma das classes. Neste contexto, temos uma base com imagens semelhantes, porém, não idênticas. A Figura 17 ilustra exemplos da base de dados LFW nas duas versões utilizadas por este trabalho.



Figura 17 – Amostras de imagens da base de dados LFW.

Fonte: [Huang et al. \(2007\)](#).

STL-10 ([COATES; NG; LEE, 2011](#)) é uma base de dados de reconhecimento de imagem utilizada para desenvolver o reconhecimento não supervisionado de características, aprendizagem profunda e aprendizagem autodidata. Cada classe da base possui poucos dados rotulados, porém, um grande conjunto de exemplos não rotulados de distribuição semelhante, mas não idêntica, está disponível. A base possui imagens de 96 x 96 pixels de imagens coloridas.

Os dados rotulados são distribuídos em 10 classes: avião, pássaro, carro, gato, veado, cachorro, cavalo, macaco, navio e caminhão. As bases de treinamento e teste possuem 500 e 800 imagens por classes respectivamente. 100 mil imagens não rotuladas para aprendizado não supervisionado são disponíveis em classes semelhantes, porém mais ampla, com outros tipos de animais e veículos. Estas imagens não rotuladas foram extraídas do ImageNet ([DENG et al., 2009](#)).

STL-10 foi utilizada como base auxiliar não rotulada da aplicação de reconhecimento facial de imagens neste trabalho. Utilizamos a base de treinamento com 5 mil imagens e desconsideramos o rótulo para utilizar no contexto autodidata. A Figura 18 ilustra exemplos das imagens retiradas da base de dados STL-10.



Figura 18 – Amostras de imagens da base de dados STL-10.

Fonte: [Coates, Ng e Lee \(2011\)](#).

As bases auxiliares mencionadas são, originalmente, em formato RGB. No entanto, a arquitetura do autoencoder empregada para o treinamento das representações recebe como entrada imagens em escala de cinza, portanto, as bases foram adaptadas. As imagens foram carregadas usando *OpenCV* e convertidas em escala de cinza usando *cv2.IMREAD\_GRAYSCALE*. Além disso, foi aplicada uma etapa de redimensionamento de 96x96x1 pixels, assegurando a conformidade das imagens com as dimensões necessárias para a entrada na arquitetura do autoencoder.

Os conjuntos de dados alvo são *Japanese Female Facial Expression* (JAFFE) ([LYONS et al., 1998a](#)) e *Extended Cohn-Kanade* (CK+) ([LUCHEY et al., 2010](#)), utilizados nas etapas de construção de características (Passo 2 - 4.2) e no treinamento do classificador (Passo 3 - 4.3). Ambas as bases são amplamente utilizados na literatura científica relacionada à análise de expressões faciais. A diversidade nas características demográficas dos sujeitos presentes nessas bases, incluindo diferentes idades e etnias, contribui para a generalização do modelo. Isso é crucial para garantir que o sistema desenvolvido seja robusto o suficiente para lidar com uma variedade de contextos e características que os sistemas REF possuem.

A base de dados *Japanese Female Facial Expression* (JAFFE) ([LYONS et al., 1998a](#)) consiste em 213 imagens de 10 sujeitos (modelos femininas japonesas), com tamanho de 256x256 pixels. Capturadas em um ambiente controlado de laboratório, as imagens retratam 6 expressões faciais básicas (felicidade, raiva, nojo, medo, tristeza e surpresa), além de uma expressão neutra. A Figura 19 apresenta um recorte das sete classes dessa base provenientes de um mesmo sujeito, evidenciando a diversidade de expressões capturadas. Nesta base, todos os 10 sujeitos possuem uma ou mais imagens para cada uma das classes.



Figura 19 – Amostras da base JAFFE de raiva, nojo, medo, felicidade, neutro, tristeza, e expressões de surpresa.

Fonte: [Lyons et al. \(1998a\)](#).

A base de dados *Extended Cohn-Kanade* (CK+) ([LUCHEY et al., 2010](#)) é uma referência amplamente utilizada para a avaliação de sistemas de reconhecimento de expressão facial. Composta por 523 sequências de 123 sujeitos imagens capturadas em um ambiente controlado de laboratório. Entre essas sequências, 327 imagens de 118 sujeitos são rotuladas com sete expressões faciais básicas (raiva, desprezo, nojo, medo, felicidade, tristeza e surpresa). A Figura 20 ilustra um exemplo de cada uma das sete expressões faciais, cada uma delas com um sujeito diferente, pois nesta base nem todos os sujeitos possuem imagens em todas as classes.



Figura 20 – Amostras da base CK+ de raiva, desprezo, desgosto, medo, feliz, tristeza, surpresa.

Fonte: [Lucey et al. \(2010\)](#).

Antes de extrair as características das imagens faciais, realizamos o pré-processamento recortando a região de interesse (face) e identificando os pontos de referência. Essa etapa é importante para garantir que as imagens estejam alinhadas e prontas para o processo de extração de características. Tanto nas imagens de treinamento quanto nas de teste, aplicamos esse procedimento para assegurar a consistência em todo o conjunto de dados.

A Figura 21 exemplifica o pré-processamento adotado em uma imagem de amostra dos conjuntos de dados JAFFE e CK+. Para detectar e recortar a área facial, empregamos o método de detecção facial Viola-Jones ([VIOLA; JONES, 2004](#)). Este método permite uma segmentação precisa da face na imagem, focando apenas na região de interesse. Além disso, na imagem recortada, identificamos pontos de referência, sendo fundamentais para garantir que as imagens estejam na mesma posição, estabelecendo um referencial comum entre as imagens. Essa padronização é essencial para assegurar que as características faciais sejam comparáveis entre diferentes imagens, eliminando variações na orientação e na posição facial, proporcionando um contexto uniforme.

O protocolo experimental utilizado considera uma validação cruzada LOSO. Este protocolo divide o conjunto de dados para que um sujeito no conjunto de dados de teste

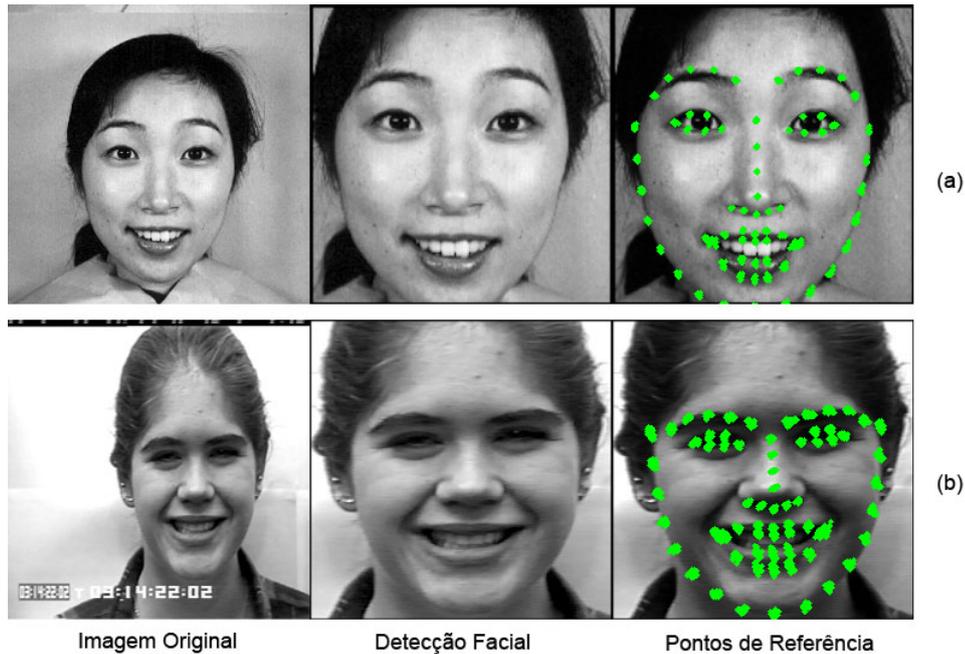


Figura 21 – Exemplo de uma imagem original, detecção de face e extração dos pontos de referências do conjunto de dados JAFFE (a) e conjunto de dados CK+ (b).

não possa estar em treinamento. Suponha que temos um conjunto de dados com  $N$  sujeitos; portanto, para cada *fold*, um sujeito será utilizado como teste e os demais como treinamento. Nesse protocolo, ao usar a seleção de conjunto dinâmico KnoraU, um subconjunto de dados (conjunto de validação) é separado do conjunto de treinamento. Tal conjunto de validação é necessário porque o KnoraU precisa calcular a competência de cada classificador.

## 5.2 Resultados Para o Uso de Estratégias

Um conjunto de cinco experimentos foi realizado. Um para avaliar cada uma das estratégias usadas para gerar as representações não supervisionadas e um último para combinar todos os resultados. Os indutores foram executados usando as implementações da biblioteca scikit-learn Python<sup>1</sup>, Os AECs foram implementados usando Keras<sup>2</sup> com TensorFlow, enquanto o método de seleção de conjunto dinâmico KnoraU foi implementado usando Deslib<sup>3</sup>.

### 5.2.1 Experimento 1 - Variando as inicializações (sementes) aleatórias

As Tabelas 7 e 8 mostram os resultados obtidos com a primeira estratégia considerando representações  $R = 10$  usando diferentes sementes para inicializar os AECs que foram aplicados para representar cada conjunto de dados de destino, JAFFE e CK+,

<sup>1</sup> Disponível em <http://scikit-learn.org/>

<sup>2</sup> Disponível em <https://keras.io/>

<sup>3</sup> Disponível em <https://deslib.readthedocs.io/en/latest/api.html>

Tabela 7 – Acurácia da variação de sementes aleatórias (10 representações) na base de dados JAFFE. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Semente 1	50.66	37.75	36.28	38.16	35.42
Semente 2	29.83	38.84	37.82	40.11	35.58
Semente 3	44.30	33.06	33.87	37.63	33.47
Semente 4	34.27	33.22	31.82	34.53	34.86
Semente 5	49.75	39.40	36.50	38.38	34.60
Semente 6	52.90	40.67	38.54	39.67	38.91
Semente 7	42.34	36.58	32.00	35.90	34.77
Semente 8	48.34	43.64	39.61	40.25	40.50
Semente 9	28.03	32.72	33.80	34.15	34.79
Semente 10	35.33	34.01	37.78	37.30	35.85
Desvio Padrão	9.09	3.71	2.76	2.18	2.15
Soma	<b>52.15</b>	52.88	56.53	56.20	50.12
Produto	51.74	<b>54.35</b>	<b>56.60</b>	56.20	50.12
Empilhamento	48.93	52.54	54.67	<b>55.68</b>	<b>54.79</b>

respectivamente. Os AECs foram treinados aqui usando o conjunto de dados de Kyoto. Podemos ver que a fusão de todas as representações nas últimas três linhas dessas tabelas mostra uma melhoria significativa, independentemente da abordagem de classificação usada (SVM, *Ensembles* ou KnoraU). O melhor resultado para JAFFE foi de 56,60%, observado ao combinar a saída dos RFs treinados em cada representação usando a regra do produto, enquanto o melhor resultado para CK+ foi de 84,39% observado ao combinar a saída dos SVMs treinados em cada representação usando a estratégia de fusão de empilhamento.

### 5.2.2 Experimento 2 - Variando os conjuntos de dados de treinamento

As Tabelas 9 e 10 apresentam os resultados obtidos pelas representações geradas com os quatro diferentes conjuntos de dados auxiliares descritos anteriormente, considerando os conjuntos de dados JAFFE e CK+, respectivamente. Com relação à proximidade do conjunto de dados auxiliar ao domínio alvo, observamos que os conjuntos de dados distantes do domínio alvo apresentaram os melhores resultados em 6 de dez experimentos. Semelhante ao primeiro conjunto de experimentos, podemos ver que a fusão de todas as representações nas três últimas linhas dessas tabelas mostram uma melhoria significativa, independentemente da abordagem de classificação usada. O melhor resultado para JAFFE foi de 61,69%, observado ao combinar a saída dos SVMs treinados em cada representação usando a estratégia de empilhamento, enquanto o melhor resultado para CK+ foi de 86,92% observado usando a mesma abordagem de classificação.

Tabela 8 – Acurácia da variação de sementes aleatórias (10 representações) na base de dados CK+. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Semente 1	84.22	61.01	60.15	65.56	60.90
Semente 2	82.10	57.90	56.35	58.23	58.96
Semente 3	80.40	63.98	59.95	65.42	60.12
Semente 4	84.06	66.51	63.95	66.93	65.53
Semente 5	81.56	63.41	60.15	63.20	61.07
Semente 6	78.09	63.03	62.34	64.97	62.16
Semente 7	80.08	66,86	64.46	66.41	67.00
Semente 8	82.71	64.06	59.80	64.84	59.92
Semente 9	81.10	64.60	62.80	65.12	62.86
Semente 10	80.93	60.50	56.56	59.61	58.26
Desvio Padrão	3.45	2.74	2.78	2.88	2.79
Soma	83.38	71.31	70.83	70.83	66.04
Produto	83.03	71.63	69.77	69.77	66.68
Empilhamento	<b>84.39</b>	<b>77.48</b>	<b>72.64</b>	<b>72.64</b>	<b>70.98</b>

Tabela 9 – Acurácia da variação dos conjuntos de dados não rotulados (4 representações) na base de dados JAFFE. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão de tudo (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Kyoto	60.35	41.08	35.55	43.40	37.57
LFW-Face	59.44	44.22	38.39	41.48	35.70
LFW	58.94	40.06	38.71	37.71	40.12
STL-10	55.21	34.94	35.04	35.51	34.80
Desvio Padrão	2.26	3.85	1.89	3.62	2.35
Soma	59.40	50.27	45.47	51.31	46.92
Produto	59.85	50.33	46.87	51.85	<b>47.42</b>
Empilhamento	<b>61.69</b>	<b>51.72</b>	<b>47.87</b>	<b>53.60</b>	47.08

Tabela 10 – Acurácia da variação dos conjuntos de dados não rotulados (4 representações) na base de dados CK+. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão de tudo (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Kyoto	84.19	64.90	59.95	64.45	61.85
LFW-Face	85.08	63.74	58.99	63.16	60.19
LFW	79.94	65.11	59.01	64.66	59.77
STL-10	83.07	65.84	64.80	66.69	65.72
Desvio Padrão	2.24	0.87	2.77	1.45	2.71
Soma	86.22	69.87	64.26	70.01	65.22
Produto	86.01	69.02	64.83	<b>70.81</b>	65.01
Empilhamento	<b>86.92</b>	<b>70.31</b>	<b>68.89</b>	69.73	<b>69.08</b>

Tabela 11 – Acurácia da variação das arquiteturas de redes (5 representações) na base de dados JAFFE. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão de tudo (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Arq-1	53.41	40.15	40.11	40.15	37.95
Arq-2	52.81	41.09	36.17	44.36	35.45
Arq-3	56.39	37.39	29.88	34.18	36.46
Arq-4	47.53	30.82	36.69	31.79	38.08
Arq-5	50.15	41.69	37.48	34.59	40.25
Desvio Padrão	3.36	4.45	3.77	5.12	1.82
Soma	59.26	<b>51.82</b>	49.95	50.00	50.36
Produto	58.76	51.84	50.53	50.45	49.91
Empilhamento	<b>59.67</b>	50.41	<b>50.96</b>	<b>51.34</b>	<b>53.79</b>

### 5.2.3 Experimento 3 - Variando a Arquitetura AEC

Neste experimento avaliamos o uso de diferentes arquiteturas de AECs. As Tabelas 11 e 12 mostram os resultados dos conjuntos de dados JAFFE e CK+, respectivamente. Também podemos observar uma contribuição significativa ao combinar as representações criadas. O melhor resultado para JAFFE foi de 59,67%, observado ao combinar a saída dos SVMs treinados em cada representação usando a estratégia de empilhamento, enquanto o melhor resultado para CK+ foi de 87,21% observado usando o mesmo classificador base, mas com regra de produto. Para o tamanho da arquitetura ( $D$ ), observamos que os melhores resultados foram obtidos com arquiteturas contendo  $D \geq 3$  camadas (6 em 10 experimentos).

Tabela 12 – Acurácia da variação das arquiteturas de redes (5 representações) na base de dados CK+. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão de tudo (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Arq-1	84.94	69.11	67.07	69.85	67.24
Arq-2	86.56	69.68	61.48	69.54	65.22
Arq-3	85.93	70.14	65.42	69.43	67.11
Arq-4	86.75	70.40	66.63	70.48	65.83
Arq-5	87.55	65.53	66.83	65.53	68.12
Desvio Padrão	0.97	1.98	2.32	1.96	1.16
Soma	87.20	74.12	70.76	73.48	70.31
Produto	<b>87.21</b>	76.94	69.43	<b>74.40</b>	70.31
Empilhamento	86.85	<b>78.20</b>	<b>73.54</b>	<b>74.40</b>	<b>72.83</b>

Tabela 13 – Acurácia da variação do tamanho do vetor latente (10 representações) na base de dados JAFFE. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão de tudo (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
150	58.92	34.29	39.87	32.33	37.91
200	59.85	41.68	36.59	39.24	37.28
250	34.90	34.11	36.03	36.48	34.41
300	59.92	33.53	33.25	38.82	34.24
400	46.57	32.74	31.79	33.60	37.08
500	59.96	42.77	41.64	44.37	41.73
1000	41.46	37.28	32.21	32.86	30.32
1500	50.49	43.54	38.77	44.84	41.59
2000	46.09	37.81	33.27	41.14	34.71
2500	51.33	41.66	44.51	42.14	39.99
Desvio Padrão	8.78	4.18	4.32	4.63	3.61
Soma	61.23	59.52	56.25	<b>59.83</b>	57.64
Produto	60.23	59.46	57.66	59.33	57.64
Empilhamento	<b>62.26</b>	<b>60.53</b>	<b>58.09</b>	59.33	<b>58.86</b>

#### 5.2.4 Experimento 4 - Variando o Tamanho do Vetor Latente

Os resultados da nossa última estratégia singular são mostrados nas Tabelas 13 e 14. A mesma arquitetura de rede é usada com diferentes tamanhos de camada intermediária ( $I$ ). O melhor resultado para JAFFE foi de 62,26%, observado ao combinar a saída dos SVMs treinados em cada representação usando a estratégia de empilhamento, enquanto o melhor resultado para CK+ foi de 86,99% observado usando o mesmo classificador base, mas com regra de produto. Para o tamanho do vetor latente ( $I$ ), observamos que os melhores resultados foram alcançados com  $I \geq 1500$  (7 em 10 experimentos).

Tabela 14 – Acurácia da variação do tamanho do vetor latente (10 representações) na base de dados CK+. Classificadores: único (SVM), *Ensembles* (BG: *Bagging*, RF: *Random Forest*), Seleção Dinâmica: KnoraU (DT: conjunto de árvores; RF: *Random Forest*). Fusão de tudo (soma, produto e empilhamento).

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
150	77.68	63.51	58.47	64.61	59.42
200	84.13	63.10	63.54	63.31	64.19
250	82.28	64.73	61.37	66.49	64.12
300	83.95	63.02	58.88	64.60	60.19
400	84.95	64.78	61.08	64.75	60.70
500	87.06	71.07	63.37	71.52	65.50
1000	86.46	68.82	66.34	67.55	66.76
1500	86.54	67.76	64.15	68.94	65.14
2000	86.96	71.96	65.28	72.59	67.21
2500	85.97	68.33	63.10	67.06	64.98
Desvio Padrão	2.88	3.32	2.58	3.09	2.75
Soma	86.96	75.46	66.97	75.29	67.79
Produto	<b>86.99</b>	76.31	67.25	75.50	67.79
Empilhamento	86.29	<b>78.79</b>	<b>74.30</b>	<b>76.49</b>	<b>72.37</b>

Tabela 15 – Acurácia de *Ensembles* combinando todas as estratégias (as diferentes representações) na base de dados JAFFE.

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Soma	60,63	62,72	61,78	62,68	64,15
Produto	61,10	63,22	61,24	63,61	<b>64,59</b>
Empilhamento	60,33	60,29	62,24	61,22	63,10

### 5.2.5 Combinando Todas as Estratégias

Finalmente, combinamos todas as 29 representações com base em diferentes sementes (10), arquiteturas AEC (5), tamanhos de vetores latentes (10) e conjuntos de dados auxiliares (4). Os resultados para os conjuntos de dados JAFFE e CK+ são mostrados nas Tabelas 15 e 16. Com relação ao conjunto de dados JAFFE, o melhor resultado (64,59% de acurácia) foi obtido ao usar KnoraU combinando a saída de RFs. Para o conjunto de dados CK+, a melhor acurácia foi de 89,22% observada ao combinar a saída dos SVMs usando a estratégia de empilhamento. É bem conhecido na literatura que o sucesso de um *ensemble* ou de um método de seleção dinâmica depende de um *pool* diversificado de classificadores precisos. Assim, pode-se corroborar que as estratégias propostas podem gerar diversas representações (KUNCHEVA; WHITAKER, 2003).

As Tabelas 17 e 18 mostram que a maioria das estratégias propostas se comparam favoravelmente com trabalhos relacionados que representam o estado da arte ao aplicar técnicas não supervisionadas semelhantes para JAFFE e conjuntos de dados CK+. No

Tabela 16 – Acurácia de *Ensembles* combinando todas as estratégias (as diferentes representações) na base de dados CK+.

Repr.	SVM	<i>Ensembles</i>		Seleção Dinâmica (KU)	
		BG	RF	DT	RF
Soma	86,05	74,29	67,14	73,77	67,07
Produto	86,27	75,98	66,93	75,04	67,92
Empilhamento	<b>89,22</b>	77,28	71,14	75,18	77,72

Tabela 17 – Comparação com trabalhos relacionados com base na aprendizagem autodidata

Método	Acurácia
<a href="#">Bhandari et al. (2018)</a>	56,45
Nossa Estratégia (único)	
- SVM	60,35
Nossa Estratégia ( <i>ensembles</i> )	
- Inicialização Aleatória	56,20
- Arquitetura da Rede	59,67
- Diferentes Bases Não Rotuladas	61,69
- Vetor Latente	62,26
- <b>KnoraU - Fusão de Estratégias</b>	<b>64,59</b>

Tabela 18 – Comparação com trabalho relacionado baseado em AAD na base de dados CK+ (área sob a curva ROC).

	Raiva	Desgosto	Medo	Feliz	Tristeza	Surpresa	Desprezo	Média
<a href="#">Long et al. (2012)</a>	77,39	71,08	69,16	89,43	<b>84,78</b>	89,05	—	80,15
SVM Empilhamento (todas as estratégias)	<b>80,00</b>	<b>94,90</b>	<b>80,00</b>	<b>98,60</b>	82,21	<b>96,40</b>	66,70	<b>85,54</b>

caso da JAFFE, nosso melhor resultado (64,59%) melhorou o estado da arte em 8.14 pontos percentuais (de 56,45% para 64,59%). Para CK+, a área sob a curva ROC (AUC) foi calculada para comparar com os resultados relatados em [Long et al. \(2012\)](#). Podemos observar na Tabela 18 que nossos resultados são melhores para a maioria das classes, perdendo apenas para *sad*. Além disso, consideramos as sete classes disponíveis no conjunto de dados CK+ em nossos experimentos.

Neste ponto, podemos responder à primeira questão de pesquisa. Observamos que as melhores abordagens estão diretamente relacionadas à arquitetura AEC. Experimentos mostraram que variar o tamanho do vetor latente (experimentos JAFFE) e o número de camadas AEC (experimentos CK+) foram as estratégias mais promissoras, além disso, podemos dizer que a utilização de um *pool* de representações não supervisionadas pode desempenhar um papel fundamental e benéfico no aprimoramento do processo de classificação final. Ao ser aplicado no contexto da classificação, o *pool* de representações não supervisionadas pode enriquecer a capacidade preditiva do modelo final, aumentando sua habilidade de generalização e, conseqüentemente, melhorando seu desempenho em novos dados não vistos.

### 5.2.6 Discussões

Diferentes estratégias para gerar representações não supervisionadas obtidas com a aprendizagem autodidata para REF foram utilizadas nos experimentos da Seção 5.2. Estas estratégias buscam diversificar as representações ao variar a inicialização, arquitetura e dados de treinamento dos *autoencoders*. Os métodos de aprendizado supervisionado SVM, *Bagging*, *Random Forest* e KnoraU (um método de seleção dinâmica que aprende com as representações AAD não supervisionadas fornecidas), foram avaliados para analisar a eficácia das representações geradas.

Os resultados experimentais nos conjuntos de dados JAFFE e CK+ indicam que as estratégias propostas apresentam desempenho comparável ou superior a trabalhos relacionados que aplicam estratégias de aprendizado de representação não supervisionadas semelhantes. Concluímos que o uso de um *pool* de representações não supervisionadas pode contribuir positivamente para o desempenho de um modelo supervisionado.

Além disso, destacamos que as estratégias mais promissoras identificadas nos experimentos estão diretamente relacionadas à arquitetura do autoencoder. Especificamente, a variação no tamanho do vetor latente e o ajuste do número de camadas do AEC demonstraram desempenho eficaz na geração de representações.

A flexibilidade proporcionada pela modificação do tamanho do vetor latente permite capturar aspectos específicos e características essenciais das imagens. Essa adaptabilidade se revela essencial para a representação eficaz de padrões e informações relevantes no contexto do problema investigado. Adicionalmente, o número de camadas no AEC destaca-se como uma técnica importante na obtenção de representações discriminativas. A variação na profundidade da arquitetura do autoencoder influencia diretamente a complexidade das características aprendidas, contribuindo para uma representação mais abrangente e informativa das imagens de entrada.

## 5.3 Resultados Para o Uso da Função de Custo Customizada Combinada com as Estratégias

Para cada conjunto de dados alvo (JAFFE e CK+), realizamos experimentos considerando os três conjuntos de dados auxiliares (Kyoto, LabelMe e LFW) e conjuntos de representação de diferentes tamanhos (5, 10, 15, 20, 30, 40, 50, 70, 100 e 150). As estratégias S, A, L, SA, SL, LA e SLA foram testadas em cada experimento. Todos os resultados detalhados desses experimentos estão disponíveis para consulta no Apêndice A. É importante notar que a estratégia A foi utilizada exclusivamente na investigação, onde geramos apenas cinco representações, pois explora a profundidade da arquitetura AEC. Assim, utilizar uma profundidade superior a cinco camadas torna o experimento quase impraticável nos recursos computacionais disponíveis para a execução dos experimentos.

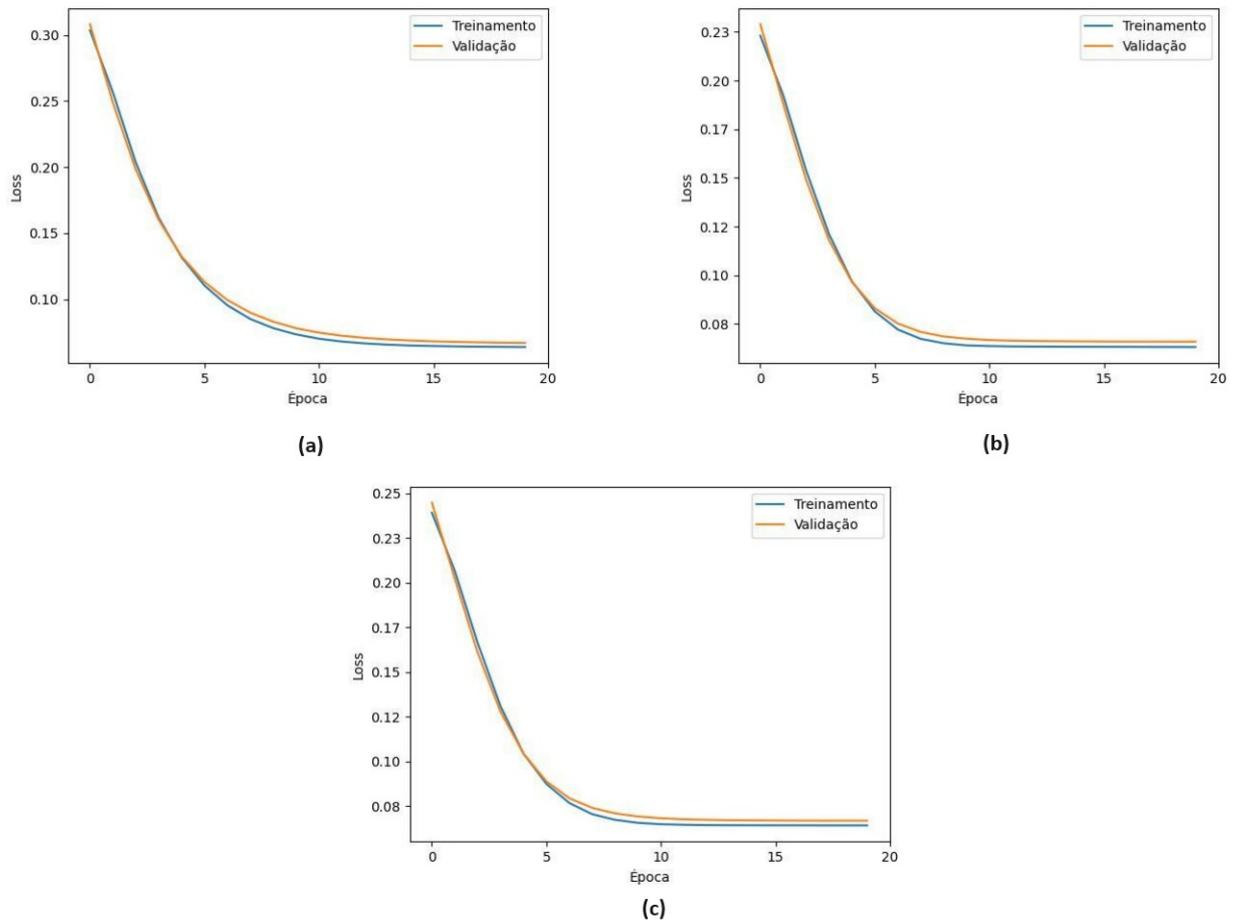


Figura 22 – Curvas de perda de treinamento de três autoencoders distintos.

Para visualizar a evolução dos autoencoders durante os treinamentos, a Figura 22 apresenta o processo de treinamento de três redes distintas. O primeiro refere-se ao treinamento de um autoencoder com profundidade 1 e um vetor latente de dimensão 2.480. O segundo ilustra o treinamento de uma rede com vetor latente de tamanho 2.202 e profundidade 4, enquanto a última imagem representa o treinamento de um autoencoder com vetor latente de dimensão 789 e profundidade 5. Cada um desses modelos foi inicializado com diferentes sementes.

Embora os valores de perda variem entre os autoencoders, podemos visualizar que as curvas de aprendizagem exibem semelhanças no processo de treinamento. Ao longo das épocas, observamos uma diminuição tanto na perda do conjunto de treinamento quanto na perda do conjunto de validação. Esse comportamento sugere que os modelos estão efetivamente aprendendo a representação dos dados de entrada.

É importante ressaltar que, apesar das diferenças nas configurações e nos valores absolutos de perda devido as diferentes técnicas de inicialização utilizadas, todos os autoencoders demonstram a capacidade de generalização. Isso é evidenciado pela diminuição da perda no conjunto de validação, indicando que os modelos não estão apenas memorizando

os dados de treinamento, mas estão conseguindo generalizar eficazmente para dados não vistos. Este padrão é crucial para garantir que os autoencoders estejam capturando características fundamentais dos dados, promovendo uma representação mais robusta.

Por fim, na última etapa de nossa abordagem baseada em AAD, utilizamos os seguintes indutores: SVM, RF, BG e seleção de conjunto dinâmico baseada em KnoraU executado em um conjunto de DTs e RF. Para cada conjunto de representações apresentamos o resultado da melhor representação e sua fusão (soma, produto e empilhamento). Além disso, o desvio padrão de todos os modelos no conjunto é calculado para demonstrar a variabilidade alcançada no conjunto gerado.

### 5.3.1 JAFFE como Base Alvo

As Tabelas 19, 20 e 21 apresentam os resultados para o conjunto de dados JAFFE quando o AEC é treinado com os conjuntos de dados Kyoto, LabelMe e LFW, respectivamente. O melhor resultado para os conjuntos de dados JAFFE foi alcançado com um conjunto de 50 representações treinadas no conjunto de dados de Kyoto, variando a dimensionalidade do vetor latente (estratégia L) e usando a seleção de conjunto dinâmico KnoraU executada em um RF em cada representação. Assim, ao combinar os resultados de cada representação utilizando empilhamento, obtivemos 66,66% de acurácia. O desvio padrão no conjunto de representações correspondente foi de (+/-) 3,33 pontos percentuais. Também é importante notar que a melhor representação única usando KnoraU forneceu 42,00% de acurácia.

A base auxiliar Kyoto apresentou a capacidade de fornecer uma ampla variedade de padrões e características relevantes para a representação facial. A diversidade e riqueza de informações presentes neste conjunto de dados possibilitaram ao AEC aprender representações mais discriminativas, contribuindo para um desempenho superior na tarefa de classificação no conjunto de dados JAFFE.

Ao utilizar o conjunto de dados LabelMe (Tabela 20), o melhor resultado foi alcançado com um conjunto composto por 50 representações variando o tamanho do vetor latente, arquitetura e sementes (SLA de estratégia). A última etapa (supervisionada) utiliza a seleção de conjunto dinâmico KnoraU executada em um RF em cada representação. A melhor representação única forneceu 44,67% de acurácia, enquanto ao combinar os resultados de cada representação usando empilhamento, alcançamos 65,25%.

Ao utilizar o conjunto de dados LFW (Tabela 21), o melhor resultado foi alcançado com um conjunto composto por dez representações. A melhor acurácia (65,25%) foi observada variando as sementes e a arquitetura AEC (estratégia SA). O indutor final é um conjunto de SVMs combinados via regra do produto. O melhor SVM único dentro do conjunto alcançou 61,39% de acurácia.

As Tabelas 19, 20 e 21 também apresentam o desvio padrão de acurácia entre as representações geradas (entre colchetes ao lado do melhor resultado único) que varia de

Tabela 19 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da acurácia do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,50 (3,24)	55,31 (4,24)	40,76 (3,63)	56,67 (4,22)	45,18 (4,12)
	Soma	59,15	61,50	60,09	59,15	58,21
	Produto	59,15	57,27	59,62	59,15	57,27
	Empilhamento	59,62	58,68	43,66	61,97	<b>63,38</b>
L	Melhor Único	60,42 (4,00)	51,37 (4,26)	44,74 (3,94)	51,89 (4,30)	42,00 (3,33)
	Soma	60,09	61,03	60,56	61,03	62,44
	Produto	60,09	61,97	60,56	61,03	62,91
	Empilhamento	61,03	60,56	48,82	61,97	<b>66,66</b>
SA	Melhor Único	61,19 (3,19)	49,54 (4,05)	44,45 (3,60)	50,47 (4,14)	42,71 (3,57)
	Soma	61,97	61,32	62,91	59,62	61,50
	Produto	61,97	60,56	<b>64,78</b>	60,09	61,97
	Empilhamento	60,56	59,62	53,05	60,56	63,84
SL	Melhor Único	61,47 (3,41)	52,63 (4,40)	43,50 (3,74)	51,72 (4,21)	44,38 (4,40)
	Soma	61,03	61,97	57,74	61,03	60,09
	Produto	61,50	60,56	57,27	62,91	60,09
	Empilhamento	62,44	59,62	43,66	62,91	<b>63,84</b>
LA	Melhor Único	60,61 (3,38)	50,05 (3,38)	47,23 (4,45)	52,88 (3,52)	48,14 (4,90)
	Soma	60,56	61,03	61,97	61,03	<b>62,91</b>
	Produto	60,56	60,09	62,44	61,50	<b>62,91</b>
	Empilhamento	60,03	61,97	51,17	61,50	61,50
SLA	Melhor Único	61,35 (3,37)	51,62 (4,36)	44,46 (4,36)	52,94 (4,11)	47,41 (4,20)
	Soma	61,50	62,91	59,15	<b>63,38</b>	59,15
	Produto	61,50	61,97	59,62	61,97	59,62
	Empilhamento	61,50	58,68	48,35	60,09	62,44

Tabela 20 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da acurácia do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	57,08 (3,32)	50,53 (4,19)	44,30 (5,23)	50,79 (4,52)	42,44 (4,77)
	Soma	55,86	61,97	59,15	61,03	61,03
	Produto	57,27	61,03	59,62	62,44	60,56
	Empilhamento	56,33	59,15	50,70	62,44	<b>62,44</b>
L	Melhor Único	58,96 (5,56)	48,66 (4,36)	42,62 (3,84)	49,58 (4,30)	43,33 (3,59)
	Soma	59,15	56,33	58,21	56,33	57,27
	Produto	58,68	56,33	58,21	55,86	56,80
	Empilhamento	58,68	55,86	42,25	58,21	<b>60,56</b>
SA	Melhor Único	61,36 (3,20)	49,58 (3,96)	43,97 (4,72)	49,06 (3,94)	44,30 (4,24)
	Soma	60,56	60,56	62,91	60,09	63,84
	Produto	61,50	58,68	62,91	60,56	63,38
	Empilhamento	58,68	62,44	53,52	59,62	<b>64,78</b>
SL	Melhor Único	59,44 (2,98)	46,35 (3,11)	44,18 (3,90)	47,86 (3,12)	44,31 (3,54)
	Soma	58,21	61,50	60,09	59,62	59,15
	Produto	57,74	55,86	60,56	59,62	59,15
	Empilhamento	59,62	59,15	46,94	<b>63,38</b>	62,44
LA	Melhor Único	60,40 (3,51)	50,82 (5,00)	45,58 (4,65)	48,72 (4,96)	43,41 (4,61)
	Soma	59,62	57,27	59,15	59,62	58,68
	Produto	60,09	58,68	60,09	58,68	58,21
	Empilhamento	60,56	61,00	47,41	61,03	<b>62,44</b>
SLA	Melhor Único	60,78 (3,89)	50,15 (3,98)	44,12 (3,84)	52,52 (4,05)	44,67 (4,14)
	Soma	60,56	59,62	60,09	60,56	60,56
	Produto	61,03	61,03	59,62	60,56	60,56
	Empilhamento	59,62	61,03	50,23	61,97	<b>65,25</b>

Tabela 21 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da acurácia do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,52 (3,65)	44,99 (3,23)	38,93 (2,65)	42,72 (3,08)	38,73 (2,67)
	Soma	58,68	53,99	50,70	52,58	51,64
	Produto	60,09	53,52	51,64	53,52	51,17
	Empilhamento	<b>60,56</b>	55,39	44,13	51,17	53,52
L	Melhor Único	60,56 (3,82)	48,04 (3,84)	47,26 (5,11)	49,42 (3,96)	47,26 (4,83)
	Soma	61,03	58,21	56,80	58,21	<b>61,97</b>
	Produto	60,56	60,56	57,74	58,68	61,03
	Empilhamento	59,15	56,80	48,82	56,33	60,09
SA	Melhor Único	61,39 (2,99)	46,33 (5,04)	39,72 (3,35)	50,34 (4,12)	42,87 (3,16)
	Soma	64,31	58,21	58,21	57,27	57,74
	Produto	<b>65,25</b>	58,68	60,09	57,27	58,68
	Empilhamento	63,38	58,21	51,17	54,92	53,05
SL	Melhor Único	58,53 (4,01)	42,44 (3,78)	39,54 (3,18)	42,72 (3,64)	40,44 (3,51)
	Soma	<b>60,09</b>	55,86	57,74	52,58	55,86
	Produto	59,62	57,74	56,80	52,58	55,39
	Empilhamento	59,62	54,92	46,94	49,29	55,86
LA	Melhor Único	59,17 (2,50)	50,83 (4,86)	43,97 (4,87)	51,21 (5,31)	45,38 (5,22)
	Soma	58,68	57,27	56,33	56,33	54,92
	Produto	<b>59,62</b>	58,21	56,33	57,74	54,46
	Empilhamento	59,15	58,68	52,11	56,33	57,74
SLA	Melhor Único	58,92 (3,02)	54,24 (6,89)	42,54 (5,32)	51,88 (6,38)	44,51 (4,66)
	Soma	61,97	56,80	59,15	54,92	58,21
	Produto	<b>62,44</b>	54,92	57,27	55,86	57,27
	Empilhamento	61,97	56,33	52,58	53,05	58,21

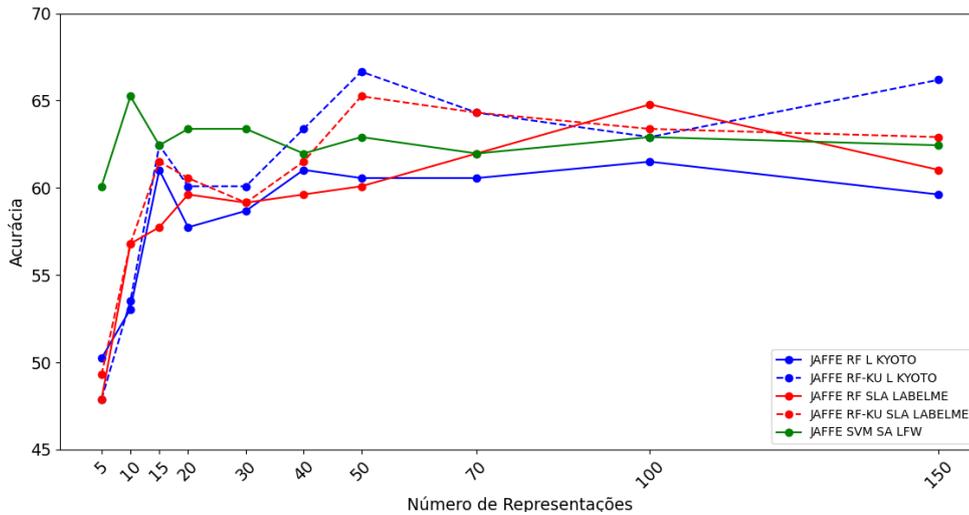


Figura 23 – O impacto da variação do tamanho do conjunto na melhor configuração alcançada para o conjunto de dados JAFFE ao usar conjuntos de dados Kyoto, LabelMe e LFW como auxiliares.

2, 50 a 6, 89 pontos percentuais (4.02 em média), o que era esperado em nosso método. O desvio padrão elevado entre as representações geradas reflete a diversidade nas características aprendidas pelos diferentes *Autoencoders* durante o treinamento. Essa variabilidade é benéfica, pois sugere que as representações abrangem um espectro mais amplo de informações faciais. A combinação dessas representações por meio de empilhamento contribui para um modelo mais robusto, capaz de lidar com a complexidade e variabilidade inerentes aos dados faciais.

A Figura 23 exibe o impacto da variação do tamanho do conjunto na melhor configuração alcançada para o conjunto de dados JAFFE, considerando conjuntos auxiliares Kyoto, LabelMe e LFW. Os conjuntos de representações, compostos por 5, 10, 15, 20, 30, 40, 50, 70, 100 e 150 membros, foram gerados utilizando o método proposto. Cada linha representa a variação do desempenho em relação ao conjunto de dados auxiliar utilizado no treinamento do autoencoder: Kyoto (linha azul), LabelMe (linha vermelha) e LFW (linha verde).

Observa-se que o conjunto de dados auxiliar Kyoto, caracterizado por seu contexto distante do problema alvo e uma menor quantidade de dados em comparação com LFW e LabelMe, alcançou o melhor desempenho. A diversidade intrínseca presente nessa base possibilitou ao Autoencoder Convolutivo (AEC) aprender representações mais distintivas, o que se refletiu em uma melhoria significativa na classificação do conjunto de dados JAFFE.

A análise dos resultados revela que, a partir de 50 representações, não há uma melhora expressiva no desempenho. Esse fenômeno sugere que o modelo atinge um ponto de saturação em termos de diversidade informativa. Adicionar representações além desse ponto pode demandar recursos computacionais substanciais sem proporcionar ganhos

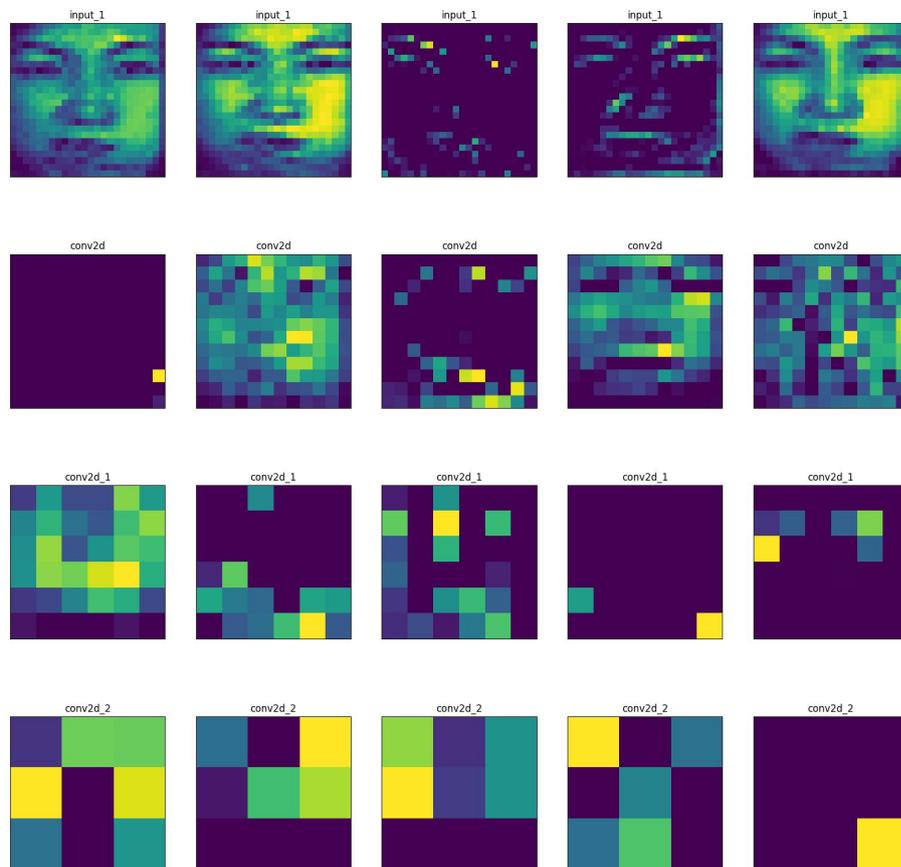


Figura 24 – Visualização dos mapas de características extraídas de uma imagem da base JAFFE nas diferentes camadas do encoder.

correspondentes em termos de performance. Por outro lado, um número menor de representações pode não ser suficiente para abranger toda a variedade de variações e padrões presentes nos dados, resultando em uma sub-representação de características cruciais.

A Figura 24 oferece uma visão das saídas nas várias camadas do encoder da rede, representando a configuração mais otimizada alcançada. Nessa configuração, o autoencoder foi treinado com a base auxiliar Kyoto, adotando a estratégia de vetor latente aleatório para a geração de representações diversificadas.

Pode-se visualizar nas diferentes imagens que o autoencoder detecta diferentes padrões ou características em cada camada. Na linha superior, que corresponde aos mapas de características da primeira camada, as camadas detectam as bordas da imagem. À medida que aprofundamos mais o modelo, as características extraídas das imagens passam por transformações, exibindo uma progressiva redução de detalhes. Esse padrão é esperado,

pois o modelo está focado em abstrair as características das imagens em conceitos mais generalizados, essenciais para a tarefa de classificação. A análise da saída em diferentes camadas do modelo fornece uma visão sobre as características específicas da imagem que são destacadas em cada camada, aprimorando nossa compreensão do processo de extração de características.

### 5.3.2 CK+ como Base Alvo

As Tabelas 22, 23 e 24 apresentam os resultados para o conjunto de dados CK+ quando o AEC é treinado com os conjuntos de dados Kyoto, LabemMe e LFW, respectivamente. O melhor resultado para o conjunto de dados CK+ (92,66%) foi alcançado com um conjunto de 50 representações treinadas no conjunto de dados de Kyoto, variando as sementes e a dimensionalidade do vetor latente (estratégia SL) e utilizando a fusão baseada no empilhamento de SVMs. Também é essencial notar que o melhor resultado baseado em um único classificador SVM neste experimento foi de 91,29%.

O conjunto de dados Kyoto apresentou um resultado superior às demais bases de dados auxiliares utilizadas nos experimentos com o CK+. A diversidade e riqueza de padrões presentes no contexto do Kyoto proporcionaram ao AEC uma capacidade aprimorada de aprendizado, resultando em representações mais discriminativas. Essa distinção contribuiu diretamente para o desempenho superior na tarefa de classificação facial, consolidando a eficácia da Kyoto em sua capacidade de gerar representações não supervisionadas distintas.

Ao utilizar o conjunto de dados LabelMe (Tabela 23), o melhor resultado de 91,74% foi alcançado com um conjunto composto por 30 representações e variando a dimensionalidade do vetor latente (estratégia L), utilizando um conjunto de SVMs. O melhor resultado ao utilizar o conjunto de dados LFW como conjunto de dados auxiliar (90,51%) foi observado utilizando a estratégia SL e um conjunto de SVMs, conforme mostrado na Tabela 24.

Ao contrário do conjunto de dados JAFFE, os métodos de conjunto (BG, RF) e seleção dinâmica (KnoraU) não foram competitivos no conjunto de dados CK+. Uma possível razão é um pequeno desvio padrão na acurácia entre as representações geradas em comparação com os resultados obtidos no conjunto de dados JAFFE. O desvio padrão na acurácia do conjunto de dados CK+ varia de 0,7 a 5,4 pontos percentuais (2,3 em média).

A Figura 25 apresenta o impacto da variação do tamanho do conjunto para o conjunto de dados CK+, considerando membros de 5 a 150 e levando em conta os três conjuntos de dados auxiliares. É possível observar que, para a base não rotulada Kyoto (linha azul), a partir de 50 representações ocorre uma queda na acurácia. Já para LabelMe (linha vermelha) e LFW (linha verde), essa queda é observada a partir de 70 representações. Assim como ocorreu para JAFFE, na base CK+, observamos que o melhor resultado foi obtido com a base Kyoto.

Tabela 22 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da acurácia do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,48 (2,63)	71,75 (3,09)	65,09 (2,01)	71,90 (2,79)	66,62 (1,90)
	Soma	87,15	74,00	68,19	73,39	68,80
	Produto	87,46	74,92	69,11	74,31	68,80
	Empilhamento	<b>89,90</b>	80,42	76,75	76,45	80,12
L	Melhor Único	89,87 (2,58)	72,45 (2,54)	69,78 (2,34)	71,10 (2,26)	69,13 (2,30)
	Soma	89,60	75,84	71,55	75,22	72,17
	Produto	89,29	76,45	71,55	76,14	72,17
	Empilhamento	<b>90,82</b>	77,98	78,89	74,31	81,34
SA	Melhor Único	89,47 (1,80)	71,55 (2,47)	66,66 (1,90)	73,88 (2,60)	67,74 (1,81)
	Soma	89,90	74,92	70,64	75,22	72,17
	Produto	89,29	77,37	70,64	75,84	72,17
	Empilhamento	<b>90,21</b>	80,12	75,84	75,84	78,89
SL	Melhor Único	91,29 (2,86)	72,57 (2,72)	68,31 (2,14)	72,14 (2,82)	67,37 (1,85)
	Soma	89,60	74,92	70,64	74,00	71,55
	Produto	89,60	75,84	70,64	74,00	71,55
	Empilhamento	<b>92,66</b>	77,67	77,06	74,31	78,89
LA	Melhor Único	87,18 (2,24)	72,06 (2,72)	68,88 (2,02)	73,33 (2,85)	69,51 (2,24)
	Soma	86,85	77,37	71,55	77,06	72,47
	Produto	87,46	77,37	72,17	77,06	72,47
	Empilhamento	<b>88,07</b>	79,81	76,14	78,28	79,81
SLA	Melhor Único	87,00 (3,05)	74,23 (3,16)	68,14 (2,20)	73,29 (3,04)	68,58 (2,28)
	Soma	87,76	76,45	71,86	75,53	71,86
	Produto	87,76	77,06	71,86	76,14	72,17
	Empilhamento	<b>88,07</b>	79,51	76,45	76,45	79,81

Tabela 23 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da acurácia do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> def SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	89,57 (4,67)	69,50 (2,23)	66,93 (2,05)	69,97 (2,22)	66,72 (2,23)
	Soma	87,46	73,70	70,03	73,08	70,03
	Produto	87,76	74,00	70,03	73,70	70,03
	Empilhamento	<b>89,90</b>	78,89	77,37	75,22	79,20
L	Melhor Único	90,38 (2,25)	72,16 (2,27)	65,43 (1,47)	71,99 (1,99)	66,76 (1,78)
	Soma	89,90	76,45	69,74	75,53	70,64
	Produto	89,90	75,84	69,72	76,45	70,64
	Empilhamento	<b>91,74</b>	81,34	78,59	74,31	77,98
SA	Melhor Único	88,87 (1,91)	69,19 (2,13)	67,09 (2,11)	70,55 (2,16)	68,37 (2,09)
	Soma	88,68	74,92	70,33	75,53	70,33
	Produto	88,68	76,75	70,64	75,53	70,33
	Empilhamento	<b>89,29</b>	79,81	75,22	77,98	77,37
SL	Melhor Único	89,71 (2,60)	71,08 (2,45)	69,03 (2,16)	71,51 (2,17)	67,31 (1,63)
	Soma	88,68	74,61	70,03	74,31	70,33
	Produto	88,99	74,61	70,33	74,92	70,03
	Empilhamento	<b>90,51</b>	79,81	75,22	75,53	77,37
LA	Melhor Único	89,39 (3,84)	72,37 (2,68)	67,92 (2,53)	71,86 (2,82)	69,49 (3,01)
	Soma	90,21	77,06	70,94	75,84	70,94
	Produto	89,90	77,67	71,25	76,75	70,94
	Empilhamento	<b>90,51</b>	80,12	75,53	77,98	79,51
SLA	Melhor Único	90,45 (2,38)	71,17 (3,21)	66,38 (2,36)	73,13 (3,43)	67,93 (1,95)
	Soma	89,29	75,22	70,33	74,92	70,64
	Produto	89,29	74,92	71,25	75,22	70,64
	Empilhamento	<b>90,51</b>	79,81	76,45	76,14	76,75

Tabela 24 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da acurácia do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> of SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	83,24 (1,69)	66,75 (1,68)	63,22 (1,28)	66,96 (2,04)	64,77 (1,17)
	Soma	85,93	73,08	68,80	71,86	69,41
	Produto	85,93	73,39	69,11	73,08	69,72
	Empilhamento	<b>87,15</b>	74,92	73,70	72,47	72,47
A	Melhor Único	84,91 (0,74)	66,82 (1,04)	65,38 (1,42)	67,20 (0,96)	66,37 (1,80)
	Soma	87,46	74,00	67,88	73,70	70,03
	Produto	<b>88,07</b>	74,92	68,19	74,00	70,33
	Empilhamento	87,76	76,45	73,70	73,39	74,31
L	Melhor Único	84,27 (3,47)	68,62 (3,21)	64,26 (2,50)	69,15 (3,20)	64,50 (2,25)
	Soma	<b>87,76</b>	75,35	68,80	75,53	68,50
	Produto	87,46	77,06	68,19	74,92	68,80
	Empilhamento	87,46	78,28	74,92	76,14	73,39
SA	Melhor Único	88,16 (3,35)	71,96 (3,05)	66,77 (1,56)	72,45 (3,01)	67,07 (2,36)
	Soma	87,76	72,17	70,64	71,55	71,55
	Produto	87,76	73,08	70,64	71,55	71,55
	Empilhamento	<b>88,99</b>	74,61	75,84	71,55	74,31
SL	Melhor Único	85,69 (0,99)	68,75 (2,29)	66,96 (2,46)	68,77 (1,88)	68,26 (2,22)
	Soma	90,21	75,53	70,33	75,84	70,64
	Produto	89,60	75,22	70,33	75,53	70,94
	Empilhamento	<b>90,51</b>	78,89	77,67	75,22	77,37
LA	Melhor Único	83,92 (1,42)	71,56 (2,68)	67,17 (1,30)	69,83 (2,02)	68,24 (1,72)
	Soma	87,15	72,47	70,94	73,08	70,33
	Produto	86,85	73,39	70,33	72,78	70,33
	Empilhamento	<b>87,76</b>	76,75	77,06	71,86	75,22
SLA	Melhor Único	86,70 (5,41)	71,14 (4,09)	66,97 (2,69)	70,67 (3,90)	65,14 (1,86)
	Soma	86,54	74,92	69,41	73,08	69,41
	Produto	85,62	75,22	69,72	74,61	69,41
	Empilhamento	<b>86,85</b>	74,61	73,70	73,70	74,00

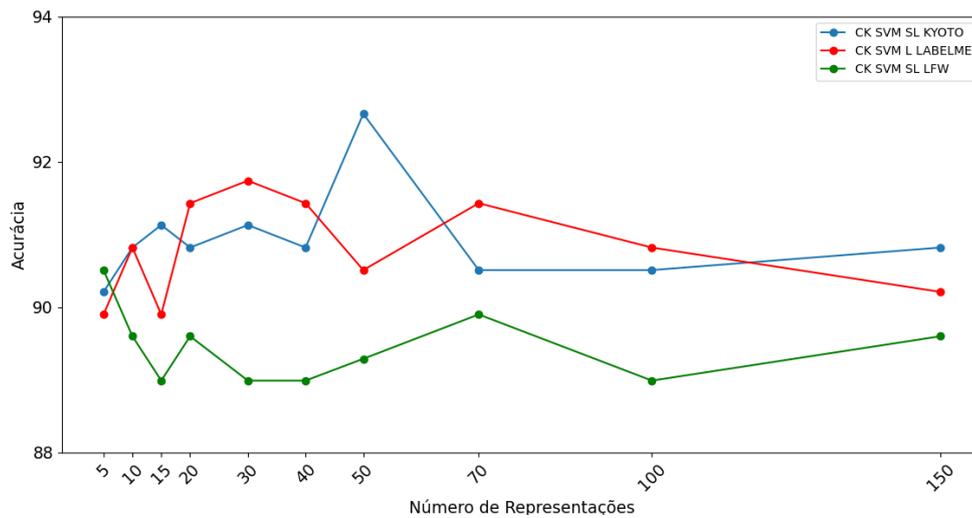


Figura 25 – O impacto da variação do tamanho do conjunto na melhor configuração alcançada para o conjunto de dados CK+ ao usar conjuntos de dados Kyoto, LabelMe e LFW como auxiliares.

Figura 26 exibe a visualização das saídas nas diversas camadas do encoder da rede, representando a configuração mais otimizada alcançada. Nesse contexto, o autoencoder foi treinado com a base auxiliar Kyoto, empregando a estratégia de vetor latente e inicialização aleatória (semente) para a geração de representações diversificadas.

Nas diferentes imagens, é possível visualizar que o autoencoder identifica padrões ou características singulares em cada camada, isso nos ajuda a entender o que o modelo procura nas imagens. Por meio da visualização de recursos, podemos compreender os atributos específicos que a rede está processando em diferentes camadas. Nas camadas mais próximas à entrada (linhas superiores), o modelo busca encontrar características como arestas, enquanto nas camadas mais profundas (linhas inferiores) o modelo destacará conceitos mais abstratos. Estas informações ajudam a compreender o funcionamento interno da rede e podem auxiliar no ajuste para melhorar o desempenho do modelo, refinando o processo de treinamento.

## 5.4 Ablation

Após gerar a primeira representação, consideramos um termo de penalidade na função de custo para garantir que as representações subsequentes sejam diversas, conforme mostrado na Equação 4.2. O parâmetro  $\beta$  é uma constante definida pelo usuário que desempenha um papel essencial neste cenário, pois é utilizado para controlar a penalidade aplicada pela geração de diversidade entre as representações. Em outras palavras, determina a importância da penalidade aplicada às representações para incentivar a exploração de diferentes regiões do espaço de características. Para alcançar resultados eficazes e aplicáveis

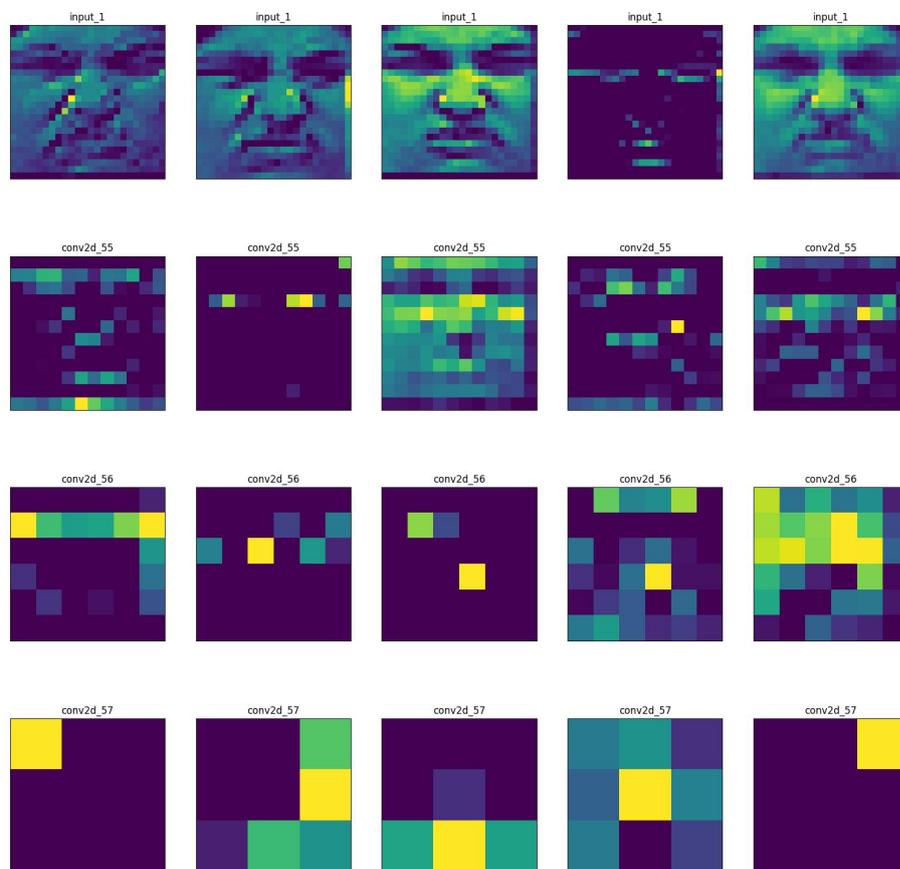


Figura 26 – Visualização dos mapas de características extraídas de uma imagem da base CK+ nas diferentes camadas do encoder.

a problemas específicos, é crucial otimizar o parâmetro  $\beta$ . Dessa forma, para abordar nossa Questão 3 de pesquisa, conduzimos experimentos exploratórios com diversos valores de  $\beta$  na Equação 4.2.

Os melhores resultados para os conjuntos de dados JAFFE e CK+ são mostrados nas Tabelas 19 e 22, respectivamente. Para o estudo do *ablation*, utilizamos as melhores configurações de desempenho. Assim, geramos 50 representações com o conjunto de dados não rotulado Kyoto e aplicamos valores diferentes para  $\beta$ . Table 25 mostra o desempenho do modelo obtido para cada valor de  $\beta$ .

Os melhores resultados nos conjuntos de dados JAFFE e CK+ foram encontrados usando  $\beta = 0,001$ . Quando aumentamos ou diminuimos o valor de  $\beta$ , reduzimos o desempenho do modelo. O valor  $\beta$  é ajustado experimentalmente para encontrar o equilíbrio certo entre a qualidade das representações geradas e a diversidade desejada. Se o  $\beta$  for

Tabela 25 – Experimento *ablation* do parâmetro  $\beta$  considerando a melhor configuração do método proposto para os conjuntos de dados JAFFE e CK+.

	$\beta$	Acurácia (%)
JAFFE	0,0001	60,56
	0,001	<b>66,66</b>
	0,01	62,91
	0,1	60,56
	–	61,03
CK	0,0001	91,13
	0,001	<b>92,66</b>
	0,01	89,60
	0,1	90,82
	–	90,82

muito baixo, a diversidade pode não ser suficientemente encorajada e as representações podem tornar-se demasiado semelhantes. Por outro lado, se  $\beta$  for muito alto, a penalidade pode sobrecarregar completamente o objetivo, resultando em representações muito diversas, mas potencialmente de baixa qualidade.

A última entrada da Tabela 25 exibe o desempenho obtido no experimento conduzido sem a implementação da função de custo personalizada apresentada na Equação 4.2. Este resultado evidencia um aumento no desempenho, em comparação com o caso em que a penalização é incorporada à função de custo, ou seja, quando se impõe a criação de representações distintas.

Além de conduzir experimentos de *ablation*, nos quais exploramos variações e a não utilização do parâmetro  $\beta$ , realizamos também uma análise do impacto dos pré-processamentos empregados neste estudo. As imagens das bases JAFFE e CK+ passam por procedimentos prévios, conforme exemplificado na Figura 21. Os resultados obtidos, também para as melhores configurações de desempenho, indicam que o pré-processamento adotado desempenha um papel crucial no funcionamento global do sistema. Os índices alcançados foram de 46,00% para JAFFE e 65,24% para CK+, valores significativamente inferiores em comparação com as melhores pontuações apresentadas nas tabelas 19 e 22, respectivamente.

## 5.5 Comparação com soluções baseadas em CNN e trabalhos relacionados

O desempenho de abordagens supervisionadas contra AAD é assunto de discussão uma vez que a primeira é amplamente utilizada na literatura e geralmente alcança os melhores resultados, portanto, para responder corretamente a nossa Questão 4 de pesquisa, avaliamos a estratégia tradicional de aprendizagem supervisionada utilizando o protocolo

LOSO apresentada na Seção 5.1, na qual as amostras dos sujeitos de teste não estão presentes na etapa de treinamento.

No estado da arte, há uma infinidade de arquiteturas CNNs prontas para uso. Para evitar uma comparação tendenciosa, a justificativa é selecionar redes bem conhecidas que proporcionaram um avanço em suas arquiteturas e alcançaram desempenhos competitivos no Desafio ImageNet (ALZUBAIDI et al., 2021).

A primeira abordagem bem-sucedida foi feita por Krizhevsky, Sutskever e Hinton (2017), implementando uma arquitetura convolucional denominada AlexNet composta por 5 camadas. Simonyan e Zisserman (2014b)(VGG) propuseram uma arquitetura mais profunda. Ao contrário da lógica de empilhamento de camadas que aumentam a profundidade da rede, Szegedy et al. (2015) propuseram os módulos iniciais compostos por um padrão de camadas convolucionais, *pooling* e concatenação de recursos. A arquitetura final é uma pilha de módulos *Inception* que fornecem uma rede mais abrangente e aprimoram a representação de recursos. Até agora, a maioria das arquiteturas propostas recorreu ao fato de que altos desempenhos deveriam ser alcançados à medida que a rede cresce. No entanto, o desaparecimento do gradiente tornou-se um problema. Para minimizar o *overfitting*, conexões de salto entre camadas adjacentes foram propostas no ResNet (HE et al., 2016a) e melhoradas no DenseNet (HUANG et al., 2017), que estende as conexões de salto entre todas as camadas da rede. Por último, EfficientNet (TAN; LE, 2019) propôs uma técnica para fornecer escalonamento do modelo para melhorar a eficiência da rede.

As arquiteturas mencionadas acima e suas variantes em termos de número de camadas foram treinadas para conjuntos de dados JAFFE e CK+. Para uma comparação justa, incluímos também a arquitetura do codificador do AEC (Seção 4.1.3). Exceto para AlexNet e AECs, usamos os pesos pré-treinados no conjunto de dados ImageNet, considerando dois esquemas de ajuste fino diferentes. Primeiro, (1) todos os pesos das camadas convolucionais são congelados e apenas as camadas totalmente conectadas são ajustadas ao problema FER. O segundo esquema também ajusta a última camada convolucional (2).

A justificativa para o uso de pesos pré-treinados baseia-se no fato de que os conjuntos de dados REF não possuem dados suficientes para ajustar corretamente os modelos profundos. Além disso, esta estratégia é uma prática comum no estado da arte (XU et al., 2015; ZHANG et al., 2019; PENG et al., 2018; SHAO; QIAN, 2019) e deve fornecer insights significativos ao comparar a estratégia TL com a estratégia AAD.

A Tabela 26 exibe os resultados de desempenho obtidos em todos os experimentos realizados. As colunas identificadas como (1) e (2) indicam os resultados empregando os esquemas de ajuste fino correspondentes para os conjuntos de dados FER. Os modelos marcados com um asterisco são aqueles que foram treinados do zero, uma vez que possuíam menos camadas disponíveis para ajustes. Esta análise detalhada proporciona uma visão abrangente do impacto dos diferentes esquemas de treinamento e da arquitetura específica de cada modelo na eficácia do reconhecimento de expressões faciais.

Tabela 26 – Arquiteturas CNNs treinadas com o protocolo LOSO e duas estratégias de ajuste fino (1, 2).

Arquitetura	CK+(1)	CK+(2)	JAFFE(1)	JAFFE(2)
VGG_16	78,9	46,8	40,6	14,1
VGG_19	81,4	61,1	44,5	14,5
InceptionV3	75,7	84,2	42,2	36,9
ResNet50	84,3	81,6	<b>56,7</b>	47,9
ResNet101	83,5	81,3	43,4	51,2
InceptionResnetV2	78,5	84,6	36,6	44,2
EfficientNetB0	80,6	<b>86,3</b>	38,3	49,0
EfficientNetB7	75,2	83,5	33,6	<b>52,6</b>
DenseNet121	81,1	81,1	54,9	45,4
DenseNet201	<b>85,8</b>	82,0	54,5	50,7
Encoder (AEC)*		69,3		39,3
Alexnet*		70,6		42,9
<b>Ours</b>		<b>92,66</b>		<b>66,66</b>

(\*)O modelo foi treinado do zero (sem pesos pré-treinados)

Os resultados obtidos a partir dos experimentos fornecem algumas conclusões. O modelo AER proposto demonstrou um desempenho superior em comparação com os modelos Convolutional Neural Networks (CNN) pré-treinados. Ao avaliar o conjunto de dados JAFFE, a melhor solução CNN, utilizando a arquitetura ResNet50, atingiu uma taxa de acerto de 56,7%, enquanto o método proposto, alcançou 66,6%.

Este padrão de desempenho superior foi alcançado também no conjunto de dados CK+. Em termos de CNN, a solução mais eficaz alcançou uma taxa de acerto de 86,3%, enquanto o método proposto obteve uma acurácia de 92,6%. Estes resultados evidenciam que o método proposto, empregando a aprendizagem autodidata, supera o desempenho da aprendizagem por transferência, revelando a capacidade na extração de características ao utilizar bases de contexto diversificado no treinamento dos *autoencoders*, indicando uma abordagem promissora em comparação com as CNNs pré-treinadas.

A compensação entre tempo de classificação e acurácia é discutida ao comparar AAD e abordagens supervisionadas tradicionais. Para esclarecer isso, avaliamos o tempo de classificação para cada conjunto de testes utilizando o protocolo LOSO. Os resultados estão representados nas Figuras 27 e 28, para conjuntos de dados JAFFE e CK+, respectivamente. Nessa análise, comparamos o melhor conjunto gerado pelo algoritmo AER com as abordagens tradicionais de aprendizagem supervisionada, discutidas anteriormente na Tabela 26. Eles são comparados considerando o tempo de execução durante o teste (tempo de classificação), acurácia e tamanho do modelo em megabytes.

A abordagem proposta geralmente apresenta uma melhor compensação em relação ao tempo de classificação versus acurácia. Um exemplo prático pode ser visto na Figura 27, onde a abordagem RF-KnoraU supera as conhecidas redes VGG16, Inception e ResNet. Na Figura 28, o resultado do algoritmo AER obtido usando um conjunto de SVMs no

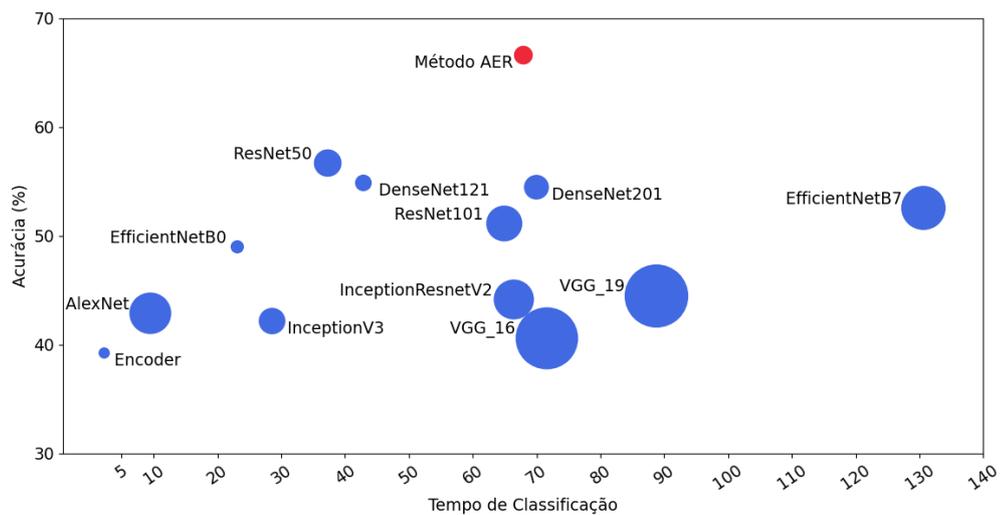


Figura 27 – Comparação do método proposto baseado em AAD (em vermelho) com arquiteturas CNNs (em azul) em termos de acurácia (%), tempo de classificação (em segundos) e espaço em disco (em megabytes) como o tamanho do círculo. Os resultados foram calculados usando o conjunto de dados JAFPE e o protocolo LOSO. O método baseado em AAD considera o classificador final RF com seleção dinâmica KnoraU.

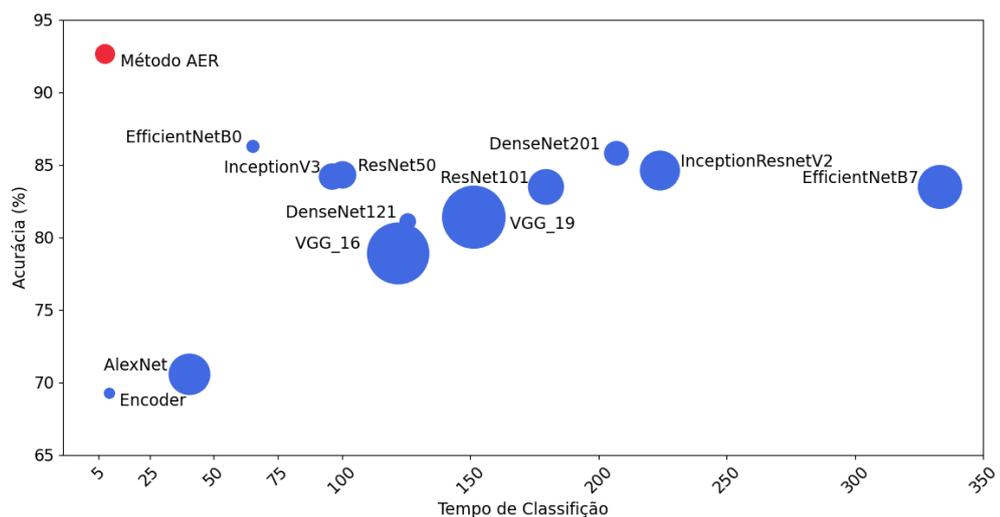


Figura 28 – Comparação do método proposto baseado em AAD (em vermelho) com arquiteturas CNNs (em azul) em termos de acurácia (%), tempo de classificação (em segundos) e espaço em disco (em megabytes) como o tamanho do círculo. Os resultados foram calculados usando o conjunto de dados CK+ e o protocolo LOSO. O método baseado em AAD considera como classificador final um conjunto de SVMs.

Tabela 27 – Comparação de referência no conjunto de dados JAFFE usando o protocolo LOSO.

Artigo	Método	Acurácia (%)	Classes	Tipo de Características
Kola e Samayamantula (2021)	LGC-HD	60,70	6	Manual
	LGC-HD	58,20	7	Manual
Kartheek, Prasad e Bhukya (2023)	RMP_Prime	61,64	6	Manual
Mandal et al. (2019)	DRADAP	57,22	6	Manual
	ARADAP	56,20	7	Manual
Nosso método		<b>66,66</b>	7	Profundas

Tabela 28 – Comparação de referência no conjunto de dados CK+ usando o protocolo LOSO.

Artigo	Método	Acurácia (%)	Classes	Tipo de Características
Du e Hu (2019)	WPLBP	91,72	6	Manual
	WPLBP	86,47	7	Manual
Wu e Lin (2018)	GM	86,83	7	Profundas
	GM+AFM	87,78	7	Profundas
	GM+ W-AFM	88,25	7	Profundas
	GM + W-CRAFM	89,84	7	Profundas
Lee, Baddar e Ro (2016)	CER-ICV	92,34	7	Manual
Kola e Samayamantula (2021)	LGC-HD	72,80	6	Manual
	LGC-HD	70,60	7	Manual
Nosso método		<b>92,66</b>	7	Profundas

conjunto de dados CK+ supera as arquiteturas CNNs em termos de acurácia e a maioria delas em termos de tempo de classificação.

Por fim, apresentamos um estudo comparativo com os resultados do nosso modelo proposto, que aborda a geração de um conjunto de representações não supervisionadas e sua relação com os mais recentes avanços no reconhecimento de emoções faciais, utilizando o mesmo protocolo experimental. Esta comparação visa fornecer uma compreensão ampliada das contribuições e melhorias trazidas pela nossa abordagem. Nosso esforço se concentrou em reunir o maior número possível de estudos que aderissem ao mesmo protocolo de avaliação adotado com base no protocolo LOSO. As Tabelas 27 e 28 apresentam as maiores precisões observadas nos conjuntos de dados JAFFE e CK+, respectivamente, que foram documentadas em publicações e conferências científicas amplamente referenciadas. Os dados revelaram que o método proposto estabelece uma nova referência para o reconhecimento de expressões faciais.

Embora alguns resultados das Tabelas 27 e 28 não sejam diretamente comparáveis devido a configurações experimentais distintas, classes de expressão diversas e métodos de pré-processamento variados, fica evidente que o método proposto gera uma taxa de reconhecimento viável e promissora, alcançando 66,66% para JAFFE e 92,66% para CK.

Através destes resultados, torna-se evidente que a geração de um conjunto de

representações não supervisionadas caracterizadas pela sua diversidade desempenha um papel fundamental na melhoria do desempenho de um sistema FER.

## 5.6 Discussões

Em relação ao conjunto de dados auxiliares, os experimentos mostraram que os melhores resultados foram encontrados quando o conjunto de representações foi aprendido utilizando o pequeno conjunto de dados de Kyoto. As evidências mostram que a variabilidade das imagens no conjunto de dados auxiliar é mais importante do que a quantidade. Além disso, o conjunto de dados de Kyoto está longe do domínio alvo, mas ainda fornece melhores resultados quando comparado com os observados utilizando o conjunto de dados LFW.

Os resultados obtidos indicam uma relação entre o número de representações e a proximidade contextual das bases auxiliares em relação à expressão facial. Ao avaliar o desempenho nas bases de dados JAFFE e CK+, observamos que a quantidade de representações para o melhor resultado variou em função do contexto da base auxiliar.

Para a base JAFFE, notou-se que os melhores resultados foram alcançados com 50 representações nas bases auxiliares KYOTO e LABELME, ambas mais distantes do contexto específico de expressão facial. De forma correspondente, a base auxiliar LFW, inserida no contexto facial, obteve desempenho otimizado com apenas 10 representações.

Paralelamente, ao analisarmos os resultados para a base CK+, constatamos uma tendência semelhante. A base auxiliar KYOTO, mais distante do contexto facial, alcançou melhor desempenho com 50 representações. Já a base auxiliar LABELME, também distante, atingiu o melhor resultado com 30 representações. Enquanto isso, a base auxiliar LFW, mais próxima do contexto de expressão facial, demonstrou eficácia com apenas 5 representações.

Esses resultados apontam para uma relação inversamente proporcional entre a proximidade ao contexto facial e a quantidade de representações necessárias. Quanto mais próxima à base auxiliar estiver do contexto facial, menor é o número de representações requeridas. No entanto, é essencial ressaltar que a escolha do número adequado de representações também está ligada à complexidade do conjunto de dados.

Com um número menor de representações, há possibilidade de o modelo não adquirir uma diversidade suficiente para abranger a complexidade do conjunto de dados, resultando em uma sub-representação de características. Por outro lado, ao considerar um número maior, como 150 representações, pode-se introduzir redundâncias, tornando o modelo excessivamente complexo e potencialmente mais suscetível a sobreajuste.

Com base nos resultados obtidos em nossos experimentos e análises, podemos afirmar de maneira positiva em relação à nossa segunda questão de pesquisa. As representações não supervisionadas, caracterizadas pela diversidade alcançada por meio do algoritmo

AER, não apenas impactaram significativamente o desempenho do modelo REF proposto, mas também se mostraram complementares. Este aspecto foi evidenciado pelo ganho de desempenho observado no pool de representações geradas, juntamente com a amplitude do desvio padrão.

Ao incorporar a diversidade de maneira não supervisionada por meio de diferentes estratégias de inicialização e uma função de custo personalizada, o algoritmo AER demonstrou sua eficácia na produção de um conjunto de representações abrangentes. Essas representações capturam características latentes e complexas em expressões faciais, permitindo ao modelo extrair características significativas e distintivas.

Obtivemos alta acurácia no reconhecimento de expressões faciais treinando o modelo com as representações não supervisionadas geradas pelo AER. O método proposto é até 5,1% e 1,3% melhor que o classificador único para conjuntos de dados JAFFE e CK+, respectivamente.

A diversidade introduzida pelo AER possibilitou que o modelo se tornasse mais adaptável a diferentes variações de expressões, incluindo nuances e diferenças sutis entre categorias de expressões. Além disso, a generalização do modelo foi bastante melhorada. As representações diversificadas permitiram ao modelo aprender características gerais que não se restringiam a um conjunto específico de dados de treinamento, tornando-o mais eficiente na identificação de expressões faciais em novos conjuntos de dados. Isso reduz a tendência de superajuste, evitando que o modelo se especialize demais em dados de treinamento específicos.

Além disso, a inclusão de um conjunto de representações não supervisionadas geradas pelo algoritmo AER facilitou o processo de transferência de conhecimento. O pré-treinamento das representações não supervisionadas permitiu que o modelo REF fosse inicializado com conhecimentos mais abrangentes e, posteriormente, ajustado para a tarefa específica do FER. Isso melhorou o desempenho geral do modelo.

A resposta à quarta questão de pesquisa só foi possível após os últimos experimentos nos quais a abordagem proposta baseada em AAD foi comparada favoravelmente com as soluções REF baseadas em CNN.

A análise comparativa entre o método proposto e os modelos CNNs ajustados para os conjuntos de dados JAFFE e CK+ revelou melhorias consideráveis em termos de acurácia. Especificamente, o método proposto superou os modelos CNNs ajustados em até 9,9% no conjunto de dados JAFFE e em até 6,3% no conjunto de dados CK+. Estes ganhos substanciais em acurácia são de grande relevância e demonstram a eficácia da abordagem baseada em AAD.

Uma comparação favorável com soluções CNNs baseadas em supervisão sugere que a abordagem proposta baseada em AAD pode fornecer resultados mais promissores e confiáveis para reconhecimento de expressão facial. Ao eliminar a necessidade de muitos rótulos nos estágios iniciais do treinamento, a abordagem AAD provou ser eficiente e

econômica, ao mesmo tempo em que alcança resultados competitivos em termos de acurácia e generalização.

## 6 Conclusão

Propusemos uma abordagem baseada em aprendizagem autodidata para o reconhecimento de expressões faciais. Os avanços da metodologia proposta em relação à literatura existente podem ser resumidos em quatro pontos-chave.

Primeiramente, conduzimos experimentos para realizar uma avaliação detalhada de diferentes estratégias para a geração de diversidade durante a geração de um *pool* não supervisionado de representações, essas estratégias incluem variação na inicialização, arquitetura e dados de treinamento. Respondendo à primeira questão de pesquisa, observamos que as melhores abordagens estão diretamente relacionadas à arquitetura CAE: variação do tamanho do vetor latente para JAFFE e da profundidade da rede para CK+, foram as estratégias mais promissoras.

Em segundo lugar, apresentamos o algoritmo AER, que tem a capacidade de gerar um conjunto diversificado de representações, utilizando dados não supervisionados provenientes de contextos distintos da tarefa alvo e aplica diferentes estratégias de inicialização para introduzir diversidade no processo. Avaliamos o SVM, *Bagging*, *Random Forest* e um método de seleção de conjunto dinâmico (KnoraU) como classificadores que se beneficiam das representações AAD não supervisionadas. Os resultados obtidos indicaram que a utilização do *pool* de representações geradas pelo AER proposto resultou em uma melhoria significativa no desempenho para as bases JAFFE e CK+, permitindo uma resposta positiva à nossa segunda questão de pesquisa.

Em terceiro lugar, propomos uma função de custo que leva em consideração a dissimilaridade entre as representações, aplicando penalidades graduais àquelas que são mais semelhantes. À medida que a similaridade entre as representações geradas aumenta, a função de custo impõe penalidades mais expressivas, promovendo, assim, uma maior diversidade entre as representações geradas pelo AEC. Os resultados provenientes da análise de *ablation* demonstraram que o planejamento adequado da função de custo pode contribuir de maneira significativa para o aprendizado e diversificação das representações, respondendo positivamente à nossa terceira questão de pesquisa.

Em quarto lugar, conduzimos uma avaliação de desempenho robusta, comparando diretamente nossa solução REF não supervisionada com métodos supervisionados em condições idênticas. Isso implica a utilização do mesmo protocolo experimental, enfrentando assim os desafios inerentes a uma tarefa genuína e desafiadora como o Reconhecimento de Expressões Faciais (REF). Os resultados apresentados em resposta à nossa quarta questão de pesquisa demonstram que a solução proposta pode ser favoravelmente comparada com as abordagens REF baseadas em CNNs.

Os resultados experimentais em dois conjuntos de dados REF bem conhecido demonstraram que o *pool* de representações gera resultados competitivos com o estado

da arte e é competitivo quando comparado às CNNs. Além disso, reduz a dependência de dados rotulados no momento de criar a solução para o problema. O método proposto, embora promissor, apresenta uma desvantagem em termos de custo computacional durante o treinamento das múltiplas representações. À medida que a quantidade necessária de representações para a geração do pool aumenta, o custo computacional acompanha esse crescimento proporcional.

Em trabalhos futuros, pretendemos avaliar um esquema alternativo para a função custo empregada no aprendizado do conjunto não supervisionado de representações latentes, visando amplificar a diversidade. Além disso, planejamos estender a avaliação do algoritmo AER proposto para cobrir diferentes problemas-alvo, explorando sua aplicabilidade e eficácia em outros contextos. Estas iniciativas visam melhorar ainda mais o alcance e a robustez das nossas contribuições.

## A Apêndice

Nesta seção, apresentamos os resultados obtidos para cada base alvo (JAFFE e CK+) a partir dos experimentos realizados com os três conjuntos de dados auxiliares (Kyoto, LabelMe e LFW) para uma variedade de tamanhos de representações geradas, incluindo 5, 10, 15, 20, 30, 40, 50, 70, 100 e 150. Além disso, para cada tamanho de representação, as estratégias S, A, L, SA, SL, LA e SLA foram testadas em cada experimento.

Os resultados para a base JAFFE, considerando cada tamanho de representações geradas para as três bases auxiliares e utilizando as estratégias de diversidade, são apresentados nas Tabelas 29 até 60. Assim como, para a base CK+, os resultados são apresentados desde a Tabela 61 até a Tabela 90.

Tabela 29 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	52,96 (2,41)	44,66 (4,37)	41,44 (3,44)	44,12 (4,39)	41,63 (5,23)
	Soma	54,92	50,70	48,35	53,05	49,76
	Produto	<b>54,46</b>	53,05	48,82	53,99	50,23
	Empilhamento	<b>54,46</b>	52,58	47,41	50,23	46,00
A	Melhor Único	56,12 (1,66)	54,08 (8,11)	43,62 (4,58)	53,19 (6,58)	42,68 (3,06)
	Soma	58,68	54,92	51,64	54,46	49,29
	Produto	58,68	53,05	53,05	54,92	49,29
	Empilhamento	<b>59,15</b>	54,46	45,07	51,17	49,76
L	Melhor Único	58,02 (2,14)	47,51 (4,76)	44,79 (4,52)	47,44 (4,35)	39,63 (2,69)
	Soma	57,74	54,46	49,29	55,39	45,07
	Produto	57,27	56,33	50,23	53,99	46,00
	Empilhamento	<b>58,68</b>	55,39	41,31	52,58	47,88
SA	Melhor Único	56,28 (2,48)	46,25 (3,06)	40,32 (3,28)	45,16 (2,47)	42,72 (3,77)
	Soma	57,27	47,41	49,76	51,17	52,58
	Produto	<b>58,21</b>	49,29	50,23	49,76	52,58
	Empilhamento	56,33	49,29	44,13	50,70	44,13
SL	Melhor Único	56,53 (2,04)	50,34 (4,68)	39,90 (2,45)	47,91 (3,95)	40,45 (3,00)
	Soma	56,80	52,11	53,99	52,11	54,46
	Produto	<b>57,27</b>	54,46	53,05	52,11	54,46
	Empilhamento	56,33	53,99	49,29	53,52	54,46
LA	Melhor Único	54,99 (4,99)	43,83 (1,72)	38,33 (2,61)	44,41 (1,85)	41,65 (4,09)
	Soma	55,39	<b>55,86</b>	49,76	53,99	50,70
	Produto	54,46	53,52	50,23	52,58	50,70
	Empilhamento	54,46	54,92	46,47	50,70	47,41
SLA	Melhor Único	57,14 (1,96)	56,65 (6,93)	46,77 (4,94)	56,13 (7,52)	45,66 (5,21)
	Soma	59,15	60,09	58,21	<b>61,03</b>	57,74
	Produto	59,15	58,21	58,68	60,56	58,21
	Empilhamento	60,09	60,56	54,46	59,15	58,68

Tabela 30 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	56,08 (2,68)	45,97 (3,40)	40,69 (2,30)	46,40 (3,47)	44,12 (3,86)
	Soma	58,68	55,39	56,33	53,52	53,52
	Produto	<b>61,50</b>	58,68	55,86	55,86	52,58
	Empilhamento	58,68	57,27	46,94	53,52	54,46
L	Melhor Único	57,98 (3,15)	45,41 (4,12)	38,47 (3,39)	44,73 (3,52)	40,55 (3,91)
	Soma	62,44	59,62	53,05	59,15	53,52
	Produto	<b>63,38</b>	58,68	53,05	58,21	52,58
	Empilhamento	61,03	55,39	42,25	56,80	51,17
SA	Melhor Único	58,88 (3,60)	45,70 (2,29)	42,59 (4,31)	45,22 (2,82)	42,16 (3,71)
	Soma	59,15	52,11	55,39	52,58	57,74
	Produto	57,27	52,58	57,27	53,52	58,68
	Empilhamento	59,62	51,17	47,88	46,47	<b>60,09</b>
SL	Melhor Único	58,93 (4,62)	45,72 (3,25)	45,07 (4,55)	44,33 (2,80)	42,10 (4,59)
	Soma	59,62	55,86	59,62	54,46	<b>61,03</b>
	Produto	58,68	58,68	60,56	57,76	60,56
	Empilhamento	59,15	56,33	49,29	53,99	54,46
LA	Melhor Único	62,01 (2,99)	43,61 (2,71)	39,11 (2,95)	46,49 (3,71)	40,95 (3,66)
	Soma	61,50	58,68	52,58	58,21	53,52
	Produto	61,03	55,39	53,99	58,68	53,99
	Empilhamento	<b>62,44</b>	55,39	45,53	53,99	54,46
SLA	Melhor Único	60,33 (2,57)	49,34 (4,41)	47,31 (5,41)	48,40 (4,32)	50,11 (6,14)
	Soma	60,09	55,39	55,86	55,86	51,17
	Produto	61,97	55,86	55,39	56,33	51,17
	Empilhamento	<b>62,44</b>	55,86	48,82	53,52	52,58

Tabela 31 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	56,17 (2,76)	48,57 (4,03)	42,91 (3,11)	48,59 (4,33)	41,60 (3,12)
	Soma	<b>58,21</b>	57,74	53,52	53,99	53,52
	Produto	57,74	55,86	54,46	55,39	53,52
	Empilhamento	<b>58,21</b>	<b>58,21</b>	48,82	51,17	52,58
L	Melhor Único	58,51 (2,38)	45,84 (2,02)	44,70 (2,83)	46,84 (2,84)	44,36 (3,90)
	Soma	58,21	56,80	61,03	56,33	61,97
	Produto	59,15	59,15	61,03	56,80	<b>62,44</b>
	Empilhamento	56,80	59,15	50,23	56,80	60,56
SA	Melhor Único	60,35 (3,53)	49,93 (4,03)	42,90 (3,67)	49,04 (3,69)	43,05 (3,26)
	Soma	61,97	58,68	60,09	55,86	61,50
	Produto	62,44	60,09	60,56	57,74	61,50
	Empilhamento	<b>62,91</b>	58,21	50,70	54,46	60,09
SL	Melhor Único	56,48 (3,12)	44,79 (3,91)	40,24 (2,87)	44,92 (3,09)	41,62 (4,52)
	Soma	56,80	55,39	55,39	57,27	55,39
	Produto	<b>57,74</b>	56,33	55,39	56,80	55,86
	Empilhamento	54,46	54,92	45,07	53,52	56,33
LA	Melhor Único	58,48 (2,76)	49,12 (4,15)	41,56 (2,97)	49,48 (3,86)	41,23 (2,53)
	Soma	60,56	59,15	57,74	60,09	58,68
	Produto	<b>60,56</b>	58,68	57,27	60,09	58,68
	Empilhamento	<b>60,56</b>	58,21	50,23	56,80	59,15
SLA	Melhor Único	59,05 (2,88)	47,49 (3,90)	45,56 (3,64)	47,02 (3,75)	47,04 (3,75)
	Soma	58,68	<b>61,03</b>	60,09	60,09	57,27
	Produto	59,62	58,68	60,09	59,62	56,80
	Empilhamento	59,15	60,56	49,76	57,27	59,15

Tabela 32 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	56,15 (2,03)	46,23 (2,87)	40,90 (2,98)	46,23 (3,18)	40,69 (3,10)
	Soma	60,09	61,50	61,97	61,97	<b>63,38</b>
	Produto	61,03	62,91	60,56	61,97	62,91
	Empilhamento	61,03	61,97	51,64	54,46	61,03
L	Melhor Único	60,44 (5,18)	9,02 (4,56)	48,11 (4,75)	48,13 (4,20)	47,18 (4,30)
	Soma	59,62	<b>61,50</b>	57,74	<b>61,50</b>	58,68
	Produto	59,15	60,56	56,80	<b>61,50</b>	58,68
	Empilhamento	61,03	63,84	49,29	60,09	60,09
SA	Melhor Único	58,48 (2,68)	45,19 (2,39)	39,60 (3,12)	44,80 (2,30)	41,20 (3,23)
	Soma	61,97	56,33	57,27	56,33	58,68
	Produto	61,50	56,80	57,74	58,21	58,68
	Empilhamento	<b>61,97</b>	54,92	46,00	56,80	59,62
SL	Melhor Único	59,48 (2,75)	50,76 (3,95)	43,51 (2,75)	52,76 (4,16)	44,00 (2,63)
	Soma	60,09	57,74	64,31	56,33	<b>64,78</b>
	Produto	60,56	60,56	<b>64,78</b>	57,74	<b>64,78</b>
	Empilhamento	61,50	61,03	53,99	54,46	64,31
LA	Melhor Único	61,00 (4,26)	52,88 (4,68)	44,06 (3,70)	53,86 (4,89)	47,70 (4,27)
	Soma	61,97	57,27	59,15	57,74	58,21
	Produto	60,56	55,86	58,68	58,21	57,74
	Empilhamento	<b>62,44</b>	56,86	49,76	56,33	58,68
SLA	Melhor Único	60,78 (2,81)	46,03 (2,87)	46,18 (4,92)	46,49 (2,68)	44,97 (4,82)
	Soma	<b>61,03</b>	54,46	54,46	53,05	56,33
	Produto	<b>61,03</b>	53,05	53,99	53,05	56,33
	Empilhamento	<b>61,03</b>	56,80	48,35	57,27	59,62

Tabela 33 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	61,41 (3,48)	48,06 (3,97)	42,73 (4,19)	48,80 (4,34)	44,70 (3,95)
	Soma	60,56	60,56	60,56	58,68	58,21
	Produto	61,03	<b>62,91</b>	62,44	60,09	58,21
	Empilhamento	61,03	60,09	47,78	58,68	60,09
L	Melhor Único	59,14 (2,73)	48,96 (4,26)	42,51 (4,01)	48,30 (3,86)	43,54 (4,12)
	Soma	61,97	60,56	58,21	57,27	58,68
	Produto	61,50	60,56	58,68	58,21	58,68
	Empilhamento	<b>63,38</b>	58,68	47,88	57,27	60,09
LA	Melhor Único	58,55 (2,80)	48,09 (3,76)	41,91 (3,18)	49,35 (4,28)	41,51 (3,60)
	Soma	61,50	61,03	60,56	61,03	58,68
	Produto	62,44	59,62	59,15	61,03	59,15
	Empilhamento	<b>62,91</b>	59,15	48,82	59,15	58,21
SL	Melhor Único	60,66 (2,80)	49,26 (4,24)	43,08 (3,60)	50,30 (3,85)	43,97 (3,53)
	Soma	59,62	58,68	55,86	58,21	55,39
	Produto	59,62	59,15	56,68	60,56	55,86
	Empilhamento	<b>61,50</b>	57,74	46,00	58,68	58,68
LA	Melhor Único	60,33 (2,67)	50,26 (3,33)	44,99 (3,32)	50,74 (3,58)	44,85 (3,19)
	Soma	62,91	57,27	61,03	57,74	60,09
	Produto	<b>63,38</b>	58,68	60,56	59,62	59,62
	Empilhamento	62,91	58,68	52,58	55,86	61,50
SLA	Melhor Único	60,42 (3,53)	56,97 (5,29)	42,47 (4,48)	55,95 (5,11)	43,79 (3,63)
	Soma	60,56	56,33	54,92	55,86	53,52
	Produto	<b>61,50</b>	58,68	55,39	55,39	53,52
	Empilhamento	61,03	58,68	48,82	58,21	53,52

Tabela 34 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,16 (3,57)	45,44 (3,81)	40,53 (4,02)	46,49 (3,87)	40,90 (3,44)
	Soma	60,09	60,56	56,80	56,80	54,46
	Produto	<b>60,56</b>	59,15	56,80	59,15	54,92
	Empilhamento	59,62	60,56	43,19	57,27	59,62
L	Melhor Único	59,21 (2,76)	47,30 (3,94)	41,52 (3,45)	47,41 (3,99)	42,89 (3,96)
	Soma	62,44	61,50	60,56	61,03	60,56
	Produto	<b>62,91</b>	61,50	61,03	60,09	60,56
	Empilhamento	60,09	59,15	50,70	62,44	<b>63,38</b>
SA	Melhor Único	62,36 (3,02)	50,42 (4,13)	49,65 (4,36)	49,83 (3,74)	47,18 (4,77)
	Soma	61,50	61,50	58,21	60,56	60,09
	Produto	<b>62,44</b>	59,15	59,15	61,97	59,15
	Empilhamento	60,56	59,15	48,35	61,50	62,44
SL	Melhor Único	60,20 (3,01)	47,37 (3,50)	43,95 (4,34)	46,33 (3,43)	43,10 (4,00)
	Soma	61,50	59,62	61,03	58,68	60,56
	Produto	60,09	61,03	<b>61,97</b>	58,68	60,56
	Empilhamento	58,68	60,56	50,70	59,62	<b>61,97</b>
LA	Melhor Único	60,28 (3,35)	51,87 (4,84)	43,02 (3,97)	52,31 (4,95)	44,81 (3,49)
	Soma	60,09	57,74	59,15	57,27	60,09
	Produto	59,62	57,77	60,56	57,77	59,62
	Empilhamento	60,56	54,46	53,99	56,80	<b>61,97</b>
SLA	Melhor Único	60,24 (4,09)	52,49 (4,58)	44,93 (4,42)	54,38 (4,54)	44,86 (4,36)
	Soma	<b>60,56</b>	59,15	58,21	58,21	55,39
	Produto	<b>60,56</b>	58,68	60,09	58,21	55,39
	Empilhamento	58,68	57,27	48,82	57,74	59,62

Tabela 35 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,50 (3,24)	55,31 (4,24)	40,76 (3,63)	56,67 (4,22)	45,18 (4,12)
	Soma	59,15	61,50	60,09	59,15	58,21
	Produto	59,15	57,27	59,62	59,15	57,27
	Empilhamento	59,62	58,68	43,66	61,97	<b>63,38</b>
L	Melhor Único	60,42 (4,00)	51,37 (4,26)	44,74 (3,94)	51,89 (4,30)	42,00 (3,33)
	Soma	60,09	61,03	60,56	61,03	62,44
	Produto	60,09	61,97	60,56	61,03	62,91
	Empilhamento	61,03	60,56	48,82	61,97	<b>66,66</b>
SA	Melhor Único	61,19 (3,19)	49,54 (4,05)	44,45 (3,60)	50,47 (4,14)	42,71 (3,57)
	Soma	61,97	61,32	62,91	59,62	61,50
	Produto	61,97	60,56	<b>64,78</b>	60,09	61,97
	Empilhamento	60,56	59,62	53,05	60,56	63,84
SL	Melhor Único	61,47 (3,41)	52,63 (4,40)	43,50 (3,74)	51,72 (4,21)	44,38 (4,40)
	Soma	61,03	61,97	57,74	61,03	60,09
	Produto	61,50	60,56	57,27	62,91	60,09
	Empilhamento	62,44	59,62	43,66	62,91	<b>63,84</b>
LA	Melhor Único	60,61 (3,38)	50,05 (3,38)	47,23 (4,45)	52,88 (3,52)	48,14 (4,90)
	Soma	60,56	61,03	61,97	61,03	<b>62,91</b>
	Produto	60,56	60,09	62,44	61,50	<b>62,91</b>
	Empilhamento	60,03	61,97	51,17	61,50	61,50
SLA	Melhor Único	61,35 (3,37)	51,62 (4,36)	44,46 (4,36)	52,94 (4,11)	47,41 (4,20)
	Soma	61,50	62,91	59,15	<b>63,38</b>	59,15
	Produto	61,50	61,97	59,62	61,97	59,62
	Empilhamento	61,50	58,68	48,35	60,09	62,44

Tabela 36 – Acurácia considerando o protocolo LOSO e um *pool* com 70 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	57,12 (3,24)	48,58 (4,44)	43,68 (3,81)	52,19 (4,65)	43,70 (4,12)
	Soma	59,62	58,21	63,84	58,21	62,44
	Produto	56,62	58,21	62,44	58,68	62,44
	Empilhamento	59,62	62,44	46,00	62,44	<b>65,72</b>
L	Melhor Único	60,99 (3,43)	47,59 (4,01)	42,66 (3,55)	50,78 (4,25)	45,16 (4,50)
	Soma	61,97	61,50	60,56	58,68	60,09
	Produto	61,03	58,68	59,15	60,09	60,09
	Empilhamento	61,50	60,09	48,82	62,44	<b>64,31</b>
SA	Melhor Único	61,03 (3,00)	49,57 (4,30)	46,72 (3,84)	51,44 (4,53)	43,91 (3,84)
	Soma	61,50	63,38	59,15	<b>63,84</b>	<b>63,84</b>
	Produto	61,97	60,56	59,62	<b>63,84</b>	62,44
	Empilhamento	61,03	62,44	49,76	58,68	62,44
SL	Melhor Único	61,35 (3,46)	50,50 (4,54)	44,97 (3,92)	51,47 (4,28)	46,72 (4,12)
	Soma	61,97	61,50	61,03	60,09	60,09
	Produto	61,50	60,09	61,50	59,15	59,15
	Empilhamento	61,03	57,74	48,35	<b>62,44</b>	<b>62,44</b>
LA	Melhor Único	61,84	50,76 (3,98)	45,64 (4,29)	50,93 (4,20)	46,73 (4,28)
	Soma	61,03	58,68	59,15	59,62	61,50
	Produto	61,03	59,15	59,62	59,62	61,97
	Empilhamento	61,97	58,21	47,88	59,15	<b>63,84</b>
SLA	Melhor Único	61,68 (2,91)	49,95 (4,37)	43,23 (4,10)	49,49 (4,38)	43,08 (4,01)
	Soma	61,50	60,56	58,68	60,56	61,03
	Produto	61,50	60,56	60,09	60,09	61,03
	Empilhamento	<b>61,97</b>	61,50	50,70	60,56	61,03

Tabela 37 – Acurácia considerando o protocolo LOSO e um *pool* com 100 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	59,82 (3,38)	46,16 (3,63)	42,68 (4,17)	49,57 (3,89)	41,73 (3,62)
	Soma	61,03	56,33	61,03	55,86	59,15
	Produto	60,56	57,27	<b>62,44</b>	57,27	59,15
	Empilhamento	60,09	55,86	46,94	58,21	59,15
L	Melhor Único	61,48 (3,08)	48,67 (4,69)	42,93 (3,76)	49,94 (4,54)	44,70 (4,00)
	Soma	59,62	60,56	61,50	59,62	<b>62,91</b>
	Produto	60,09	59,15	60,56	59,62	62,44
	Empilhamento	61,03	58,68	46,00	62,44	<b>62,91</b>
SA	Melhor Único	62,34 (3,11)	51,54 (4,26)	45,17 (3,98)	53,90 (4,17)	44,14 (3,82)
	Soma	62,91	<b>63,84</b>	60,09	61,50	59,15
	Produto	62,44	61,03	60,09	60,56	60,09
	Empilhamento	61,50	60,56	48,35	60,09	61,97
SL	Melhor Único	60,08 (3,27)	49,87 (4,74)	42,24 (3,90)	47,62 (4,31)	41,93 (3,94)
	Soma	59,15	56,80	<b>63,84</b>	55,39	60,56
	Produto	59,15	57,27	63,38	57,27	60,56
	Empilhamento	60,56	56,33	46,94	60,56	64,78
LA	Melhor Único	62,72 (3,32)	51,89 (4,56)	45,50 (3,83)	53,44 (4,79)	44,51 (4,06)
	Soma	<b>62,44</b>	61,97	59,62	59,62	61,03
	Produto	61,50	56,33	61,03	58,68	61,03
	Empilhamento	61,03	60,09	48,82	61,03	61,03
SLA	Melhor Único	60,11 (3,45)	50,61 (4,03)	46,60 (4,08)	52,39 (3,88)	45,23
	Soma	61,03	<b>62,44</b>	61,97	61,97	61,03
	Produto	61,03	61,03	61,03	61,97	61,03
	Empilhamento	60,09	61,50	50,23	59,15	59,62

Tabela 38 – Acurácia considerando o protocolo LOSO e um *pool* com 150 representações geradas com o método proposto. Kyoto e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Dinamic Selection (KU)	
			BG	RF	DT	RF
S	Melhor Único	<b>63,30</b> (3,49)	50,45 (4,05)	46,31 (3,99)	50,08 (3,74)	44,35 (3,77)
	Soma	57,74	59,62	60,09	57,74	58,68
	Produto	57,74	57,74	59,62	58,21	58,21
	Empilhamento	58,68	58,68	45,07	60,56	62,91
L	Melhor Único	63,19 (4,96)	52,47 (4,43)	45,97 (4,22)	49,17 (4,08)	44,37 (3,94)
	Soma	60,56	60,09	59,15	60,09	59,15
	Produto	61,50	54,46	59,62	59,62	59,62
	Empilhamento	59,15	57,27	46,94	61,50	<b>66,19</b>
SA	Melhor Único	63,63 (4,30)	52,20 (4,37)	46,18 (4,35)	51,64 (4,15)	44,84 (4,25)
	Soma	61,50	60,56	62,44	61,03	61,50
	Produto	61,50	56,33	62,91	61,03	61,50
	Empilhamento	60,09	61,97	52,11	60,56	<b>65,72</b>
SL	Melhor Único	64,69 (3,82)	50,37 (4,00)	46,45 (3,43)	49,51 (3,73)	45,17(3,54)
	Soma	59,62	60,09	61,97	61,03	60,09
	Produto	60,56	56,33	61,97	61,03	60,56
	Empilhamento	58,21	59,62	48,35	62,44	<b>64,78</b>
LA	Melhor Único	64,44 (3,46)	51,42 (4,06)	47,10 (4,15)	52,24 (4,10)	48,30 (4,01)
	Soma	61,03	59,15	61,03	58,68	61,97
	Produto	61,50	58,21	61,50	58,21	61,97
	Empilhamento	51,64	61,03	51,64	61,50	<b>65,25</b>
SLA	Melhor Único	64,08 (3,36)	52,81 (4,14)	46,73 (3,63)	52,25 (4,20)	47,75 (4,07)
	Soma	61,03	58,68	63,84	59,62	62,44
	Produto	61,50	56,80	64,31	59,15	62,44
	Empilhamento	61,97	61,50	52,58	59,62	<b>64,31</b>

Tabela 39 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	<b>54,24 (1,39)</b>	44,08 (3,00)	37,53 (2,53)	43,58 (2,75)	36,26 (1,60)
	Soma	50,23	52,11	46,00	52,58	47,88
	Produto	48,82	51,17	46,94	52,58	48,82
	Empilhamento	50,70	50,70	45,07	47,88	43,66
A	Melhor Único	59,31 (3,25)	46,06 (3,92)	38,16 (3,31)	47,47 (5,79)	39,50 (2,35)
	Soma	59,15	51,17	49,29	52,58	45,53
	Produto	<b>61,50</b>	50,23	47,41	52,58	45,07
	Empilhamento	60,56	51,17	47,88	48,35	46,94
L	Melhor Único	59,98 (3,46)	40,03 (1,65)	33,75 (1,93)	40,79 (1,25)	37,52 (1,35)
	Soma	<b>60,09</b>	51,17	46,94	49,76	48,35
	Produto	<b>60,09</b>	52,58	47,41	50,23	48,82
	Empilhamento	59,62	52,58	41,31	46,00	47,88
SA	Melhor Único	60,46 (4,29)	46,82 (3,17)	43,09 (2,01)	45,89 (3,91)	40,85 (1,58)
	Soma	60,56	60,09	56,33	58,68	54,92
	Produto	<b>61,03</b>	60,09	56,33	59,15	55,86
	Empilhamento	57,74	59,15	48,82	54,92	54,46
SL	Melhor Único	56,55 (2,65)	50,54 (4,32)	40,87 (3,68)	46,81 (3,16)	38,18 (2,01)
	Soma	57,27	51,64	51,17	52,11	49,29
	Produto	55,86	53,05	49,76	52,11	49,29
	Empilhamento	<b>56,80</b>	52,11	50,23	54,46	51,64
LA	Melhor Único	60,80 (3,97)	46,59 (1,85)	37,42 (1,74)	45,19 (1,31)	38,33 (1,74)
	Soma	60,09	58,68	48,82	54,92	48,35
	Produto	60,56	58,68	47,88	55,86	47,41
	Empilhamento	<b>61,50</b>	55,86	42,72	53,99	47,88
SLA	Melhor Único	58,37 (1,94)	47,80 (3,46)	38,19 (3,47)	47,48 (2,68)	40,62 (4,25)
	Soma	58,68	54,46	46,94	54,46	47,41
	Produto	58,21	52,58	47,88	4,92	47,88
	Empilhamento	<b>59,15</b>	53,99	43,66	51,64	49,29

Tabela 40 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	56,19 (2,59)	48,50 (4,25)	38,84 (3,19)	47,22 (3,81)	37,34 (3,99)
	Soma	54,92	57,27	56,33	57,74	56,33
	Produto	54,92	59,15	54,46	59,15	55,86
	Empilhamento	56,33	<b>58,68</b>	44,13	56,80	53,52
L	Melhor Único	61,21 (3,35)	49,22 (4,92)	43,79 (4,34)	52,51 (5,37)	42,96 (4,29)
	Soma	<b>61,97</b>	55,86	55,86	56,80	59,62
	Produto	61,50	56,80	55,39	57,27	59,15
	Empilhamento	60,09	58,21	47,41	56,80	58,68
SA	Melhor Único	60,78 (4,17)	45,65 (3,84)	39,83 (3,04)	45,39 (3,72)	39,26 (3,73)
	Soma	60,09	53,99	52,58	52,11	53,99
	Produto	<b>61,50</b>	54,46	52,58	52,58	53,52
	Empilhamento	59,62	53,99	44,60	50,70	54,46
SL	Melhor Único	58,05 (5,10)	43,63 (3,17)	43,44 (4,77)	44,94 (3,64)	47,80 (5,56)
	Soma	56,80	54,46	54,46	53,52	56,80
	Produto	57,27	52,58	54,46	54,92	55,86
	Empilhamento	<b>58,21</b>	54,46	47,88	52,11	<b>58,21</b>
LA	Melhor Único	61,27 (2,83)	49,25 (3,11)	46,66 (4,18)	51,44 (3,92)	47,14 (4,47)
	Soma	60,56	57,27	55,86	58,86	54,46
	Produto	61,03	58,68	57,74	56,33	53,99
	Empilhamento	<b>61,97</b>	59,62	52,11	60,09	52,11
SLA	Melhor Único	61,23 (2,69)	49,07 (5,16)	42,81 (2,87)	51,32 (5,41)	44,05 (3,98)
	Soma	<b>61,97</b>	58,21	55,86	54,92	57,27
	Produto	61,50	56,33	56,80	55,39	56,80
	Empilhamento	60,56	55,86	50,23	53,99	55,86

Tabela 41 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	56,84 (2,63)	44,89 (2,48)	40,25 (2,38)	48,56 (3,09)	39,93 (2,86)
	Soma	56,33	56,33	<b>59,62</b>	56,80	57,27
	Produto	55,86	56,80	<b>59,62</b>	57,27	56,80
	Empilhamento	57,74	57,27	48,82	56,80	58,21
L	Melhor Único	59,55 (3,95)	47,55 (3,95)	41,66 (2,82)	48,01 (4,08)	41,09 (3,09)
	Soma	60,09	57,74	59,15	55,39	57,74
	Produto	60,56	57,27	58,68	56,80	57,27
	Empilhamento	<b>60,09</b>	55,86	52,58	52,11	59,62
SA	Melhor Único	62,19 (3,72)	50,79 (4,81)	43,21 (4,35)	49,90 (4,65)	49,01 (4,31)
	Soma	62,44	57,74	61,03	57,74	61,50
	Produto	<b>62,91</b>	57,27	60,09	59,15	61,50
	Empilhamento	61,50	58,68	52,11	57,74	60,56
SL	Melhor Único	58,37 (2,14)	47,62 (4,47)	43,67 (3,84)	47,59 (4,59)	41,79 (3,70)
	Soma	58,21	56,80	55,86	56,33	55,86
	Produto	58,21	58,21	57,27	57,27	56,33
	Empilhamento	<b>59,15</b>	56,33	43,66	54,92	56,33
LA	Melhor Único	60,42 (2,06)	47,35 (2,77)	42,61 (3,92)	49,71 (2,95)	44,51 (3,79)
	Soma	60,09	58,21	58,21	55,39	58,68
	Produto	<b>60,56</b>	57,27	57,74	55,86	58,68
	Empilhamento	60,09	56,80	47,88	54,46	57,74
SLA	Melhor Único	62,23 (2,18)	51,69 (3,59)	47,47 (4,00)	50,78 (3,56)	45,06 (3,71)
	Soma	61,97	56,33	57,74	57,74	59,62
	Produto	<b>62,91</b>	54,46	57,27	58,21	59,62
	Empilhamento	61,03	59,15	48,82	55,86	61,50

Tabela 42 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	55,05 (2,22)	47,75 (4,55)	42,27 (3,94)	47,23 (3,79)	44,98 (4,11)
	Soma	55,39	59,62	59,62	59,15	59,15
	Produto	56,33	60,56	60,09	61,03	59,62
	Empilhamento	58,86	<b>61,03</b>	51,17	57,27	59,62
L	Melhor Único	60,30 (2,99)	48,07 (3,33)	44,17 (4,67)	48,55 (3,11)	43,04 (3,72)
	Soma	58,68	59,62	57,74	58,68	53,99
	Produto	59,62	57,74	55,86	57,74	53,99
	Empilhamento	<b>60,56</b>	57,74	46,47	55,86	53,05
SA	Melhor Único	60,23 (3,98)	51,06 (5,32)	42,16 (3,04)	49,20 (4,78)	41,59 (3,26)
	Soma	59,62	58,21	58,21	57,27	58,21
	Produto	59,62	59,15	56,80	58,68	57,74
	Empilhamento	60,56	56,33	49,29	55,39	59,15
SL	Melhor Único	58,61 (2,48)	47,95 (3,50)	42,19 (4,20)	46,34 (3,78)	42,36 (4,51)
	Soma	56,80	<b>62,91</b>	56,80	62,44	57,27
	Produto	57,77	59,62	57,74	61,97	57,74
	Empilhamento	58,68	61,97	46,00	62,44	61,03
LA	Melhor Único	60,22 (3,59)	49,61 (4,01)	41,67 (2,91)	51,42 (3,57)	42,29 (2,88)
	Soma	<b>60,56</b>	56,80	57,27	56,33	56,33
	Produto	60,09	55,39	56,80	55,86	56,33
	Empilhamento	59,62	55,39	48,35	54,46	58,21
SLA	Melhor Único	61,73 (4,09)	52,00 (3,74)	46,65 (3,73)	52,52 (3,53)	48,03 (4,03)
	Soma	<b>62,44</b>	61,97	58,68	60,09	58,21
	Produto	<b>62,44</b>	61,50	59,62	61,03	58,21
	Empilhamento	61,97	61,97	51,17	58,68	60,56

Tabela 43 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	55,75 (4,85)	48,16 (4,26)	41,21 (4,23)	46,89 (3,99)	41,65 (3,46)
	Soma	55,86	57,74	59,15	57,27	58,21
	Produto	55,39	57,27	57,74	56,80	57,27
	Empilhamento	55,86	57,27	44,13	57,74	<b>61,03</b>
L	Melhor Único	58,25 (2,83)	46,47 (3,49)	45,51 (4,46)	46,83 (3,50)	45,81 (4,04)
	Soma	58,21	61,50	61,97	61,97	60,56
	Produto	58,21	60,56	61,03	60,09	60,56
	Empilhamento	58,68	62,44	48,35	62,44	<b>62,91</b>
SA	Melhor Único	61,95 (3,45)	55,68 (5,23)	46,62 (4,16)	55,73 (4,85)	45,41 (3,74)
	Soma	63,38	62,44	<b>63,84</b>	61,97	63,38
	Produto	62,44	<b>63,84</b>	63,38	61,97	62,91
	Empilhamento	61,50	<b>63,84</b>	52,58	61,50	62,91
SL	Melhor Único	59,06 (4,20)	47,88 (5,02)	42,03 (5,17)	47,95 (4,27)	41,80 (3,89)
	Soma	59,15	61,50	60,09	61,50	58,68
	Produto	59,62	59,62	61,03	61,03	59,15
	Empilhamento	59,15	<b>61,97</b>	44,60	<b>61,97</b>	61,03
LA	Melhor Único	60,40 (4,35)	53,03 (4,95)	46,43 (4,57)	53,94 (4,49)	46,43 (4,60)
	Soma	60,56	59,62	61,50	56,33	61,97
	Produto	61,03	58,21	<b>63,38</b>	57,27	61,50
	Empilhamento	59,62	55,86	51,64	58,68	62,44
SLA	Melhor Único	62,32 (3,68)	50,55 (4,13)	42,38 (3,59)	51,31 (3,93)	42,32 (3,66)
	Soma	61,50	58,21	57,74	57,27	55,86
	Produto	<b>62,44</b>	56,80	59,15	58,21	56,33
	Empilhamento	60,56	58,68	48,35	60,09	59,15

Tabela 44 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Tabela 45

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	56,07 (2,82)	47,08 (4,57)	41,53 (3,96)	46,48 (4,79)	43,12 (3,88)
	Soma	53,99	56,80	<b>61,03</b>	53,99	58,21
	Produto	53,52	56,80	60,56	54,92	58,68
	Empilhamento	56,33	54,46	44,13	56,80	56,80
L	Melhor Único	59,98 (3,60)	47,73 (3,54)	43,13 (3,19)	51,89 (3,31)	43,47 (3,25)
	Soma	59,62	58,68	60,56	57,27	<b>61,97</b>
	Produto	60,09	57,74	58,68	58,21	61,50
	Empilhamento	60,09	59,62	46,47	57,27	63,38
SA	Melhor Único	61,49 (3,05)	50,07 (4,44)	45,93 (4,83)	51,24 (4,43)	44,72 (3,65)
	Soma	61,50	60,09	59,62	60,09	60,56
	Produto	61,03	58,68	61,50	60,03	59,62
	Empilhamento	59,62	60,56	51,17	<b>61,97</b>	61,50
SL	Melhor Único	59,01 (3,96)	48,94 (4,39)	47,39 (3,79)	50,37 (4,24)	42,33 (3,51)
	Soma	57,74	61,03	61,03	58,21	59,15
	Produto	58,21	56,80	61,03	59,62	59,15
	Empilhamento	59,15	59,15	50,70	56,80	<b>64,31</b>
LA	Melhor Único	61,09 (4,97)	49,87 (4,07)	43,39 (4,30)	49,01 (3,94)	45,79 (4,94)
	Soma	60,09	59,15	61,03	57,27	<b>62,91</b>
	Produto	61,03	61,03	61,03	56,80	62,44
	Empilhamento	59,62	61,97	51,17	61,50	61,03
SLA	Melhor Único	61,48 (3,56)	54,04 (5,19)	49,04 (4,38)	52,72 (4,88)	50,00 (4,28)
	Soma	<b>61,50</b>	60,09	59,62	58,68	60,09
	Produto	<b>61,50</b>	59,62	59,62	60,09	60,09
	Empilhamento	58,68	61,50	48,35	61,03	<b>61,50</b>

Tabela 46 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Tabela 47

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Dinamic Selection (KU)	
			BG	RF	DT	RF
S	Melhor Único	57,08 (3,32)	50,53 (4,19)	44,30 (5,23)	50,79 (4,52)	42,44 (4,77)
	Soma	55,86	61,97	59,15	61,03	61,03
	Produto	57,27	61,03	59,62	62,44	60,56
	Empilhamento	56,33	59,15	50,70	62,44	<b>62,44</b>
L	Melhor Único	58,96 (5,56)	48,66 (4,36)	42,62 (3,84)	49,58 (4,30)	43,33 (3,59)
	Soma	59,15	56,33	58,21	56,33	57,27
	Produto	58,68	56,33	58,21	55,86	56,80
	Empilhamento	58,68	55,86	42,25	58,21	<b>60,56</b>
SA	Melhor Único	61,36 (3,20)	49,58 (3,96)	43,97 (4,72)	49,06 (3,94)	44,30 (4,24)
	Soma	60,56	60,56	62,91	60,09	63,84
	Produto	61,50	58,68	62,91	60,56	63,38
	Empilhamento	58,68	62,44	53,52	59,62	<b>64,78</b>
SL	Melhor Único	59,44 (2,98)	46,35 (3,11)	44,18 (3,90)	47,86 (3,12)	44,31 (3,54)
	Soma	58,21	61,50	60,09	59,62	59,15
	Produto	57,74	55,86	60,56	59,62	59,15
	Empilhamento	59,62	59,15	46,94	<b>63,38</b>	62,44
LA	Melhor Único	60,40 (3,51)	50,82 (5,00)	45,58 (4,65)	48,72 (4,96)	43,41 (4,61)
	Soma	59,62	57,27	59,15	59,62	58,68
	Produto	60,09	58,68	60,09	58,68	58,21
	Empilhamento	60,56	61,00	47,41	61,03	<b>62,44</b>
SLA	Melhor Único	60,78 (3,89)	50,15 (3,98)	44,12 (3,84)	52,52 (4,05)	44,67 (4,14)
	Soma	60,56	59,62	60,09	60,56	60,56
	Produto	61,03	61,03	59,62	60,56	60,56
	Empilhamento	59,62	61,03	50,23	61,97	<b>65,25</b>

Tabela 48 – Acurácia considerando o protocolo LOSO e um *pool* com 70 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,16 (3,67)	48,65 (4,08)	43,22 (3,82)	50,86 (4,06)	43,09 (3,87)
	Soma	57,74	56,33	60,56	54,46	56,80
	Produto	58,21	58,68	<b>61,97</b>	56,33	57,27
	Empilhamento	57,74	59,15	46,47	59,15	61,03
L	Melhor Único	60,46 (4,68)	50,99 (4,90)	42,62 (4,20)	50,49 (4,57)	43,00 (4,09)
	Soma	60,09	61,97	61,03	60,09	58,21
	Produto	59,62	58,68	61,97	61,50	58,21
	Empilhamento	60,56	59,62	47,41	<b>64,78</b>	61,50
SA	Melhor Único	61,00 (3,41)	50,28 (4,04)	43,38 (3,68)	51,63 (4,04)	46,33 (4,06)
	Soma	60,09	<b>62,44</b>	61,97	61,03	58,68
	Produto	61,03	59,62	<b>62,44</b>	61,97	59,15
	Empilhamento	60,09	60,56	46,94	57,27	61,97
SL	Melhor Único	60,00 (3,62)	53,83 (4,04)	46,73 (4,19)	49,93 (3,57)	44,50 (3,90)
	Soma	59,62	60,09	60,56	60,09	59,62
	Produto	60,09	60,09	60,56	60,09	59,62
	Empilhamento	58,68	60,56	47,88	62,44	<b>64,78</b>
LA	Melhor Único	62,07 (3,79)	52,30 (4,57)	48,13 (4,30)	51,71 (4,55)	46,58 (3,87)
	Soma	62,44	60,09	62,44	59,15	61,97
	Produto	61,50	58,68	61,03	59,62	61,97
	Stacking	61,03	60,56	49,76	60,09	<b>63,84</b>
SLA	Melhor Único	60,84 (3,64)	53,27 (4,00)	44,07 (3,80)	55,17 (4,18)	47,25 (4,34)
	Soma	60,56	61,50	61,03	61,03	61,03
	Produto	61,03	60,09	61,97	60,09	61,03
	Empilhamento	60,09	58,21	51,17	60,09	<b>64,31</b>

Tabela 49 – Acurácia considerando o protocolo LOSO e um *pool* com 100 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,27 (4,02)	48,19 (3,93)	47,26 (4,19)	48,57 (3,97)	45,16 (4,36)
	Soma	55,39	59,15	60,09	57,27	57,27
	Produto	56,33	57,27	59,62	59,62	57,74
	Empilhamento	58,21	60,56	47,41	60,56	<b>61,03</b>
L	Melhor Único	60,51 (4,32)	49,28 (4,87)	45,02 (4,46)	50,90 (4,82)	44,64 (4,18)
	Soma	60,09	<b>60,56</b>	59,62	<b>60,56</b>	59,62
	Produto	<b>60,56</b>	56,80	59,62	60,09	59,15
	Empilhamento	59,15	57,27	46,00	60,09	60,56
SA	Melhor Único	61,09 (3,74)	51,67 (4,13)	48,03 (3,73)	51,98 (4,07)	46,15 (4,09)
	Soma	61,50	61,50	61,50	60,56	60,56
	Produto	61,97	57,27	62,44	60,56	60,56
	Empilhamento	60,56	61,50	49,29	60,09	<b>63,84</b>
SL	Melhor Único	59,16 (3,67)	51,46 (4,15)	44,65 (3,88)	52,38 (4,13)	45,30 (3,59)
	Soma	60,09	59,62	61,50	58,68	60,56
	Produto	60,09	58,21	61,05	60,09	60,56
	Empilhamento	58,68	57,27	47,88	<b>62,44</b>	<b>62,44</b>
LA	Melhor Único	60,99 (2,74)	50,71 (3,73)	46,30 (3,94)	53,51 (3,91)	48,18 (4,02)
	Soma	61,03	58,21	59,62	58,21	59,62
	Produto	61,03	56,80	60,09	59,15	59,62
	Empilhamento	60,09	60,09	50,70	60,56	61,50
SLA	Melhor Único	61,27 (3,77)	52,23 (4,27)	47,77 (4,25)	53,68 (3,99)	48,89 (4,20)
	Soma	61,50	61,03	<b>64,78</b>	60,09	63,38
	Produto	61,03	57,74	64,31	61,03	62,44
	Empilhamento	59,62	60,09	50,23	61,03	63,38

Tabela 50 – Acurácia considerando o protocolo LOSO e um *pool* com 150 representações geradas com o método proposto. LabelMe e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,79 (3,54)	53,71 (4,27)	43,89 (4,01)	54,21 (4,19)	43,12 (3,97)
	Soma	57,74	60,09	61,03	60,56	57,27
	Produto	59,62	56,80	60,56	59,62	57,74
	Empilhamento	59,62	55,86	45,53	<b>61,97</b>	60,56
L	Melhor Único	60,92 (3,74)	49,49 (3,91)	47,00 (4,21)	58,38 (3,85)	44,75 (3,84)
	Soma	60,56	60,09	62,91	59,62	61,50
	Produto	60,56	57,27	<b>63,38</b>	60,56	61,97
	Empilhamento	61,03	58,68	47,41	63,38	64,31
SA	Melhor Único	61,95 (3,26)	51,33 (4,07)	46,61 (4,08)	51,90 (3,95)	47,80 (4,28)
	Soma	62,44	61,50	61,97	61,50	63,38
	Produto	62,44	57,27	62,91	61,03	62,91
	Empilhamento	58,68	60,50	49,76	61,97	64,31
SL	Melhor Único	63,25(4,45)	49,95 (4,32)	44,93 (3,82)	50,53 (4,12)	43,94 (3,59)
	Soma	59,62	59,62	63,38	59,15	59,15
	Produto	60,09	57,74	63,38	59,15	59,62
	tacking	60,56	58,21	48,82	63,38	<b>63,84</b>
LA	Melhor Único	61,30 (3,33)	54,26 (4,43)	45,63 (3,67)	53,25 (4,50)	46,68 (3,87)
	Soma	61,50	61,50	61,03	61,50	61,03
	Produto	61,50	56,80	<b>62,44</b>	61,50	61,97
	Empilhamento	60,56	61,03	50,23	60,56	<b>62,44</b>
SLA	Melhor Único	61,01 (2,92)	53,58 (4,32)	46,81 (4,39)	54,79 (4,35)	45,82 (4,16)
	Soma	61,03	61,50	61,03	61,50	62,91
	Produto	61,03	61,03	60,56	61,03	62,91
	Empilhamento	60,56	<b>63,38</b>	47,88	61,97	62,91

Tabela 51 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,04 (4,03)	40,18 (1,82)	34,12 (0,85)	40,16(1,87)	34,44 (2,62)
	Soma	<b>65,20</b>	52,11	51,17	51,64	48,35
	Produto	61,97	52,11	49,76	52,11	48,35
	Empilhamento	64,31	52,11	43,66	55,39	49,29
A	Melhor Único	55,36 (1,36)	54,51 (6,81)	42,65 (3,70)	55,87 (7,18)	43,58 (4,15)
	Soma	58,21	54,92	49,76	55,39	48,82
	Produto	58,21	55,39	49,76	55,39	48,35
	Empilhamento	<b>58,68</b>	53,99	47,41	52,52	52,58
L	Melhor Único	58,96 (3,31)	47,76 (4,75)	34,99 (3,93)	46,48 (4,00)	38,34 (4,00)
	Soma	59,62	54,46	49,29	52,11	48,82
	Produto	<b>60,56</b>	53,05	50,70	53,99	48,82
	Empilhamento	59,15	52,11	43,19	50,23	47,41
SA	Melhor Único	55,95 (1,64)	48,52 (2,65)	43,31 (2,53)	47,30 (1,64)	42,40 (2,63)
	Soma	59,15	59,62	57,27	56,33	56,33
	Produto	60,09	58,21	58,21	56,80	57,74
	Empilhamento	58,68	58,68	54,46	53,99	57,74
SL	Melhor Único	58,59 (2,85)	46,25 (5,33)	40,66 (3,70)	47,68 (5,54)	41,67 (5,05)
	Soma	<b>61,03</b>	54,46	49,29	52,58	49,76
	Produto	59,62	53,05	48,82	53,99	49,29
	Empilhamento	60,56	53,99	45,53	52,58	48,35
LA	Melhor Único	62,13 (3,34)	46,22 (2,97)	39,30 (3,58)	45,77 (2,69)	40,75 (2,80)
	Soma	<b>63,38</b>	53,99	52,58	52,11	49,29
	Produto	62,91	53,52	53,52	53,05	49,29
	Empilhamento	62,44	52,11	50,23	51,64	50,23
SLA	Melhor Único	62,20 (7,11)	48,18 (5,38)	40,39 (3,93)	49,09 (5,51)	39,76 (3,08)
	Soma	61,03	53,52	48,26	53,99	48,82
	Produto	61,03	53,52	46,47	54,46	48,82
	Empilhamento	<b>62,91</b>	54,92	44,13	49,29	49,76

Tabela 52 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,52 (3,65)	44,99 (3,23)	38,93 (2,65)	42,72 (3,08)	38,73 (2,67)
	Soma	58,68	53,99	50,70	52,58	51,64
	Produto	60,09	53,52	51,64	53,52	51,17
	Empilhamento	<b>60,56</b>	55,39	44,13	51,17	53,52
L	Melhor Único	60,56 (3,82)	48,04 (3,84)	47,26 (5,11)	49,42 (3,96)	47,26 (4,83)
	Soma	61,03	58,21	56,80	58,21	<b>61,97</b>
	Produto	60,56	60,56	57,74	58,68	61,03
	Empilhamento	59,15	56,80	48,82	56,33	60,09
SA	Melhor Único	61,39 (2,99)	46,33 (5,04)	39,72 (3,35)	50,34 (4,12)	42,87 (3,16)
	Soma	64,31	58,21	58,21	57,27	57,74
	Produto	<b>65,25</b>	58,68	60,09	57,27	58,68
	Empilhamento	63,38	58,21	51,17	54,92	53,05
SL	Melhor Único	58,53 (4,01)	42,44 (3,78)	39,54 (3,18)	42,72 (3,64)	40,44 (3,51)
	Soma	<b>60,09</b>	55,86	57,74	52,58	55,86
	Produto	59,62	57,74	56,80	52,58	55,39
	Empilhamento	59,62	54,92	46,94	49,29	55,86
LA	Melhor Único	59,17 (2,50)	50,83 (4,86)	43,97 (4,87)	51,21 (5,31)	45,38 (5,22)
	Soma	58,68	57,27	56,33	56,33	54,92
	Produto	<b>59,62</b>	58,21	56,33	57,74	54,46
	Empilhamento	59,15	58,68	52,11	56,33	57,74
SLA	Melhor Único	58,92 (3,02)	54,24 (6,89)	42,54 (5,32)	51,88 (6,38)	44,51 (4,66)
	Soma	61,97	56,80	59,15	54,92	58,21
	Produto	<b>62,44</b>	54,92	57,27	55,86	57,27
	Empilhamento	61,97	56,33	52,58	53,05	58,21

Tabela 53 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	59,03 (3,18)	50,05 (4,82)	43,05 (3,97)	48,79 (4,41)	40,08 (3,53)
	Soma	<b>62,44</b>	59,62	57,27	60,56	58,21
	Produto	61,50	60,09	57,27	61,03	57,74
	Empilhamento	61,50	60,09	48,35	57,27	57,74
L	Melhor Único	58,66 (3,12)	49,54 (3,69)	44,78 (3,89)	49,00 (4,25)	42,12 (3,72)
	Soma	60,56	<b>63,84</b>	55,39	59,62	53,52
	Produto	60,56	62,44	53,52	61,50	53,52
	Empilhamento	61,97	61,97	50,23	60,56	56,80
SA	Melhor Único	57,06 (2,36)	52,14 (4,91)	45,95 (4,27)	55,46 (5,33)	47,50 (4,41)
	Soma	61,03	59,62	58,21	57,74	55,86
	Produto	<b>62,44</b>	56,33	57,27	57,27	54,92
	Empilhamento	61,03	60,56	49,76	58,21	60,09
SL	Melhor Único	57,17 (2,74)	52,12 (4,96)	37,73 (3,71)	51,68 (4,74)	38,69 (3,16)
	Soma	58,68	58,68	59,15	59,15	54,46
	Produto	57,74	57,74	<b>61,03</b>	59,62	54,46
	Empilhamento	58,68	56,33	47,88	59,15	55,86
LA	Melhor Único	59,09 (2,70)	50,22 (3,90)	45,56 (4,45)	47,97 (3,99)	41,01 (3,60)
	Soma	61,50	56,80	60,09	58,21	60,09
	Produto	60,09	57,27	59,62	57,27	60,56
	Empilhamento	<b>62,44</b>	56,80	52,11	55,39	56,80
SLA	Melhor Único	57,66 (4,04)	48,53 (4,65)	44,24 (4,57)	49,16 (4,61)	46,67 (5,45)
	Soma	<b>61,50</b>	56,38	55,39	57,74	55,86
	Produto	60,56	54,46	54,46	55,86	55,86
	Empilhamento	60,09	58,21	53,05	54,46	59,62

Tabela 54 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,17 (3,30)	47,35 (4,19)	41,36 (4,24)	50,17 (4,48)	38,41 (3,91)
	Soma	60,09	59,15	56,33	57,74	53,99
	Produto	<b>61,03</b>	53,52	56,33	57,74	54,46
	Empilhamento	59,15	57,27	43,66	59,62	54,92
L	Melhor Único	56,81 (2,59)	46,89 (3,57)	40,92 (3,62)	48,23 (3,16)	41,32 (3,74)
	Soma	60,09	<b>62,44</b>	57,74	56,33	58,21
	Produto	60,09	60,56	57,74	59,62	58,68
	Empilhamento	59,62	56,80	45,53	55,39	60,09
SA	Melhor Único	59,40 (2,94)	50,24 (3,68)	45,15 (3,31)	49,65 (3,52)	44,67 (4,20)
	Soma	61,97	56,33	58,21	56,33	56,80
	Produto	<b>63,38</b>	57,74	57,74	59,62	57,27
	Empilhamento	61,97	58,68	46,47	55,39	58,68
SL	Melhor Único	57,69 (2,31)	47,77 (3,37)	41,84 (3,05)	47,47 (3,38)	41,23 (3,06)
	Soma	<b>61,50</b>	58,21	57,27	57,27	55,39
	Produto	58,68	59,15	57,74	57,74	55,39
	Empilhamento	<b>61,50</b>	58,21	48,35	56,33	55,39
LA	Melhor Único	61,28 (3,02)	49,65 (3,75)	43,22 (3,89)	50,80 (4,01)	47,14 (4,71)
	Soma	61,97	61,97	60,09	57,74	58,68
	Produto	61,97	61,03	58,68	58,68	58,68
	Empilhamento	<b>64,31</b>	59,15	51,17	57,27	61,97
SLA	Melhor Único	61,46 (4,36)	51,54 (4,40)	42,16 (3,82)	52,01 (4,52)	43,45 (4,00)
	Soma	60,56	60,09	57,27	59,62	59,62
	Produto	60,09	60,56	57,74	60,09	60,09
	Empilhamento	<b>61,50</b>	58,68	49,76	57,27	60,56

Tabela 55 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,79 (3,51)	49,53 (4,02)	45,83 (2,97)	49,98 (3,48)	45,29 (4,05)
	Soma	60,56	59,62	61,03	57,74	60,09
	Produto	61,03	59,62	<b>61,50</b>	58,68	60,56
	Empilhamento	60,56	60,56	45,53	56,33	58,21
L	Melhor Único	60,93 (4,04)	50,80 (3,84)	42,05 (4,11)	49,41 (3,53)	45,38 (4,17)
	Soma	61,50	60,09	61,03	58,68	60,56
	Produto	61,03	59,62	<b>62,44</b>	60,09	61,03
	Empilhamento	61,03	58,68	48,35	58,68	60,09
SA	Melhor Único	60,37 (3,21)	51,45 (4,01)	43,97 (4,38)	51,41 (4,07)	45,25 (4,43)
	Soma	61,03	60,09	61,50	59,15	58,68
	Produto	<b>63,38</b>	60,09	61,97	61,50	58,68
	Empilhamento	61,97	58,21	53,05	58,21	60,56
SL	Melhor Único	60,11 (2,49)	48,66 (4,43)	40,44 (3,21)	50,83 (4,51)	40,65 (3,55)
	Soma	60,56	56,33	58,68	59,15	<b>61,97</b>
	Produto	60,56	60,09	59,62	57,74	<b>61,97</b>
	Empilhamento	59,15	56,80	50,23	58,68	61,03
LA	Melhor Único	59,88 (4,57)	49,63 (3,91)	43,52 (3,41)	49,63 (3,92)	45,75 (3,69)
	Soma	60,56	60,09	61,97	58,68	60,56
	Produto	60,56	60,56	61,50	60,09	60,56
	Empilhamento	<b>62,44</b>	<b>62,44</b>	49,29	58,21	61,97
SLA	Melhor Único	63,51 (4,05)	51,34 (5,28)	42,33 (3,68)	53,32 (5,12)	45,36 (4,33)
	Soma	<b>63,84</b>	60,56	<b>63,84</b>	60,09	<b>63,84</b>
	Produto	<b>63,84</b>	59,62	62,91	60,09	<b>63,84</b>
	Empilhamento	62,91	61,03	49,76	61,97	<b>63,84</b>

Tabela 56 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,35 (3,28)	47,23 (3,30)	45,35 (3,68)	47,27 (3,47)	44,89 (3,69)
	Soma	<b>61,50</b>	59,15	60,56	58,21	61,03
	Produto	59,15	58,68	60,09	57,74	60,56
	Empilhamento	<b>61,50</b>	56,33	48,82	58,21	57,74
L	Melhor Único	58,20 (3,10)	47,95 (3,99)	43,65 (4,07)	48,49 (3,92)	44,19 (3,85)
	Soma	59,15	58,21	<b>62,44</b>	60,56	59,62
	Produto	61,03	60,09	61,97	61,03	60,09
	Empilhamento	60,06	58,21	46,47	61,03	61,03
SA	Melhor Único	61,23 (2,93)	47,56 (3,23)	41,74 (3,35)	48,16 (3,18)	42,53 (3,40)
	Soma	61,97	58,21	60,09	57,27	61,03
	Produto	61,50	58,21	<b>62,44</b>	56,33	60,56
	Empilhamento	61,03	58,68	50,70	56,80	61,50
SL	Melhor Único	60,46 (3,36)	49,23 (4,53)	44,13 (4,05)	50,53 (4,23)	45,99 (4,15)
	Soma	62,44	58,21	58,68	57,74	60,56
	Produto	61,03	57,74	59,15	59,15	60,09
	Empilhamento	<b>62,91</b>	54,46	44,60	61,97	57,74
LA	Melhor Único	61,70 (3,29)	49,14 (3,99)	43,61 (3,68)	49,47 (3,42)	44,72 (4,22)
	Soma	60,56	59,62	61,03	61,97	61,50
	Produto	61,97	<b>62,44</b>	61,03	61,50	61,03
	Empilhamento	<b>62,44</b>	59,62	49,76	61,03	61,03
SLA	Melhor Único	59,40 (3,18)	48,63 (3,25)	47,66 (4,06)	50,13 (3,53)	46,58 (3,75)
	Soma	<b>63,84</b>	60,56	58,68	60,09	57,74
	Produto	63,38	57,74	57,27	59,15	57,27
	Empilhamento	62,91	61,50	49,76	61,50	57,74

Tabela 57 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	58,98 (3,07)	53,10 (4,11)	43,32 (3,32)	53,11 (3,84)	43,32 (3,62)
	Soma	61,50	57,27	58,68	59,15	56,80
	Produto	60,09	53,99	59,15	58,21	56,80
	Empilhamento	61,50	58,21	46,00	58,21	<b>62,91</b>
L	Melhor Único	60,03 (3,32)	52,24 (4,51)	43,55 (3,57)	52,61 (4,55)	44,48 (3,87)
	Soma	59,15	60,09	59,62	58,68	59,62
	Produto	59,15	61,03	60,56	56,80	60,56
	Empilhamento	59,62	58,21	45,53	57,74	<b>62,44</b>
SA	Melhor Único	62,45 (2,78)	49,37 (4,02)	47,63 (3,54)	49,93 (4,31)	47,85 (4,16)
	Soma	61,97	61,03	62,91	60,09	61,50
	Produto	61,50	61,03	63,38	61,03	61,50
	Empilhamento	62,91	60,56	46,94	60,09	<b>63,84</b>
SL	Melhor Único	60,03 (4,01)	46,25 (3,88)	46,06 (4,39)	46,12 (3,96)	42,33 (4,26)
	Soma	61,03	58,68	61,03	58,68	60,09
	Produto	60,09	58,68	60,56	60,56	60,09
	Empilhamento	<b>62,91</b>	56,33	46,47	61,03	62,44
LA	Melhor Único	60,22 (3,68)	49,09 (4,03)	45,36 (4,01)	51,47 (3,65)	45,34 (3,91)
	Soma	62,44	60,56	60,09	57,74	59,62
	Produto	<b>63,84</b>	60,09	61,03	58,68	59,62
	Empilhamento	63,38	57,74	50,23	56,80	60,09
SLA	Melhor Único	60,78 (2,97)	55,40 (4,73)	45,28 (4,18)	54,31 (4,96)	45,42 (4,11)
	Soma	61,97	58,68	60,56	59,62	60,09
	Produto	<b>62,91</b>	57,27	62,44	59,15	60,56
	Empilhamento	61,03	58,21	52,58	60,09	62,44

Tabela 58 – Acurácia considerando o protocolo LOSO e um *pool* com 70 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,91 (3,75)	50,10 (3,89)	44,65 (3,66)	50,06 (3,53)	45,54 (3,72)
	Soma	60,56	60,56	58,21	57,27	58,21
	Produto	60,56	56,80	59,62	59,62	59,15
	Empilhamento	<b>61,03</b>	59,15	44,60	61,97	60,56
L	Melhor Único	61,26 (3,92)	53,18 (4,25)	43,06 (3,74)	54,12 (3,91)	41,75 (3,69)
	Soma	60,56	61,03	60,56	61,50	56,80
	Produto	60,09	61,50	60,09	61,50	57,27
	Empilhamento	61,50	58,21	46,47	60,09	<b>61,97</b>
SA	Melhor Único	61,20 (3,63)	51,95 (4,55)	47,25 (4,88)	50,70 (4,17)	48,22 (5,16)
	Soma	61,97	61,50	62,44	61,03	61,50
	Produto	61,50	61,50	62,44	61,50	61,97
	Empilhamento	61,97	60,09	47,41	60,09	<b>62,44</b>
SL	Melhor Único	61,01 (3,31)	48,53 (3,77)	43,70 (4,08)	48,10 (3,77)	41,53 (3,30)
	Soma	60,56	58,68	57,74	58,68	59,62
	Produto	61,03	59,15	58,21	60,09	59,62
	Empilhamento	59,62	59,62	46,47	<b>61,50</b>	60,09
LA	Melhor Único	61,24 (3,57)	53,41 (4,48)	46,45 (4,16)	54,36 (4,57)	47,54 (4,59)
	Soma	61,97	61,97	62,44	61,03	62,44
	Produto	62,44	57,27	<b>62,91</b>	60,56	62,44
	Empilhamento	60,56	61,97	50,70	60,09	62,44
SLA	Melhor Único	62,17 (3,32)	52,08 (4,15)	48,31 (4,27)	51,15 (4,15)	47,66 (4,40)
	Soma	62,44	60,56	61,97	60,09	59,62
	Produto	61,97	59,62	61,50	59,15	60,56
	Empilhamento	61,50	59,62	47,88	60,09	<b>62,91</b>

Tabela 59 – Acurácia considerando o protocolo LOSO e um *pool* com 100 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos Ensembles		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,60 (3,56)	49,07 (3,61)	43,57 (3,95)	50,02 (3,42)	43,90 (3,94)
	Soma	<b>61,50</b>	57,74	60,09	57,27	59,62
	Produto	<b>61,50</b>	58,68	59,62	56,80	59,15
	Empilhamento	<b>61,50</b>	59,15	43,19	59,15	61,03
L	Melhor Único	60,01 (5,05)	50,57 (4,31)	46,71 (4,12)	50,98 (4,27)	43,32 (3,95)
	Soma	59,15	59,15	60,56	58,68	59,15
	Produto	59,15	58,68	61,03	59,62	60,56
	Stacking	60,09	58,68	48,82	60,56	<b>61,50</b>
SA	Melhor Único	61,60 (3,23)	54,32 (4,54)	43,80 (3,97)	53,86 (4,72)	46,60 (3,80)
	Soma	62,91	61,97	62,91	60,09	<b>63,38</b>
	Produto	62,91	59,15	<b>63,38</b>	60,09	<b>63,38</b>
	Stacking	62,91	60,56	48,35	58,68	62,44
SL	Melhor Único	63,14 (3,71)	51,17 (4,21)	43,81 (3,96)	49,79 (3,89)	43,97 (4,00)
	Soma	60,56	59,62	61,97	57,74	60,09
	Produto	61,50	59,62	62,44	57,74	60,09
	Empilhamento	61,50	54,46	49,29	61,50	<b>64,78</b>
LA	Melhor Único	61,13 (3,50)	50,28 (4,05)	50,21 (4,24)	52,18 (3,97)	48,88 (4,53)
	Soma	61,97	<b>63,84</b>	60,09	62,44	61,50
	Produto	61,50	61,03	59,62	62,44	61,50
	Empilhamento	62,44	59,15	51,17	61,50	62,91
SLA	Melhor Único	61,91 (3,54)	52,65 (4,21)	47,83 (4,47)	54,25 (4,38)	47,60 (4,25)
	Soma	61,97	60,56	61,03	60,09	61,50
	Produto	61,50	58,68	<b>62,91</b>	59,62	61,50
	Empilhamento	61,97	61,50	51,17	62,44	<b>62,91</b>

Tabela 60 – Acurácia considerando o protocolo LOSO e um *pool* com 150 representações geradas com o método proposto. LFW e JAFFE são usados como conjuntos de dados auxiliares e alvo, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões da camada latente e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos Ensembles		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	60,01 (3,27)	53,31 (4,29)	45,04 (4,09)	54,26 (4,46)	44,49 (4,24)
	Soma	59,62	60,09	59,15	59,15	56,80
	Produto	58,21	57,27	59,62	60,09	57,27
	Empilhamento	60,09	60,09	46,47	61,03	<b>61,50</b>
L	Melhor Único	59,47 (4,49)	49,80 (4,18)	48,02 (4,30)	50,28 (4,20)	47,20 (4,39)
	Soma	59,15	60,09	60,09	59,15	58,68
	Produto	59,15	55,39	61,50	59,62	57,74
	Empilhamento	59,62	58,68	48,35	61,50	<b>62,91</b>
SA	Melhor Único	61,34 (3,26)	54,76 (4,24)	45,88 (3,68)	55,91 (4,33)	45,88 (3,97)
	Soma	62,44	61,50	63,38	62,44	60,56
	Produto	60,56	60,56	<b>63,84</b>	63,38	60,56
	Empilhamento	61,97	61,97	50,23	61,50	63,38
SL	Melhor Único	61,12 (3,46)	50,20 (3,85)	45,80 (4,20)	51,63 (3,76)	46,91 (4,15)
	Soma	60,09	60,56	61,50	60,09	59,62
	Produto	60,09	56,80	61,03	59,15	60,09
	Empilhamento	61,50	58,21	46,94	63,38	<b>63,84</b>
LA	Melhor Único	61,30 (3,47)	56,37 (3,99)	46,07 (3,45)	56,78 (4,20)	48,06 (4,04)
	Soma	61,97	61,03	<b>62,91</b>	60,56	61,97
	Produto	61,03	57,74	<b>62,91</b>	60,56	61,97
	Empilhamento	61,97	60,09	50,70	59,15	60,56
SLA	Melhor Único	62,48 (3,90)	53,42 (4,65)	46,60 (4,50)	52,60 (4,60)	46,81 (4,50)
	Soma	62,44	60,09	61,03	61,03	61,50
	Produto	62,91	60,09	60,56	61,03	60,56
	Empilhamento	<b>63,84</b>	60,09	48,35	60,09	62,91

Tabela 61 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	85,83 (1,35)	65,87 (1,16)	62,66 (2,30)	66,89 (1,66)	63,62 (2,19)
	Soma	87,76	71,86	69,72	71,55	70,03
	Produto	88,07	73,70	70,03	73,08	70,33
	Empilhamento	<b>89,60</b>	73,08	73,39	71,55	73,39
A	Melhor Único	87,09 (0,83)	66,93 (1,05)	64,16 (2,45)	68,27 (1,36)	64,40 (1,73)
	Soma	88,99	74,	68,50	72,17	70,94
	Produto	89,29	75,22	69,11	73,70	70,64
	Empilhamento	<b>89,60</b>	76,45	70,94	71,55	75,84
L	Melhor Único	89,88 (1,79)	71,66 (4,06)	64,59 (2,35)	73,12 (4,05)	65,28 (2,17)
	Soma	90,21	77,06	68,80	76,75	69,72
	Produto	90,51	78,89	68,80	77,37	69,72
	Empilhamento	<b>90,82</b>	79,20	73,39	71,25	72,78
SA	Melhor Único	88,79 (1,65)	72,35 (3,06)	65,02 (1,46)	71,15 (2,08)	64,56 (1,04)
	Soma	88,99	72,47	70,64	71,55	71,25
	Produto	88,99	73,39	70,64	71,55	71,25
	Empilhamento	<b>89,90</b>	74,31	74,00	68,19	73,39
SL	Melhor Único	88,47 (0,72)	71,18 (1,50)	64,15 (1,24)	72,69 (2,03)	65,11 (1,16)
	Soma	88,68	76,75	68,19	77,37	75,84
	Produto	88,99	77,06	68,80	77,67	68,80
	Empilhamento	<b>90,21</b>	77,98	68,80	71,86	75,84
LA	Melhor Único	90,55 (3,40)	70,74 (1,72)	64,32 (0,72)	71,52 (2,10)	66,51 (1,05)
	Soma	88,99	72,78	69,72	74,00	70,64
	Produto	88,68	73,70	70,33	73,70	70,64
	Empilhamento	<b>91,13</b>	76,75	74,31	69,41	72,78
SLA	Melhor Único	88,72 (1,04)	73,54 (4,48)	64,61 (1,78)	71,63 (3,81)	67,10 (2,01)
	Soma	<b>88,99</b>	73,39	67,88	72,47	70,03
	Produto	88,68	74,00	67,88	73,39	70,33
	Empilhamento	<b>88,99</b>	74,31	73,08	69,41	74,61

Tabela 62 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	87,31 (1,63)	68,06 (2,21)	64,99 (2,42)	67,57 (1,57)	65,84 (1,79)
	Soma	87,46	73,70	70,64	73,08	70,94
	Produto	87,15	74,31	70,33	74,00	70,64
	Empilhamento	<b>88,99</b>	78,89	75,22	72,17	75,84
L	Melhor Único	90,09 (4,20)	73,89 (4,14)	69,22 (3,15)	73,88 (3,58)	68,20 (2,96)
	Soma	88,37	76,14	70,64	75,84	70,03
	Produto	88,68	75,22	70,94	75,84	70,33
	Empilhamento	<b>90,21</b>	80,12	75,53	72,78	76,45
SA	Melhor Único	88,75 (1,75)	68,00 (2,03)	66,79 (1,75)	68,99 (2,13)	67,00 (1,90)
	Soma	88,07	75,53	69,41	74,31	70,64
	Produto	88,07	75,84	69,11	75,53	70,33
	Empilhamento	<b>88,99</b>	78,89	73,70	72,47	76,14
SL	Melhor Único	90,35 (3,95)	70,14 (3,14)	68,84 (2,63)	71,12 (3,24)	68,53 (2,89)
	Soma	88,68	71,86	71,55	72,17	70,33
	Produto	88,37	72,47	71,86	72,17	70,33
	Empilhamento	<b>90,82</b>	77,37	76,75	71,55	75,84
LA	Melhor Único	89,15 (1,46)	70,38 (1,85)	65,66 (1,74)	70,64 (2,37)	66,62 (1,65)
	Soma	89,60	76,45	70,33	77,06	70,94
	Produto	<b>89,90</b>	77,67	70,03	76,75	71,25
	Empilhamento	89,60	77,98	75,22	70,94	76,45
SLA	Melhor Único	88,92 (1,59)	70,35 (2,98)	67,86 (2,53)	71,44 (3,13)	67,31 (2,14)
	Soma	89,60	74,31	70,64	74,61	70,94
	Produto	89,29	75,22	70,94	74,92	71,25
	Empilhamento	<b>90,21</b>	78,28	73,08	71,55	75,22

Tabela 63 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,40 (2,74)	71,66 (2,89)	64,51 (1,71)	70,81 (2,91)	66,94 (1,73)
	Soma	87,76	74,00	69,72	74,00	69,72
	Produto	88,07	74,31	70,03	73,39	70,33
	Empilhamento	<b>90,21</b>	77,98	74,61	72,17	74,00
L	Melhor Único	91,12 (3,05)	70,33 (3,05)	66,63 (2,45)	72,35 (3,15)	67,92 (1,94)
	Soma	90,51	75,22	71,25	75,84	72,17
	Produto	90,82	76,74	70,94	76,14	72,47
	Empilhamento	<b>91,43</b>	78,89	76,45	74,31	74,00
SA	Melhor Único	88,68 (2,45)	70,14 (2,02)	67,74 (2,00)	70,42 (1,70)	68,41 (1,86)
	Soma	88,68	73,70	71,25	72,78	71,25
	Produto	88,68	76,75	71,25	73,70	71,25
	Empilhamento	<b>90,21</b>	80,12	73,39	71,25	74,00
SL	Melhor Único	91,01 (3,48)	74,43 (3,53)	65,91 (1,93)	74,57 (3,45)	67,17 (1,79)
	Soma	89,60	75,53	70,03	74,92	69,41
	Produto	89,90	76,14	70,33	75,22	70,03
	Empilhamento	<b>91,13</b>	78,59	75,84	73,39	73,39
LA	Melhor Único	89,03 (1,35)	70,94 (2,26)	67,34 (1,78)	71,62 (2,15)	67,58 (1,80)
	Soma	<b>89,60</b>	77,67	71,86	75,84	70,94
	Produto	<b>89,60</b>	77,98	71,86	77,37	71,55
	Empilhamento	<b>89,60</b>	81,03	71,61	72,78	76,14
SLA	Melhor Único	90,13 (2,67)	71,00 (3,25)	65,91 (2,96)	71,99 (3,50)	68,29 (2,86)
	Soma	90,21	75,84	70,94	75,53	71,25
	Produto	90,21	77,06	70,64	76,14	71,86
	Empilhamento	<b>90,51</b>	78,28	72,47	73,08	73,08

Tabela 64 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,37 (2,67)	72,72 (2,61)	65,67 (1,93)	71,21 (2,67)	66,08 (1,93)
	Soma	88,07	74,61	70,64	74,00	70,64
	Produto	88,37	76,14	71,25	74,61	70,94
	Empilhamento	<b>90,51</b>	78,89	74,92	74,61	73,70
L	Melhor Único	89,90 (3,51)	69,15 (2,49)	66,58 (2,29)	70,25 (2,76)	66,93 (2,62)
	Soma	89,90	74,92	70,64	74,92	69,72
	Produto	89,90	75,22	70,03	75,53	69,72
	Empilhamento	<b>91,74</b>	79,20	75,53	74,00	74,92
SA	Melhor Único	88,37 (1,36)	69,36 (2,17)	66,45 (2,23)	69,75 (2,22)	68,08 (2,17)
	Soma	88,07	74,92	70,94	75,53	70,33
	Produto	88,37	77,37	71,25	75,22	70,33
	Empilhamento	<b>89,60</b>	78,89	72,78	71,55	73,39
SL	Melhor Único	90,35 (3,07)	70,28 (2,31)	66,42 (2,05)	70,80 (2,34)	68,05 (2,44)
	Soma	88,99	73,39	69,11	72,17	70,33
	Produto	89,29	73,70	70,33	73,39	70,64
	Empilhamento	<b>90,82</b>	77,06	73,70	73,70	74,92
LA	Melhor Único	89,60 (3,75)	71,58 (2,48)	67,54 (2,98)	72,16 (2,82)	69,12 (2,93)
	Soma	89,90	76,75	70,94	76,75	71,86
	Produto	89,90	77,37	71,55	77,06	71,86
	Empilhamento	<b>90,21</b>	78,59	74,00	74,86	76,14
SLA	Melhor Único	88,89 (1,58)	73,06 (2,91)	66,97 (2,44)	73,51 (3,04)	68,51 (2,52)
	Soma	89,29	76,75	70,94	77,06	73,08
	Produto	89,90	78,28	71,25	76,75	72,78
	Empilhamento	<b>91,13</b>	80,03	75,22	75,22	75,53

Tabela 65 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	89,13 (2,54)	71,27 (2,47)	66,29 (2,18)	70,86 (2,44)	68,57 (2,29)
	Soma	88,68	75,22	70,03	75,84	70,03
	Produto	88,68	75,53	69,41	76,45	70,33
	Empilhamento	<b>90,82</b>	80,42	74,61	76,75	74,92
L	Melhor Único	92,28 (2,37)	73,23 (3,14)	69,11 (2,65)	74,50 (3,11)	68,14 (2,26)
	Soma	90,21	75,53	69,41	75,84	70,33
	Produto	90,21	75,84	70,03	76,45	70,33
	Empilhamento	<b>92,35</b>	79,20	74,61	73,39	73,39
SA	Melhor Único	88,97 (1,93)	70,05 (2,03)	67,35 (1,90)	71,11 (1,96)	66,87 (1,68)
	Soma	89,29	74,92	69,72	75,53	70,33
	Produto	89,29	75,84	70,03	75,84	70,64
	Empilhamento	<b>89,90</b>	79,51	72,78	71,31	74,61
SL	Melhor Único	89,54 (2,40)	72,41 (2,46)	66,38 (1,67)	73,68 (2,49)	68,74 (1,88)
	Soma	90,51	75,84	70,03	75,53	70,33
	Produto	90,51	77,06	70,33	76,14	70,33
	Empilhamento	<b>91,13</b>	81,34	75,22	75,22	74,31
LA	Melhor Único	89,88 (2,48)	74,16 (2,64)	68,48 (2,29)	72,92 (2,43)	71,23 (2,17)
	Soma	89,90	78,89	72,17	79,51	71,55
	Produto	<b>90,21</b>	78,89	72,47	79,20	71,55
	Empilhamento	89,60	79,81	72,78	74,61	77,67
SLA	Melhor Único	89,15 (1,99)	74,23 (2,81)	67,16 (2,39)	74,26 (2,85)	68,53 (2,56)
	Soma	<b>89,60</b>	75,84	70,64	75,84	71,86
	Produto	89,29	76,45	70,64	76,14	71,86
	Empilhamento	88,99	80,12	74,31	73,70	76,45

Tabela 66 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,44 (2,85)	69,50 (2,23)	66,76 (1,88)	69,71 (2,33)	67,16 (2,05)
	Soma	87,46	74,92	70,03	73,70	70,33
	Produto	87,46	76,75	70,33	74,31	70,33
	Empilhamento	<b>89,29</b>	81,34	74,31	75,53	74,31
L	Melhor Único	90,35 (4,50)	72,25 (2,85)	66,92 (2,33)	71,72 (2,94)	68,30 (2,35)
	Soma	89,60	75,84	70,64	75,53	70,64
	Produto	90,21	76,45	70,64	75,53	70,94
	Empilhamento	<b>90,82</b>	79,81	74,00	74,00	74,92
SA	Melhor Único	90,02 (2,69)	68,55 (1,86)	66,65 (1,77)	70,22 (1,98)	69,08 (1,99)
	Soma	89,29	75,84	70,03	74,92	70,64
	Produto	89,60	78,59	70,03	75,22	70,64
	Empilhamento	<b>90,21</b>	80,42	72,47	73,08	74,00
SL	Melhor Único	89,74 (2,04)	70,59 (2,29)	67,30 (2,17)	72,00 (2,53)	68,50 (1,99)
	Soma	89,60	73,70	69,41	73,39	70,33
	Produto	89,29	74,92	69,41	74,00	70,94
	Empilhamento	<b>90,82</b>	78,89	77,06	75,22	74,31
LA	Melhor Único	90,11 (2,59)	73,53 (2,97)	67,51 (2,38)	72,92 (2,86)	70,07 (2,55)
	Soma	<b>90,21</b>	77,06	71,86	75,84	72,17
	Produto	<b>90,21</b>	77,37	72,47	76,75	72,47
	Empilhamento	<b>90,21</b>	80,12	74,61	75,84	74,92
SLA	Melhor Único	89,78 (2,78)	73,40 (2,97)	67,42 (2,08)	73,68 (3,00)	68,48 (2,14)
	Soma	89,60	75,84	71,55	75,53	72,17
	Produto	89,29	76,45	71,55	75,84	72,17
	Empilhamento	<b>89,90</b>	78,59	73,39	75,84	76,45

Tabela 67 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> of SVMs	Métodos <i>Ensembles</i>		Dinamic Selection (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,48 (2,63)	71,75 (3,09)	65,09 (2,01)	71,90 (2,79)	66,62 (1,90)
	Soma	87,15	74,00	68,19	73,39	68,80
	Produto	87,46	74,92	69,11	74,31	68,80
	Empilhamento	<b>89,90</b>	80,42	76,75	76,45	80,12
L	Melhor Único	89,87 (2,58)	72,45 (2,54)	69,78 (2,34)	71,10 (2,26)	69,13 (2,30)
	Soma	89,60	75,84	71,55	75,22	72,17
	Produto	89,29	76,45	71,55	76,14	72,17
	Empilhamento	<b>90,82</b>	77,98	78,89	74,31	81,34
SA	Melhor Único	89,47 (1,80)	71,55 (2,47)	66,66 (1,90)	73,88 (2,60)	67,74 (1,81)
	Soma	89,90	74,92	70,64	75,22	72,17
	Produto	89,29	77,37	70,64	75,84	72,17
	Empilhamento	<b>90,21</b>	80,12	75,84	75,84	78,89
SL	Melhor Único	91,29 (2,86)	72,57 (2,72)	68,31 (2,14)	72,14 (2,82)	67,37 (1,85)
	Soma	89,60	74,92	70,64	74,00	71,55
	Produto	89,60	75,84	70,64	74,00	71,55
	Empilhamento	<b>92,66</b>	77,67	77,06	74,31	78,89
LA	Melhor Único	87,18 (2,24)	72,06 (2,72)	68,88 (2,02)	73,33 (2,85)	69,51 (2,24)
	Soma	86,85	77,37	71,55	77,06	72,47
	Produto	87,46	77,37	72,17	77,06	72,47
	Empilhamento	<b>88,07</b>	79,81	76,14	78,28	79,81
SLA	Melhor Único	87,00 (3,05)	74,23 (3,16)	68,14 (2,20)	73,29 (3,04)	68,58 (2,28)
	Soma	87,76	76,45	71,86	75,53	71,86
	Produto	87,76	77,06	71,86	76,14	72,17
	Empilhamento	<b>88,07</b>	79,51	76,45	76,45	79,81

Tabela 68 – Acurácia considerando o protocolo LOSO e um *pool* com 70 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,79 (2,15)	67,13 (2,19)	66,05 (1,79)	68,02 (2,16)	67,10 (1,94)
	Soma	87,76	69,72	69,11	70,03	69,11
	Produto	87,76	69,72	69,11	70,03	69,11
	Empilhamento	<b>89,60</b>	73,39	74,61	73,70	74,61
L	Melhor Único	91,03 (2,51)	66,72 (2,15)	67,55 (2,61)	68,34 (2,48)	68,94 (2,61)
	Soma	89,90	70,03	70,94	70,64	71,25
	Produto	89,90	70,94	71,55	70,64	71,55
	Empilhamento	<b>91,13</b>	74,00	75,84	76,14	75,53
SA	Melhor Único	88,96 (2,49)	67,38 (1,91)	66,86 (2,18)	68,37 (1,79)	68,78 (2,26)
	Soma	88,68	71,86	71,55	71,86	71,86
	Produto	88,68	71,86	71,86	72,17	71,86
	Empilhamento	<b>89,90</b>	73,70	74,31	76,14	75,84
SL	Melhor Único	90,38 (3,29)	68,62 (2,29)	68,60 (2,15)	68,60	68,81 (2,29)
	Soma	90,21	71,59	70,64	71,25	70,33
	Produto	90,21	71,59	71,55	71,25	70,64
	Empilhamento	<b>90,51</b>	75,22	74,31	75,22	75,53
LA	Melhor Único	89,71 (2,29)	70,25 (2,15)	67,21 (2,29)	70,35 (2,27)	68,61 (2,09)
	Soma	<b>89,90</b>	71,86	71,25	72,17	71,55
	Produto	<b>89,90</b>	72,17	71,86	72,17	72,17
	Empilhamento	<b>89,90</b>	75,53	73,08	76,14	73,70
SLA	Melhor Único	90,19 (2,39)	69,16 (2,11)	67,76 (2,36)	68,88	68,20 (2,34)
	Soma	89,90	71,25	71,55	70,64	71,86
	Produto	<b>90,21</b>	71,55	71,55	70,94	72,47
	Empilhamento	89,60	74,92	73,39	73,70	75,22

Tabela 69 – Acurácia considerando o protocolo LOSO e um *pool* com 100 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	89,43 (2,47)	71,48 (2,51)	66,93 (2,08)	70,93 (2,51)	69,83 (2,19)
	Soma	89,37	73,08	70,03	73,39	70,64
	Produto	88,37	74,31	70,33	73,39	70,64
	Empilhamento	<b>90,21</b>	78,28	75,22	75,53	75,53
L	Melhor Único	90,77 (3,32)	72,37 (2,34)	68,74 (2,15)	72,00 (2,31)	68,87 (2,10)
	Soma	89,60	74,61	73,03	74,31	70,64
	Produto	89,60	74,92	70,64	74,31	70,94
	Empilhamento	<b>90,82</b>	79,51	77,67	75,53	81,65
SA	Melhor Único	88,92 (1,93)	71,85 (2,63)	68,02 (1,97)	72,90 (2,59)	69,50 (1,90)
	Soma	88,68	76,14	70,33	75,53	70,64
	Produto	88,68	77,06	70,94	76,45	70,64
	Empilhamento	<b>88,99</b>	79,81	73,08	77,67	76,14
SL	Melhor Único	90,39 (2,39)	72,62 (2,95)	68,09 (2,19)	73,68 (2,99)	69,18 (2,45)
	Soma	89,29	75,22	70,33	75,22	70,33
	Produto	88,99	76,45	70,33	76,45	70,94
	Empilhamento	<b>90,51</b>	77,98	73,39	78,59	77,67
LA	Melhor Único	90,39 (2,61)	75,07 (2,99)	69,12 (2,46)	75,63 (2,87)	70,16 (2,37)
	Soma	89,90	76,75	71,55	77,06	72,47
	Produto	<b>90,51</b>	77,67	71,86	77,37	72,47
	Empilhamento	89,90	81,65	74,61	77,06	76,14
SLA	Melhor Único	89,78 (2,82)	73,06 (2,52)	69,26 (2,26)	72,96 (2,56)	69,19 (2,17)
	Soma	<b>89,90</b>	78,28	71,55	76,75	72,47
	Produto	<b>89,90</b>	78,28	71,86	77,98	72,78
	Empilhamento	89,60	81,34	74,31	75,84	74,92

Tabela 70 – Acurácia considerando o protocolo LOSO e um *pool* com 150 representações geradas com o método proposto. Kyoto e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em um conjunto de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	90,00 (2,32)	72,75 (2,64)	66,73 (2,13)	73,29 (2,65)	69,75 (2,18)
	Soma	88,07	74,92	69,11	74,61	70,03
	Produto	88,37	75,53	69,11	75,22	70,03
	Empilhamento	<b>90,21</b>	78,59	75,84	75,53	79,51
L	Melhor Único	90,80 (3,08)	72,52 (2,44)	67,88 (2,21)	73,96 (2,47)	67,64 (2,02)
	Soma	89,60	75,22	70,94	74,92	70,64
	Produto	89,90	76,14	70,94	75,22	70,64
	Empilhamento	<b>90,82</b>	78,59	76,14	77,37	81,03
SA	Melhor Único	89,85 (2,98)	72,28 (2,37)	67,55 (2,14)	72,54 (2,39)	68,67 (2,10)
	Soma	89,29	75,22	71,25	74,92	71,55
	Produto	88,99	76,75	71,55	75,84	71,86
	Empilhamento	<b>90,21</b>	79,20	75,22	76,14	80,73
SL	Melhor Único	90,80 (2,55)	73,06 (2,74)	68,10 (2,27)	74,33 (2,70)	70,15 (2,28)
	Soma	90,21	74,92	70,64	75,53	70,64
	Produto	89,90	75,53	70,94	75,53	70,94
	Empilhamento	<b>90,82</b>	77,98	75,84	75,53	81,65
LA	Melhor Único	90,04 (2,12)	74,74 (2,84)	67,98 (2,30)	74,47 (2,75)	69,97 (2,31)
	Soma	<b>90,21</b>	76,14	71,86	76,14	72,78
	Produto	<b>90,21</b>	77,98	71,86	76,75	72,78
	Empilhamento	88,99	81,34	77,06	76,75	80,42
SLA	Melhor Único	89,94 (2,60)	74,22 (2,93)	68,55 (2,21)	75,39 (2,94)	69,26 (2,27)
	Soma	89,90	76,14	71,86	75,84	72,17
	Produto	<b>90,21</b>	76,14	71,86	76,14	72,47
	Empilhamento	89,29	80,73	75,84	76,45	80,12

Tabela 71 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	86,35 (1,54)	68,78 (1,95)	63,61 (1,32)	69,26 (1,63)	65,49 (1,00)
	Soma	86,54	72,78	64,91	73,08	69,41
	Produto	86,85	73,08	70,03	73,70	70,03
	Empilhamento	<b>89,29</b>	78,28	75,53	71,25	74,31
A	Melhor Único	87,99 (1,38)	68,84 (1,76)	64,46 (1,18)	69,19 (1,89)	65,48 (1,05)
	Soma	88,37	74,61	69,11	75,22	70,33
	Produto	<b>88,68</b>	75,22	69,72	75,53	70,33
	Empilhamento	<b>88,68</b>	78,89	74,31	70,94	74,61
L	Melhor Único	88,47 (1,99)	70,12 (1,38)	67,99 (2,49)	70,97 (1,65)	67,52 (1,80)
	Soma	88,68	75,84	69,72	74,92	70,03
	Produto	88,37	74,92	70,03	75,22	69,72
	Empilhamento	<b>89,90</b>	78,28	74,31	72,78	74,92
SA	Melhor Único	85,83 (1,67)	67,95 (2,23)	65,07 (1,70)	68,94 (2,31)	65,42 (1,42)
	Soma	87,76	72,47	68,19	72,17	68,80
	Produto	88,07	72,47	69,11	71,55	69,11
	Empilhamento	<b>88,99</b>	76,75	74,92	68,80	72,78
SL	Melhor Único	90,16 (0,95)	66,05 (1,26)	64,88 (1,17)	68,70 (1,95)	66,04 (1,93)
	Soma	89,60	71,55	70,94	71,86	70,94
	Produto	89,60	71,86	70,94	71,86	71,86
	Empilhamento	<b>90,21</b>	75,84	76,14	72,17	74,31
LA	Melhor Único	88,01 (2,41)	70,84 (0,97)	66,38 (1,32)	71,08 (0,97)	67,72 (1,51)
	Soma	88,07	75,53	70,33	75,84	70,64
	Produto	87,46	77,06	70,33	75,53	70,33
	Empilhamento	<b>88,68</b>	78,28	75,22	70,03	72,78
SLA	Melhor Único	87,22 (3,37)	69,37 (1,28)	67,38 (2,61)	70,46 (1,69)	68,55 (3,16)
	Soma	87,15	74,31	69,11	72,78	70,33
	Produto	87,15	74,61	69,11	74,00	70,03
	Empilhamento	<b>87,46</b>	74,92	74,31	73,08	73,70

Tabela 72 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	87,31 (2,27)	70,36 (2,22)	65,76 (1,75)	70,81 (1,88)	65,64 (1,55)
	Soma	87,15	73,70	70,03	72,17	69,11
	Produto	88,07	74,61	70,33	73,39	69,11
	Empilhamento	<b>88,99</b>	77,67	76,45	74,92	76,75
L	Melhor Único	89,90 (2,68)	73,31 (2,87)	66,93 (2,29)	73,10 (2,79)	68,88 (2,90)
	Soma	90,82	73,39	68,80	72,78	68,80
	Produto	90,51	75,22	68,50	73,08	68,80
	Empilhamento	<b>90,82</b>	78,89	76,14	74,00	77,06
SA	Melhor Único	88,89 (1,38)	70,57 (2,26)	65,43 (1,42)	70,38 (2,31)	67,44 (1,92)
	Soma	88,37	74,61	70,94	74,92	70,94
	Produto	88,37	75,22	71,25	75,22	71,25
	Empilhamento	<b>89,60</b>	78,59	73,08	74,61	75,22
SL	Melhor Único	89,78 (3,20)	71,05 (2,53)	66,41 (1,89)	70,91 (2,34)	67,01 (1,96)
	Soma	88,99	74,31	70,03	74,31	70,03
	Produto	88,68	75,22	70,33	74,61	70,03
	Empilhamento	<b>90,21</b>	79,51	77,37	76,14	77,37
LA	Melhor Único	89,60 (1,78)	70,87 (2,70)	66,93 (1,88)	72,40 (2,98)	69,12 (2,46)
	Soma	<b>89,90</b>	75,53	71,55	77,06	70,94
	Produto	89,60	76,14	71,25	76,14	70,94
	Empilhamento	89,60	78,28	74,61	74,61	76,75
SLA	Melhor Único	91,01 (3,74)	71,21 (2,24)	67,03 (1,65)	72,47 (2,97)	67,66 (1,87)
	Soma	90,82	74,92	69,72	73,39	70,64
	Produto	<b>91,43</b>	76,75	69,41	74,31	70,33
	Empilhamento	90,21	79,81	76,14	74,31	75,22

Tabela 73 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	87,38 (1,79)	70,39 (2,34)	67,30 (2,43)	71,55 (2,49)	68,16 (2,23)
	Soma	87,46	74,61	69,11	74,61	69,41
	Produto	87,15	74,92	69,72	74,92	69,41
	Empilhamento	<b>90,21</b>	79,81	77,37	75,35	77,67
L	Melhor Único	89,53 (3,22)	72,10 (2,62)	64,84 (1,67)	72,50 (2,93)	66,32 (1,58)
	Soma	89,26	73,08	69,72	72,47	71,25
	Produto	89,29	73,70	70,03	73,70	70,64
	Empilhamento	<b>89,90</b>	77,98	76,14	75,53	78,89
SA	Melhor Único	89,20 (2,28)	71,69 (2,35)	66,77 (1,58)	73,10 (2,26)	68,26 (1,89)
	Soma	88,68	74,61	71,25	74,92	70,64
	Produto	89,29	76,75	71,55	75,84	70,94
	Empilhamento	<b>89,90</b>	80,42	75,53	76,75	76,75
SL	Melhor Único	89,81 (1,22)	72,16 (2,13)	66,37 (1,60)	71,10 (1,89)	67,71 (1,93)
	Soma	90,82	74,61	70,03	74,00	71,25
	Produto	90,82	75,53	70,03	74,61	71,55
	Empilhamento	<b>91,13</b>	78,59	77,98	75,22	77,67
LA	Melhor Único	90,08 (1,96)	73,55 (3,00)	67,40 (1,97)	73,62 (2,73)	66,11 (1,42)
	Soma	90,21	75,53	71,25	75,84	71,86
	Produto	89,90	75,84	71,25	75,84	71,86
	Empilhamento	<b>90,51</b>	78,89	76,45	76,14	78,59
SLA	Melhor Único	89,57 (2,09)	71,79 (2,57)	67,09 (2,30)	73,37 (2,66)	69,81 (2,28)
	Soma	88,99	75,22	71,55	74,92	72,47
	Produto	88,68	76,45	72,17	74,31	72,47
	Empilhamento	<b>89,60</b>	77,98	74,31	76,14	77,06

Tabela 74 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	87,34 (3,50)	71,75 (2,68)	65,21 (1,78)	71,22 (2,51)	66,66 (1,65)
	Soma	86,54	72,78	69,72	73,39	70,33
	Produto	86,85	74,00	70,03	74,00	70,64
	Empilhamento	<b>89,90</b>	81,95	75,53	76,75	76,75
L	Melhor Único	90,02 (1,74)	73,47 (2,20)	66,58 (1,58)	74,53 (2,47)	68,41 (1,97)
	Soma	90,51	76,45	70,94	76,14	71,25
	Produto	90,82	76,45	71,25	76,45	71,55
	Empilhamento	<b>91,43</b>	81,34	78,59	76,75	77,98
SA	Melhor Único	89,18 (2,26)	72,98 (3,15)	68,16 (2,05)	73,23 (2,97)	69,12 (1,91)
	Soma	89,29	76,14	69,11	75,53	70,03
	Produto	89,60	77,06	69,72	76,75	70,33
	Empilhamento	<b>89,90</b>	78,59	74,61	77,06	74,92
SL	Melhor Único	89,22 (2,38)	72,35 (2,77)	66,12 (1,95)	72,47 (3,07)	67,81 (2,22)
	Soma	89,29	74,61	70,03	75,53	70,64
	Produto	89,29	75,84	69,72	74,92	71,25
	Empilhamento	<b>90,51</b>	77,98	76,14	75,53	79,20
LA	Melhor Único	89,23 (2,49)	71,10 (2,45)	66,96 (1,70)	71,38 (1,91)	68,61 (1,94)
	Soma	<b>89,90</b>	78,89	70,94	77,37	72,17
	Produto	89,60	79,81	70,94	78,59	72,17
	Empilhamento	89,29	79,20	74,61	77,06	77,98
SLA	Melhor Único	90,06 (2,08)	67,65 (2,31)	67,65 (2,31)	68,50 (2,45)	68,50 (2,45)
	Soma	90,21	71,25	71,25	72,17	72,17
	Produto	<b>90,51</b>	71,25	71,25	72,17	72,17
	Empilhamento	90,21	75,84	75,84	77,98	77,98

Tabela 75 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> of SVMs	Métodos de <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Best Single	89,57 (4,67)	69,50 (2,23)	66,93 (2,05)	69,97 (2,22)	66,72 (2,23)
	Soma	87,46	73,70	70,03	73,08	70,03
	Produto	87,76	74,00	70,03	73,70	70,03
	Empilhamento	<b>89,90</b>	78,89	77,37	75,22	79,20
L	Best Single	90,38 (2,25)	72,16 (2,27)	65,43 (1,47)	71,99 (1,99)	66,76 (1,78)
	Soma	89,90	76,45	69,74	75,53	70,64
	Produto	89,90	75,84	69,72	76,45	70,64
	Empilhamento	<b>91,74</b>	81,34	78,59	74,31	77,98
SA	Best Single	88,87 (1,91)	69,19 (2,13)	67,09 (2,11)	70,55 (2,16)	68,37 (2,09)
	Soma	88,68	74,92	70,33	75,53	70,33
	Produto	88,68	76,75	70,64	75,53	70,33
	Empilhamento	<b>89,29</b>	79,81	75,22	77,98	77,37
SL	Best Single	89,71 (2,60)	71,08 (2,45)	69,03 (2,16)	71,51 (2,17)	67,31 (1,63)
	Soma	88,68	74,61	70,03	74,31	70,33
	Produto	88,99	74,61	70,33	74,92	70,03
	Empilhamento	<b>90,51</b>	79,81	75,22	75,53	77,37
LA	Best Single	89,39 (3,84)	72,37 (2,68)	67,92 (2,53)	71,86 (2,82)	69,49 (3,01)
	Soma	90,21	77,06	70,94	75,84	70,94
	Produto	89,90	77,67	71,25	76,75	70,94
	Empilhamento	<b>90,51</b>	80,12	75,53	77,98	79,51
SLA	Best Single	90,45 (2,38)	71,17 (3,21)	66,38 (2,36)	73,13 (3,43)	67,93 (1,95)
	Soma	89,29	75,22	70,33	74,92	70,64
	Produto	89,29	74,92	71,25	75,22	70,64
	Empilhamento	<b>90,51</b>	79,81	76,45	76,14	76,75

Tabela 76 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,99 (2,19)	70,29 (2,70)	66,44 (1,91)	70,60 (2,51)	66,69 (1,99)
	Soma	87,76	74,92	70,03	73,70	69,72
	Produto	87,46	75,84	70,33	74,92	70,03
	Empilhamento	<b>89,90</b>	81,65	75,53	75,22	79,51
L	Melhor Único	90,80 (2,43)	70,22 (2,23)	65,17 (1,61)	70,22 (2,20)	67,54 (1,52)
	Soma	90,21	75,22	70,03	74,92	69,72
	Produto	90,21	75,84	70,03	75,22	69,72
	Empilhamento	<b>91,43</b>	77,98	77,98	74,92	78,59
SA	Melhor Único	90,04 (2,36)	71,69 (2,21)	66,24 (1,68)	71,61 (2,13)	68,02 (2,06)
	Soma	89,29	74,00	70,33	74,61	70,94
	Produto	89,60	74,00	70,64	75,22	70,94
	Empilhamento	<b>90,51</b>	80,42	76,45	75,53	78,59
SL	Melhor Único	90,01 (1,94)	71,20 (2,20)	65,76 (1,68)	71,56 (2,39)	67,75 (1,56)
	Soma	88,68	75,53	69,11	74,61	70,03
	Produto	88,68	76,75	69,11	75,84	70,03
	Empilhamento	<b>90,82</b>	78,28	77,67	73,70	78,89
LA	Melhor Único	90,12 (3,36)	73,41 (2,68)	70,59 (2,35)	73,44 (2,45)	68,77 (2,13)
	Soma	<b>90,51</b>	75,84	72,47	75,22	71,86
	Produto	90,21	77,06	72,47	77,37	72,17
	Empilhamento	90,21	79,20	75,22	75,53	78,59
SLA	Melhor Único	90,11 (2,58)	73,50 (2,84)	67,25 (2,24)	74,25 (3,09)	68,70 (2,43)
	Soma	90,21	77,37	71,55	77,06	71,86
	Produto	90,21	78,28	71,55	77,67	71,86
	Empilhamento	90,21	79,81	76,75	77,37	79,81

Tabela 77 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,57 (2,49)	70,29 (2,20)	65,59 (2,13)	70,53 (2,32)	67,00 (2,02)
	Soma	87,15	74,61	70,64	73,39	70,33
	Produto	87,15	75,22	70,94	74,61	70,33
	Empilhamento	<b>89,60</b>	79,81	75,53	74,31	79,51
L	Melhor Único	89,77 (3,48)	72,74 (2,82)	67,03 (2,61)	71,87 (2,45)	70,53 (2,37)
	Soma	89,90	75,53	70,64	75,84	70,64
	Produto	89,60	76,45	70,64	75,53	70,64
	Empilhamento	<b>90,51</b>	80,73	78,28	75,84	81,34
SA	Melhor Único	89,64 (2,12)	72,64 (2,75)	65,69 (1,60)	74,26 (2,75)	66,86 (1,64)
	Soma	89,90	77,06	70,33	76,45	70,64
	Produto	89,60	77,67	70,03	77,06	70,94
	Empilhamento	<b>90,51</b>	81,65	76,45	78,28	79,20
SL	Melhor Único	90,84 ( )	72,11 (2,48)	66,92 (1,99)	72,66 (2,49)	68,87 (2,07)
	Soma	89,90	76,14	70,94	75,84	71,86
	Produto	89,90	77,67	71,25	76,14	72,47
	Empilhamento	<b>91,13</b>	78,89	77,98	75,84	82,26
LA	Melhor Único	89,88 (2,80)	73,14 (2,85)	67,64 (1,78)	74,80 (2,99)	68,65 (1,73)
	Soma	<b>89,90</b>	76,14	71,25	75,22	71,55
	Produto	<b>89,90</b>	77,06	71,25	76,75	71,55
	Empilhamento	89,29	77,67	76,14	74,61	79,20
SLA	Melhor Único	89,42 (3,06)	72,99 (2,82)	67,28 (2,19)	74,15 (2,68)	68,46 (2,32)
	Soma	89,29	77,98	70,03	76,75	70,94
	Produto	89,29	78,28	70,64	77,06	71,25
	Empilhamento	<b>89,60</b>	78,59	75,84	76,14	79,20

Tabela 78 – Acurácia considerando o protocolo LOSO e um *pool* com 70 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool</i> de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	90,01 (2,41)	70,25 (2,59)	67,83 (2,17)	71,03 (2,60)	67,31 (1,83)
	Soma	88,68	74,31	69,41	73,70	70,33
	Produto	88,68	74,92	69,41	74,61	70,33
	Empilhamento	<b>90,21</b>	81,34	74,61	77,06	80,73
L	Melhor Único	90,42 (3,40)	72,16 (2,81)	69,08 (2,43)	72,99 (2,92)	68,61 (2,32)
	Soma	89,29	74,61	70,33	73,70	69,41
	Produto	89,29	76,14	70,64	74,61	69,72
	Empilhamento	<b>91,43</b>	80,73	75,22	74,00	80,73
SA	Melhor Único	88,31 (2,32)	71,72 (2,48)	66,38 (1,92)	72,89 (2,42)	66,97 (1,99)
	Soma	88,07	75,22	70,33	74,00	70,64
	Produto	88,07	76,75	70,33	74,61	70,64
	Empilhamento	<b>88,99</b>	81,03	75,53	75,53	78,89
SL	Melhor Único	90,42 (2,03)	73,24 (2,86)	67,59 (2,04)	73,88 (2,81)	68,62 (1,90)
	Soma	89,90	74,92	70,94	75,84	71,25
	Produto	89,90	74,92	70,94	76,14	71,55
	Empilhamento	<b>90,82</b>	78,59	76,45	75,84	81,03
LA	Melhor Único	88,96 (2,87)	73,88 (2,86)	68,82 (2,23)	74,05 (2,70)	69,08 (2,05)
	Soma	<b>89,90</b>	78,28	71,25	77,37	71,86
	Produto	<b>89,90</b>	80,12	71,55	78,28	71,86
	Empilhamento	88,68	78,89	77,37	75,22	79,81
SLA	Melhor Único	89,88 (2,74)	75,96 (2,81)	68,99 (1,91)	75,88 (2,77)	68,31 (1,76)
	Soma	89,60	76,14	70,94	75,84	71,25
	Produto	89,60	76,75	71,55	76,75	71,86
	Empilhamento	<b>89,90</b>	81,65	76,45	74,92	79,20

Tabela 79 – Acurácia considerando o protocolo LOSO e um *pool* com 100 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	89,54 (2,60)	72,89 (2,47)	65,64 (1,82)	71,86 (2,38)	66,92 (1,94)
	Soma	86,85	74,00	70,03	73,39	70,33
	Produto	87,76	75,84	69,11	74,31	70,33
	Empilhamento	<b>89,90</b>	77,98	74,00	74,31	77,98
L	Melhor Único	90,52 (5,30)	72,66 (2,75)	67,47 (2,17)	73,20 (2,93)	68,23 (2,23)
	Soma	89,60	73,70	70,03	74,31	70,03
	Produto	89,60	76,14	70,33	75,22	70,03
	Empilhamento	<b>90,82</b>	78,89	76,45	74,00	80,42
SA	Melhor Único	89,64 (2,20)	73,58 (2,54)	67,16 (1,83)	73,16 (2,49)	69,19 (1,96)
	Soma	89,60	75,53	71,25	74,31	71,86
	Produto	<b>89,90</b>	75,22	71,25	74,61	71,86
	Empilhamento	89,60	81,03	75,22	76,75	80,73
SL	Melhor Único	90,28 (2,32)	72,85 (2,52)	68,87 (1,90)	72,61 (2,44)	68,87 (2,00)
	Soma	89,60	74,31	70,33	74,00	69,41
	Produto	89,60	74,31	70,33	74,31	69,72
	Empilhamento	<b>90,51</b>	78,28	76,75	75,53	80,42
LA	Melhor Único	90,02 (2,21)	72,38 (2,81)	70,42 (2,04)	73,06 (2,79)	69,40 (2,02)
	Soma	89,29	76,14	70,94	75,22	70,94
	Produto	<b>90,21</b>	76,45	70,94	77,37	71,25
	Empilhamento	89,29	81,03	77,06	74,00	80,12
SLA	Melhor Único	90,45 (2,61)	74,42 (2,75)	68,61 (2,18)	74,63 (2,86)	69,20 (2,26)
	Soma	<b>90,51</b>	76,45	72,17	76,45	72,47
	Produto	90,21	77,98	72,17	77,98	72,47
	Empilhamento	<b>89,90</b>	81,95	76,45	76,14	81,34

Tabela 80 – Acurácia considerando o protocolo LOSO e um *pool* com 150 representações geradas com o método proposto. Os conjuntos de dados LabelMe e CK+ são usados como conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, dimensões vetoriais latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	89,29 (2,52)	71,35 (2,65)	67,03 (1,95)	71,94 (2,53)	67,92 (1,93)
	Soma	87,46	74,61	69,11	74,31	68,80
	Produto	87,46	75,53	69,41	75,22	69,11
	Empilhamento	<b>89,90</b>	80,12	74,00	75,22	77,06
L	Melhor Único	90,16 (2,52)	73,24 (2,64)	68,23 (2,26)	74,11 (2,57)	70,83 (2,31)
	Soma	89,90	74,61	70,03	74,31	70,33
	Produto	89,29	75,84	70,03	74,92	70,33
	Empilhamento	<b>90,21</b>	79,81	74,61	74,92	76,75
SA	Melhor Único	89,53 (2,07)	72,52 (2,45)	69,16 (2,21)	72,11 (2,39)	68,77 (1,97)
	Soma	88,99	76,14	70,94	74,61	70,94
	Produto	89,29	76,75	71,55	76,14	70,94
	Empilhamento	<b>89,60</b>	79,20	74,31	74,92	74,92
SL	Melhor Único	90,16 (3,02)	74,61 (2,75)	68,23 (2,10)	74,83 (2,59)	69,47 (2,29)
	Soma	89,60	74,61	70,64	74,92	70,94
	Produto	89,60	76,14	70,64	75,53	70,94
	Empilhamento	<b>90,21</b>	77,98	74,61	75,53	78,59
LA	Melhor Único	89,53 (2,66)	74,08 (2,85)	68,98 (2,19)	73,77 (2,75)	68,78 (2,09)
	Soma	<b>89,60</b>	77,37	71,55	76,75	71,86
	Produto	<b>89,60</b>	77,67	70,94	77,98	71,86
	Empilhamento	88,99	81,65	74,92	75,53	74,92
SLA	Melhor Único	89,57 (2,10)	74,50 (2,69)	69,33 (1,90)	73,91 (2,66)	69,87 (1,85)
	Soma	<b>89,60</b>	77,67	71,86	76,75	72,17
	Produto	<b>89,60</b>	78,89	71,86	77,98	72,47
	Empilhamento	89,29	81,65	75,53	76,45	76,75

Tabela 81 – Acurácia considerando o protocolo LOSO e um *pool* com 5 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		<i>Pool of SVMs</i>	<i>Ensembles Methods</i>		Dinamic Selection (KU)	
			BG	RF	DT	RF
S	Best Single	83,24 (1,69)	66,75 (1,68)	63,22 (1,28)	66,96 (2,04)	64,77 (1,17)
	Soma	85,93	73,08	68,80	71,86	69,41
	Produto	85,93	73,39	69,11	73,08	69,72
	Empilhamento	<b>87,15</b>	74,92	73,70	72,47	72,47
A	Best Single	84,91 (0,74)	66,82 (1,04)	65,38 (1,42)	67,20 (0,96)	66,37 (1,80)
	Soma	87,46	74,00	67,88	73,70	70,03
	Produto	<b>88,07</b>	74,92	68,19	74,00	70,33
	Empilhamento	87,76	76,45	73,70	73,39	74,31
L	Best Single	84,27 (3,47)	68,62 (3,21)	64,26 (2,50)	69,15 (3,20)	64,50 (2,25)
	Soma	<b>87,76</b>	75,35	68,80	75,53	68,50
	Produto	87,46	77,06	68,19	74,92	68,80
	Empilhamento	87,46	78,28	74,92	76,14	73,39
SA	Best Single	88,16 (3,35)	71,96 (3,05)	66,77 (1,56)	72,45 (3,01)	67,07 (2,36)
	Soma	87,76	72,17	70,64	71,55	71,55
	Produto	87,76	73,08	70,64	71,55	71,55
	Empilhamento	<b>88,99</b>	74,61	75,84	71,55	74,31
SL	Best Single	85,69 (0,99)	68,75 (2,29)	66,96 (2,46)	68,77 (1,88)	68,26 (2,22)
	Soma	90,21	75,53	70,33	75,84	70,64
	Produto	89,60	75,22	70,33	75,53	70,94
	Empilhamento	<b>90,51</b>	78,89	77,67	75,22	77,37
LA	Best Single	83,92 (1,42)	71,56 (2,68)	67,17 (1,30)	69,83 (2,02)	68,24 (1,72)
	Soma	87,15	72,47	70,94	73,08	70,33
	Produto	86,85	73,39	70,33	72,78	70,33
	Empilhamento	<b>87,76</b>	76,75	77,06	71,86	75,22
SLA	Best Single	86,70 (5,41)	71,14 (4,09)	66,97 (2,69)	70,67 (3,90)	65,14 (1,86)
	Soma	86,54	74,92	69,41	73,08	69,41
	Produto	85,62	75,22	69,72	74,61	69,41
	Empilhamento	<b>86,85</b>	74,61	73,70	73,70	74,00

Tabela 82 – Acurácia considerando o protocolo LOSO e um *pool* com 10 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	84,19 (2,28)	69,92 (2,34)	64,91 (1,64)	69,78 (2,15)	66,35 (1,50)
	Soma	86,85	75,22	70,64	73,70	70,64
	Produto	87,15	74,61	70,64	74,31	70,64
	Empilhamento	<b>87,76</b>	78,89	76,14	73,08	76,75
L	Melhor Único	86,96 (2,45)	73,50 (3,29)	66,82 (1,99)	73,13 (3,56)	66,38 (2,09)
	Soma	89,90	75,22	68,80	74,92	69,72
	Produto	89,60	75,22	68,80	74,61	69,11
	Empilhamento	88,99	77,98	74,00	76,45	76,14
SA	Melhor Único	83,71 (1,51)	69,73 (2,84)	64,95 (1,47)	69,29 (2,43)	65,98 (1,88)
	Soma	87,46	76,75	70,03	75,22	70,03
	Produto	<b>88,07</b>	76,14	70,64	76,14	70,03
	Empilhamento	87,46	78,89	75,84	74,92	76,45
SL	Melhor Único	89,12 (2,21)	69,95 (2,50)	66,52 (2,11)	70,90 (2,77)	66,01 (2,08)
	Soma	<b>89,60</b>	71,25	69,11	72,17	69,41
	Produto	<b>89,60</b>	71,86	69,11	71,86	69,72
	Empilhamento	<b>89,60</b>	77,98	75,22	74,31	76,45
LA	Melhor Único	86,70 (1,90)	70,91 (3,02)	66,96 (1,64)	72,28 (2,91)	66,55 (1,56)
	Soma	87,46	76,45	71,55	76,75	71,86
	Produto	87,46	77,67	71,86	76,75	72,17
	Empilhamento	<b>87,76</b>	79,20	75,22	76,14	77,06
SLA	Melhor Único	85,76 (1,98)	71,90 (2,70)	65,60 (1,79)	71,97 (3,07)	66,97 (2,12)
	Soma	<b>86,85</b>	75,84	70,64	74,92	70,03
	Produto	<b>86,85</b>	77,06	69,72	75,84	70,33
	Empilhamento	86,23	79,20	75,53	76,14	75,53

Tabela 83 – Acurácia considerando o protocolo LOSO e um *pool* com 15 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	85,98 (1,76)	68,47 (2,65)	66,59 (2,35)	68,92 (2,83)	67,37 (2,39)
	Soma	88,68	73,39	70,03	73,39	70,33
	Produto	88,37	74,92	70,03	73,08	70,64
	Empilhamento	<b>89,29</b>	77,37	78,28	75,84	79,20
L	Melhor Único	86,25 (1,97)	72,10 (2,98)	68,05 (2,13)	73,23 (3,23)	68,97 (2,20)
	Soma	89,29	75,22	71,86	74,61	70,94
	Produto	89,60	76,14	72,17	75,84	70,94
	Empilhamento	89,90	78,89	77,37	78,59	73,08
SA	Melhor Único	85,73 (2,40)	72,06 (2,19)	65,66 (1,76)	72,23 (1,94)	65,87 (1,34)
	Soma	88,37	73,08	70,03	72,78	69,72
	Produto	88,37	77,98	69,72	74,00	69,72
	Empilhamento	88,99	81,03	74,31	73,70	76,45
SL	Melhor Único	85,79 (2,48)	74,32 (3,07)	68,43 (2,88)	73,47 (2,93)	69,03 (2,17)
	Soma	<b>88,99</b>	75,84	70,03	75,22	70,33
	Produto	<b>88,99</b>	75,84	70,33	76,14	70,33
	Empilhamento	88,07	78,59	78,28	75,53	78,89
LA	Melhor Único	85,67 (1,46)	71,72 (1,64)	66,61 (1,96)	72,07 (1,55)	68,93 (2,21)
	Soma	87,76	77,06	70,94	76,75	71,25
	Produto	<b>88,68</b>	77,67	71,25	76,75	71,25
	Empilhamento	<b>88,68</b>	79,20	74,61	76,75	76,75
SLA	Melhor Único	86,27 (4,47)	71,70 (3,17)	67,35 (3,04)	72,27 (3,20)	68,13 (3,46)
	Soma	87,15	75,84	72,17	75,22	71,55
	Produto	<b>88,07</b>	76,45	71,86	75,22	71,86
	Empilhamento	<b>88,07</b>	79,20	76,14	75,22	77,06

Tabela 84 – Acurácia considerando o protocolo LOSO e um *pool* com 20 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	85,29 (2,04)	72,72 (2,61)	65,14 (2,43)	71,21 (2,67)	67,10 (2,29)
	Soma	88,37	74,61	69,72	74,00	70,64
	Produto	89,29	76,14	71,25	74,61	70,64
	Empilhamento	<b>89,29</b>	78,89	75,22	74,61	74,92
L	Melhor Único	86,25 (1,97)	70,31 (2,91)	67,86 (2,54)	71,53 (2,99)	68,07 (2,91)
	Soma	89,29	75,22	69,11	74,31	70,33
	Produto	89,60	75,84	68,80	75,84	70,33
	Empilhamento	<b>89,90</b>	79,51	74,31	74,00	74,31
SA	Melhor Único	85,02 (1,41)	69,36 (2,17)	66,31 (2,06)	69,75 (2,22)	67,72 (2,19)
	Soma	88,99	74,92	71,25	75,53	71,55
	Produto	88,68	77,37	71,55	75,22	71,25
	Empilhamento	<b>90,21</b>	78,89	72,78	71,55	75,53
SL	Melhor Único	87,86 (2,02)	70,07 (2,44)	65,72 (1,32)	70,80 (2,54)	66,52 (1,59)
	Soma	89,29	74,61	69,41	73,39	69,72
	Produto	<b>89,60</b>	74,92	69,41	74,31	69,72
	Empilhamento	<b>89,60</b>	76,75	72,47	72,47	73,70
LA	Melhor Único	86,62 (3,48)	72,13 (2,60)	68,00 (2,15)	73,05 (2,99)	68,70 (2,06)
	Soma	87,46	75,84	70,64	75,53	72,17
	Produto	87,15	77,06	70,64	76,14	72,17
	Empilhamento	<b>88,68</b>	77,67	72,17	72,17	75,84
SLA	Melhor Único	86,32 (3,71)	72,90 (2,73)	68,84 (2,94)	73,40 (2,84)	70,90 (3,01)
	Soma	87,76	77,06	71,55	77,06	71,25
	Produto	87,76	77,98	72,17	77,67	71,55
	Empilhamento	<b>88,68</b>	81,34	75,84	73,70	75,84

Tabela 85 – Acurácia considerando o protocolo LOSO e um *pool* com 30 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	85,93 (2,57)	71,27 (2,47)	67,38 (2,10)	70,86 (2,44)	68,31 (1,97)
	Soma	88,37	75,22	70,64	75,84	70,33
	Produto	88,68	75,53	70,64	76,45	70,64
	Empilhamento	<b>89,29</b>	80,42	73,70	76,75	76,45
L	Melhor Único	87,78 (2,69)	74,71 (3,25)	66,48 (2,21)	76,17 (3,45)	68,53 (2,47)
	Soma	<b>89,60</b>	75,22	69,41	75,53	70,03
	Produto	89,29	76,75	70,33	75,84	70,33
	Empilhamento	88,68	79,51	74,61	73,39	74,00
SA	Melhor Único	87,62 (1,93)	70,05 (2,03)	66,37 (1,96)	71,15 (1,96)	66,77 (1,64)
	Soma	87,15	74,92	70,64	75,53	70,03
	Produto	<b>88,37</b>	75,84	70,64	75,84	70,03
	Empilhamento	88,07	79,51	72,78	74,31	74,92
SL	Melhor Único	88,58 (2,08)	71,28 (2,49)	68,98 (1,92)	71,63 (2,35)	69,20 (1,97)
	Soma	<b>88,99</b>	75,22	69,72	76,14	69,72
	Produto	<b>88,99</b>	76,75	69,72	76,14	69,72
	Empilhamento	<b>88,99</b>	80,73	74,92	75,53	74,00
LA	Melhor Único	87,27 (2,32)	74,01 (2,94)	67,69 (2,20)	72,74 (2,61)	68,81 (2,18)
	Soma	<b>88,07</b>	78,28	71,55	78,59	72,17
	Produto	<b>88,07</b>	79,81	71,86	79,20	72,17
	Empilhamento	87,76	79,20	74,00	74,61	75,53
SLA	Melhor Único	86,56 (2,52)	71,77 (2,62)	67,85 (2,64)	71,38 (2,53)	68,26 (2,48)
	Soma	87,46	75,84	71,86	76,14	72,17
	Produto	87,46	77,37	71,86	76,45	71,17
	Empilhamento	<b>88,68</b>	80,42	73,39	74,61	73,70

Tabela 86 – Acurácia considerando o protocolo LOSO e um *pool* com 40 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	86,10 (2,15)	69,67 (2,23)	65,90 (1,60)	69,71 (2,33)	68,37
	Soma	88,37	74,92	70,03	73,70	70,64
	Produto	88,68	76,75	70,33	74,31	70,94
	Empilhamento	<b>89,90</b>	81,34	74,00	75,53	74,61
L	Melhor Único	88,30 (2,51)	72,50 (2,96)	67,20 (2,38)	73,17 (2,92)	67,92 (2,34)
	Soma	88,68	75,53	72,17	75,53	72,17
	Produto	88,07	76,45	72,47	75,84	72,47
	Empilhamento	<b>89,29</b>	79,51	74,00	74,00	76,45
SA	Melhor Único	86,01 (1,67)	68,53 (1,86)	65,69 (1,97)	70,22 (1,98)	67,89 (1,95)
	Soma	87,46	75,84	70,64	74,92	71,55
	Produto	<b>87,76</b>	78,59	71,55	75,22	71,25
	Empilhamento	<b>87,76</b>	80,42	72,17	73,08	73,70
SL	Melhor Único	87,52 (2,81)	72,33 (2,41)	65,11 (1,94)	72,89 (2,61)	68,22 (2,14)
	Soma	<b>88,99</b>	73,70	68,80	72,78	69,41
	Produto	<b>88,99</b>	74,31	68,80	73,39	69,72
	Empilhamento	88,68	79,20	74,00	74,92	74,00
LA	Melhor Único	87,88 (2,72)	73,70 (3,20)	67,71 (2,47)	74,19 (3,10)	69,98 (2,37)
	Soma	<b>88,99</b>	76,45	71,86	76,75	71,86
	Produto	<b>88,99</b>	77,98	71,86	77,06	71,86
	Empilhamento	88,07	80,12	73,70	76,14	76,75
SLA	Melhor Único	87,50 (2,60)	73,37 (2,90)	68,78 (2,33)	73,75 (3,29)	68,58 (2,69)
	Soma	<b>88,37</b>	75,84	71,55	75,22	71,86
	Produto	88,07	76,75	71,55	75,84	71,86
	Empilhamento	<b>88,37</b>	78,28	74,00	75,53	76,45

Tabela 87 – Acurácia considerando o protocolo LOSO e um *pool* com 50 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	85,87 (2,14)	71,75 (2,43)	65,56 (1,84)	73,26 (2,74)	65,94 (1,52)
	Soma	89,29	74,00	69,11	73,70	69,72
	Produto	89,29	76,14	69,11	74,92	70,03
	Empilhamento	<b>89,60</b>	79,51	76,14	76,45	79,51
L	Melhor Único	87,13 (2,98)	70,70 (2,62)	65,38 (1,96)	70,83 (2,63)	66,86 (2,19)
	Soma	89,29	75,53	71,25	75,53	71,25
	Produto	<b>89,90</b>	76,75	70,94	76,45	71,25
	Empilhamento	89,60	79,20	74,61	75,84	79,81
SA	Melhor Único	86,00 (1,74)	69,81 (2,36)	68,95 (1,77)	70,35 (2,29)	68,05 (1,70)
	Soma	<b>88,68</b>	74,92	70,33	74,00	70,33
	Produto	<b>88,68</b>	75,84	70,33	74,00	70,64
	Empilhamento	88,07	80,12	75,22	74,61	78,59
SL	Melhor Único	88,32 (2,49)	70,56 (2,33)	67,69 (2,17)	71,27 (2,33)	68,16 (2,97)
	Soma	88,99	76,75	70,33	76,14	70,64
	Produto	<b>89,29</b>	77,06	70,94	76,75	71,25
	Empilhamento	88,68	79,51	76,14	73,70	81,03
LA	Melhor Único	87,62 (2,39)	73,47 (3,02)	69,75 (2,39)	72,81 (2,67)	68,54 (2,11)
	Soma	88,37	77,06	71,25	76,45	72,47
	Produto	<b>88,68</b>	77,06	71,25	76,75	72,47
	Empilhamento	87,46	81,34	75,22	77,37	78,28
SLA	Melhor Único	87,62 (2,67)	74,12 (3,62)	67,71 (2,29)	73,46 (3,42)	70,11 (2,37)
	Soma	87,15	77,06	71,25	75,84	71,25
	Produto	87,46	77,06	71,55	77,06	71,25
	Empilhamento	<b>87,76</b>	81,03	75,84	77,06	79,51

Tabela 88 – Acurácia considerando o protocolo LOSO e um *pool* com 70 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	87,34 (1,82)	69,75 (2,50)	67,13 (2,05)	71,73 (2,59)	68,87 (2,11)
	Soma	88,99	74,92	69,72	74,00	70,03
	Produto	<b>89,60</b>	74,61	69,72	74,31	70,03
	Empilhamento	<b>89,60</b>	77,98	75,53	75,53	79,20
L	Melhor Único	87,57 (3,06)	73,47 (3,30)	67,88 (2,13)	72,54 (3,32)	70,18 (2,45)
	Soma	88,99	75,53	70,03	75,53	70,33
	Produto	88,68	76,41	70,33	75,53	70,33
	Empilhamento	<b>89,90</b>	78,28	76,45	74,92	80,42
SA	Melhor Único	86,65 (2,19)	73,19 (2,93)	68,23 (1,99)	72,85 (2,72)	68,44 (1,83)
	Soma	88,07	76,45	71,86	75,22	71,86
	Produto	88,37	77,06	71,86	76,45	72,17
	Empilhamento	<b>89,60</b>	81,95	75,53	75,22	78,28
SL	Melhor Único	87,99 (2,44)	74,66 (2,88)	68,34 (2,34)	74,61 (2,73)	69,81 (2,30)
	Soma	89,29	76,45	71,55	74,92	70,94
	Produto	89,29	76,14	71,55	74,92	71,25
	Empilhamento	<b>89,90</b>	78,89	77,06	76,14	83,18
LA	Melhor Único	87,78 (2,52)	72,16 (2,30)	69,40 (2,21)	73,29 (2,44)	70,29 (2,44)
	Soma	88,07	76,45	71,86	75,84	71,86
	Produto	<b>88,37</b>	77,06	72,17	75,84	72,17
	Empilhamento	87,46	81,03	75,84	76,45	80,42
SLA	Melhor Único	87,27 (2,24)	71,83 (2,65)	68,46 (2,12)	72,82 (2,50)	69,66 (2,22)
	Soma	88,07	77,98	71,25	77,06	70,94
	Produto	<b>88,37</b>	76,75	71,25	77,37	70,94
	Empilhamento	88,07	81,34	76,14	74,92	79,81

Tabela 89 – Acurácia considerando o protocolo LOSO e um *pool* com 100 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	88,44 (2,20)	70,80 (2,37)	66,90 (1,88)	72,50 (2,52)	67,07 (1,92)
	Soma	88,99	73,39	69,72	73,08	69,41
	Produto	88,99	74,00	69,72	73,08	70,03
	Empilhamento	<b>89,60</b>	78,59	77,37	76,45	79,81
L	Melhor Único	91,01 (3,52)	74,53 (3,04)	68,16 (2,35)	74,29 (2,79)	69,02 (2,46)
	Soma	89,29	74,61	69,41	74,61	70,33
	Produto	89,60	75,84	69,41	75,22	70,33
	Empilhamento	<b>91,13</b>	80,12	76,14	77,06	80,73
SA	Melhor Único	87,76 (2,36)	72,28 (2,58)	68,30 (2,05)	71,93 (2,60)	69,19 (2,07)
	Soma	87,76	76,14	70,03	75,53	70,94
	Produto	88,07	76,75	70,33	76,14	71,55
	Empilhamento	<b>88,68</b>	81,65	75,22	76,14	79,51
SL	Melhor Único	88,68 (2,97)	73,00 (2,88)	67,90 (2,24)	72,54 (2,74)	68,82 (2,34)
	Soma	<b>88,99</b>	75,53	70,94	75,53	71,25
	Produto	<b>88,99</b>	76,14	71,25	76,75	71,25
	Empilhamento	88,68	78,28	77,67	75,53	81,95
LA	Melhor Único	87,10 (2,82)	73,43 (3,01)	67,93 (2,17)	74,09 (2,98)	69,61 (2,19)
	Soma	<b>87,76</b>	76,75	72,17	76,75	72,17
	Produto	<b>87,76</b>	77,98	72,47	77,37	72,17
	Empilhamento	87,46	81,65	76,14	76,45	81,03
SLA	Melhor Único	87,74 (2,58)	73,27 (2,75)	68,84 (2,29)	73,09 (2,73)	69,42 (2,36)
	Soma	87,76	77,67	71,86	76,45	71,86
	Produto	87,76	78,59	71,86	77,98	72,17
	Empilhamento	<b>88,07</b>	81,03	76,45	77,06	80,12

Tabela 90 – Acurácia considerando o protocolo LOSO e um *pool* com 150 representações geradas com o método proposto. LFW e CK+ são conjuntos de dados auxiliares e de destino, respectivamente. Modelos REF baseados em um conjunto de SVMs, RF, BG e seleção dinâmica baseada em KnoraU em uma coleção de DTs e RF. As estratégias S, L e A representam diferentes sementes, vetores latentes e arquiteturas AEC. Para cada conjunto, o melhor classificador e as seguintes estratégias de fusão: soma, produto e empilhamento. (\*) é o desvio padrão da precisão do *pool*. Os melhores resultados estão em negrito.

Estratégia		Pool de SVMs	Métodos <i>Ensembles</i>		Seleção Dinâmica (KU)	
			BG	RF	DT	RF
S	Melhor Único	86,15 (2,17)	73,12 (2,64)	66,73 (2,13)	73,00 (2,59)	69,75 (2,18)
	Soma	89,60	74,92	69,11	74,61	70,03
	Produto	<b>89,90</b>	75,53	69,11	75,22	70,03
	Empilhamento	89,60	77,98	75,84	75,22	79,51
L	Melhor Único	90,45 (4,02)	72,96 (3,04)	68,12 (2,43)	72,33 (2,94)	68,17 (2,34)
	Soma	88,99	75,22	70,94	74,92	71,25
	Produto	88,68	76,75	71,25	75,53	71,25
	Empilhamento	90,51	79,20	76,45	76,14	81,03
SA	Melhor Único	87,41 (2,28)	73,54 (2,45)	67,55 (2,14)	73,82 (2,51)	68,67 (2,10)
	Soma	88,07	75,84	71,25	74,61	71,55
	Produto	88,37	77,06	71,55	75,53	71,86
	Empilhamento	<b>88,68</b>	80,73	75,22	75,22	80,73
SL	Melhor Único	89,19 (2,86)	74,61 (2,86)	68,10 (2,27)	72,54 (2,72)	70,15 (2,28)
	Soma	<b>89,60</b>	75,22	70,64	74,92	70,64
	Produto	<b>89,60</b>	75,84	70,94	75,53	70,94
	Empilhamento	88,99	78,59	75,84	74,61	81,65
LA	Melhor Único	87,93 (1,99)	74,33 (2,68)	67,98 (2,30)	74,54 (2,73)	69,97 (2,31)
	Soma	<b>88,07</b>	76,45	71,86	76,14	72,78
	Produto	87,76	77,37	71,86	76,45	72,78
	Empilhamento	87,76	81,65	77,06	77,06	80,42
SLA	Melhor Único	87,55 (2,25)	73,96 (2,92)	68,55 (2,21)	74,47 (2,84)	69,26 (2,27)
	Soma	88,68	75,84	71,86	75,53	72,17
	Produto	89,60	76,14	71,86	75,84	72,47
	Empilhamento	88,37	81,65	75,84	76,75	80,12

## Referências

- AGGARWAL, Charu C. *Data classification: algorithms and applications*. [S.l.]: CRC press, 2014. Citado na página 67.
- AIFANTI, Niki; PAPACHRISTOU, Christos; DELOPOULOS, Anastasios. The mug facial expression database. In: IEEE. *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*. [S.l.], 2010. p. 1–4. Citado 2 vezes nas páginas 74 e 75.
- AKHAND, MAH; ROY, Shuvendu; SIDDIQUE, Nazmul; KAMAL, Md Abdus Samad; SHIMAMURA, Tetsuya. Facial emotion recognition using transfer learning in the deep cnn. *Electronics*, MDPI, v. 10, n. 9, p. 1036, 2021. Citado na página 72.
- ALLOGNON, Sevegni Odilon Clement; BRITTO, Alceu de S; KOERICH, Alessandro L. Continuous emotion recognition via deep convolutional autoencoder and support vector regressor. In: IEEE. *2020 International Joint Conference on Neural Networks (IJCNN)*. [S.l.], 2020. p. 1–8. Citado na página 40.
- ALSHAREKH, Mohammed F. Facial emotion recognition in verbal communication based on deep learning. *Sensors*, MDPI, v. 22, n. 16, p. 6105, 2022. Citado na página 46.
- ALZUBAIDI, Laith; ZHANG, Jinglan; HUMAIDI, Amjad J; AL-DUJAILI, Ayad; DUAN, Ye; AL-SHAMMA, Omran; SANTAMARÍA, José; FADHEL, Mohammed A; AL-AMIDIE, Muthana; FARHAN, Laith. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, Springer, v. 8, p. 1–74, 2021. Citado na página 128.
- ANTONIADIS, Anestis; LAMBERT-LACROIX, Sophie; POGGI, Jean-Michel. Random forests for global sensitivity analysis: A selective review. *Reliability Engineering & System Safety*, Elsevier, v. 206, p. 107312, 2021. Citado na página 50.
- AWAD, Mariette; KHANNA, Rahul; AWAD, Mariette; KHANNA, Rahul. Support vector regression. *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, Springer, p. 67–80, 2015. Citado na página 81.
- BARROS, Pablo; MAGG, Sven; WEBER, Cornelius; WERMTER, Stefan. A multichannel convolutional neural network for hand posture recognition. In: . [S.l.: s.n.], 2014. p. 403–410. ISBN 978-3-319-11178-0. Citado na página 73.
- BASTIEN, Frederic; BENGIO, Y.; BERGERON, Arnaud; BOULANGER-LEWANDOWSKI, Nicolas; BREUEL, Thomas; CHHERAWALA, Youssof; CISSE, Moustapha; Côté, Myriam; ERHAN, Dumitru; EUSTACHE, Jeremy; GLOROT, Xavier; MULLER, Xavier; LEBEUF, Sylvain; PASCANU, Razvan; RIFAI, Salah; SAVARD, Francois; SICARD, Guillaume. Deep self-taught learning for handwritten character recognition. *Mach. Learn.*, v. 2, 09 2010. Citado na página 40.
- BAUDAT, Gaston; ANOUAR, Fatiha. Generalized discriminant analysis using a kernel approach. *Neural computation*, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 12, n. 10, p. 2385–2404, 2000. Citado 2 vezes nas páginas 83 e 84.

- BAY, Herbert; ESS, Andreas; TUYTELAARS, T; GOOL, V. “speeded-up robust features (surf). computer vision and image understanding (cviu). In: *Proc of the 9th European Conference on Computer Vision. Austria: Springer*. [S.l.: s.n.], 2006. Citado na página 40.
- BELL, Anthony J; SEJNOWSKI, Terrence J. The “independent components” of natural scenes are edge filters. *Vision research*, Elsevier, v. 37, n. 23, p. 3327–3338, 1997. Citado 3 vezes nas páginas 80, 81 e 82.
- BENDJILLALI, Ridha Ilyas; BELADGHAM, Mohammed; MERIT, Khaled; TALEB-AHMED, Abdelmalik. Improved facial expression recognition based on dwt feature for deep cnn. *Electronics*, MDPI, v. 8, n. 3, p. 324, 2019. Citado na página 46.
- BENGIO, Yoshua; BASTIEN, Frédéric; BERGERON, Arnaud; BOULANGER-LEWANDOWSKI, Nicolas; BREUEL, Thomas; CHHERAWALA, Youssouf; CISSE, Moustapha; CÔTÉ, Myriam; ERHAN, Dumitru; EUSTACHE, Jeremy et al. Deep learners benefit more from out-of-distribution examples. In: *JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. [S.l.], 2011. p. 164–172. Citado 4 vezes nas páginas 54, 55, 79 e 83.
- BHANDARI, Piyush; BIJARNIYA, Rakesh Kumar; CHATTERJEE, Subhamoy; KOLEKAR, Maheshkumar. Analysis for self-taught and transfer learning based approaches for emotion recognition. In: *International Conference on Signal Processing and Integrated Networks (SPIN)*. [S.l.: s.n.], 2018. Citado 3 vezes nas páginas 81, 82 e 112.
- BIAU, Gérard; SCORNET, Erwan. A random forest guided tour. *Test*, Springer, v. 25, n. 2, p. 197–227, 2016. Citado na página 62.
- BISWAS, Suparna; SIL, Jaya. An efficient expression recognition method using contourlet transform. In: *Proceedings of the 2nd International Conference on Perception and Machine Intelligence*. [S.l.: s.n.], 2015. p. 167–174. Citado na página 44.
- BONACCORSO, Giuseppe. *Machine Learning Algorithms*. [S.l.]: Packt Publishing Ltd, 2017. Citado na página 50.
- BREIMAN, Leo. Bagging predictors. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996. Citado 3 vezes nas páginas 63, 77 e 78.
- BREIMAN, Leo. Random forests. *Machine learning*, Springer, v. 45, n. 1, p. 5–32, 2001. Citado 4 vezes nas páginas 62, 63, 81 e 85.
- BRERETON, Richard G; LLOYD, Gavin R. Support vector machines for classification and regression. *Analyst*, Royal Society of Chemistry, v. 135, n. 2, p. 230–267, 2010. Citado na página 80.
- BROWNLEE, Jason. *Master Machine Learning Algorithms*. [S.l.]: Springer, 2016. Citado na página 50.
- BROWNLEE, Jason. *Master Machine Learning Algorithms: discover how they work and implement them from scratch*. [S.l.]: Machine Learning Mastery, 2016. Citado na página 62.
- BROWNLEE, Jason. *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. [S.l.]: Machine Learning Mastery, 2018. Citado na página 55.

- CAI, Deng; HE, Xiaofei; HAN, Jiawei. Spectral regression for efficient regularized subspace learning. In: IEEE. *2007 IEEE 11th international conference on computer vision*. [S.l.], 2007. p. 1–8. Citado na página 81.
- CALVO, Manuel G; LUNDQVIST, Daniel. Facial expressions of emotion (kdef): Identification under different display-duration conditions. *Behavior research methods*, Springer, v. 40, n. 1, p. 109–115, 2008. Citado na página 72.
- CANAL, Felipe Zago; MÜLLER, Tobias Rossi; MATIAS, Jhennifer Cristine; SCOTTON, Gustavo Gino; JUNIOR, Antonio Reis de Sa; POZZEBON, Eliane; SOBIERANSKI, Antonio Carlos. A survey on facial emotion recognition techniques: A state-of-the-art literature review. *Information Sciences*, Elsevier, v. 582, p. 593–617, 2022. Citado 4 vezes nas páginas 39, 40, 44 e 54.
- CANEDO, Daniel; NEVES, António JR. Facial expression recognition using computer vision: A systematic review. *Applied Sciences*, MDPI, v. 9, n. 21, p. 4678, 2019. Citado na página 46.
- CANO, Pedro; GÓMEZ, Emilia; GOUYON, Fabien; HERRERA, Perfecto; KOPPENBERGER, Markus; ONG, Beesuan; SERRA, Xavier; STREICH, Sebastian; WACK, Nicolas. Ismir 2004 audio description contest. *Music Technology Group of the Universitat Pompeu Fabra, Tech. Rep*, 2006. Citado na página 83.
- CELEBI, Emre M.; AYDIN, Kemal. *Unsupervised Learning Algorithms*. [S.l.]: Springer International Publishing, 2016. 10.1007/978-3-319-24211-8. Citado na página 50.
- CERVANTES, Jair; GARCIA-LAMONT, Farid; RODRÍGUEZ-MAZAHUA, Lisbeth; LOPEZ, Asdrubal. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, Elsevier, v. 408, p. 189–215, 2020. Citado na página 50.
- CHANDRA, Mayank Arya; BEDI, SS. Survey on svm and their application in image classification. *International Journal of Information Technology*, Springer, v. 13, p. 1–11, 2021. Citado na página 62.
- CHARTE, David; CHARTE, Francisco; GARCÍA, Salvador; JESUS, María J del; HERRERA, Francisco. A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, Elsevier, v. 44, p. 78–96, 2018. Citado 2 vezes nas páginas 54 e 60.
- CHEN, Li-Fen; YEN, Yu-Shiuan. Taiwanese facial expression image database. *Brain Mapping Laboratory, Institute of Brain Science, National Yang-Ming University, Taipei, Taiwan*, 2007. Citado na página 74.
- CHEN, Yanbei; MANCINI, Massimiliano; ZHU, Xiatian; AKATA, Zeynep. Semi-supervised and unsupervised deep visual learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, 2022. Citado na página 51.
- CHEPLYGINA, Veronika; BRUIJNE, Marleen de; PLUIM, Josien PW. Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical image analysis*, Elsevier, v. 54, p. 280–296, 2019. Citado na página 51.

- CHOWDARY, M Kalpana; NGUYEN, Tu N; HEMANTH, D Jude. Deep learning-based facial emotion recognition for human-computer interaction applications. *Neural Computing and Applications*, Springer, p. 1–18, 2021. Citado na página 71.
- CIERI, Chris; GRAFF, David; LIBERMAN, Mark; MARTEY, Nii; STRASSEL, Stephanie et al. The tdt-2 text and speech corpus. In: *Proceedings of the DARPA Broadcast News workshop*. [S.l.: s.n.], 1999. p. 57–60. Citado na página 84.
- COATES, Adam; NG, Andrew; LEE, Honglak. An analysis of single-layer networks in unsupervised feature learning. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. [S.l.: s.n.], 2011. p. 215–223. Citado 3 vezes nas páginas 101, 103 e 104.
- COMON, Pierre. Independent component analysis, a new concept? *Signal processing*, Elsevier, v. 36, n. 3, p. 287–314, 1994. Citado na página 54.
- COOTES, T.F.; EDWARDS, G.J.; TAYLOR, Christopher. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, v. 23, p. 681 – 685, 07 2001. Citado na página 73.
- CRAVEN, Mark; DIPASQUO, Dan; FREITAG, Dayne; MCCALLUM, Andrew; MITCHELL, Tom; NIGAM, Kamal; SLATTERY, Seán. Learning to extract symbolic knowledge from the world wide web. *AAAI/IAAI*, v. 3, n. 3.6, p. 2, 1998. Citado na página 84.
- CROW, Franklin C. Summed-area tables for texture mapping. In: *SIGGRAPH '84*. [S.l.: s.n.], 1984. Citado na página 47.
- DALAL, Navneet; TRIGGS, Bill. Histograms of oriented gradients for human detection. In: IEEE. *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. [S.l.], 2005. v. 1, p. 886–893. Citado 2 vezes nas páginas 40 e 74.
- DENG, Jia; DONG, Wei; SOCHER, Richard; LI, Li-Jia; LI, Kai; FEI-FEI, Li. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255. Citado na página 103.
- DHALL, Abhinav; GOECKE, Roland; LUCEY, Simon; GEDEON, Tom. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: IEEE. *2011 IEEE international conference on computer vision workshops (ICCV workshops)*. [S.l.], 2011. p. 2106–2112. Citado na página 75.
- DHALL, Abhinav; GOECKE, Roland; LUCEY, Simon; GEDEON, Tom et al. Collecting large, richly annotated facial-expression databases from movies. *IEEE multimedia*, v. 19, n. 3, p. 34, 2012. Citado na página 75.
- DHANKHAR, Poonam. Resnet-50 and vgg-16 for recognizing facial emotions. *International Journal of Innovations in Engineering and Technology (IJIET)*, v. 13, n. 4, p. 126–130, 2019. Citado na página 71.
- DOI, E.; INUI, T.; LEE, T.-W.; WACHTLER, T.; SEJNOWSKI, T. J. Spatiochromatic receptive field properties derived from information-theoretic analyses of cone mosaic responses to natural scenes. *Neural Computation*, v. 15, p. 397–417, 2003. Citado 4 vezes nas páginas 80, 81, 101 e 102.

- DU, Lingshuang; HU, Haifeng. Weighted patch-based manifold regularization dictionary pair learning model for facial expression recognition using iterative optimization classification strategy. *Computer Vision and Image Understanding*, Elsevier, v. 186, p. 13–24, 2019. Citado 2 vezes nas páginas 75 e 131.
- ENGELEN, Jesper E Van; HOOS, Holger H. A survey on semi-supervised learning. *Machine learning*, Springer, v. 109, n. 2, p. 373–440, 2020. Citado 2 vezes nas páginas 49 e 50.
- ERHAN, Dumitru; COURVILLE, Aaron; BENGIO, Yoshua; VINCENT, Pascal. Why does unsupervised pre-training help deep learning? In: JMLR WORKSHOP AND CONFERENCE PROCEEDINGS. *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. [S.l.], 2010. p. 201–208. Citado 2 vezes nas páginas 54 e 79.
- FAYYAZ, Mohsen; SAFFAR, Mohammad Hajizadeh; SABOKROU, Mohammad; MOJTABA, H.; FATHY, Mahmood. A novel approach for finger vein verification based on self-taught learning. In: . [S.l.: s.n.], 2015. p. 88–91. Citado 2 vezes nas páginas 56 e 86.
- FENG, Siwei; YU, Han; DUARTE, Marco F. Autoencoder based sample selection for self-taught learning. *Knowledge-Based Systems*, Elsevier, v. 192, p. 105343, 2020. Citado na página 40.
- FERNÁNDEZ-DELGADO, Manuel; SIRSAT, Manisha Sanjay; CERNADAS, Eva; ALAWADI, Sadi; BARRO, Senén; FEBRERO-BANDE, Manuel. An extensive experimental survey of regression methods. *Neural Networks*, Elsevier, v. 111, p. 11–34, 2019. Citado na página 50.
- FERNANDEZ, Pedro D Marrero; PENA, Fidel A Guerrero; REN, Tsang; CUNHA, Alexandre. Feratt: Facial expression recognition with attention net. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2019. p. 0–0. Citado na página 72.
- FLOREA, Corneliu; FLOREA, Laura; BUTNARU, Raluca; BANDRABUR, Alessandra; VERTAN, Constantin. Pain intensity estimation by a self-taught selection of histograms of topographical features. *Image and Vision Computing*, v. 56, p. 13–27, 2016. Citado 2 vezes nas páginas 81 e 82.
- FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, v. 55, p. 119–139, 1997. Citado na página 47.
- FRIEDMAN, HW; FRIEDMAN, JH; FRIEDMAN, J. Another approach to polychotomous classification. 1996. Citado na página 62.
- FRIEDMAN, Jerome; HASTIE, Trevor; TIBSHIRANI, Robert. *The elements of statistical learning*. [S.l.]: Springer series in statistics New York, 2001. v. 1. Citado na página 50.
- FULLER, Wayne A. *Sampling statistics*. [S.l.]: John Wiley & Sons, 2011. Citado 2 vezes nas páginas 80 e 86.
- GAN, Junying; LI, Lichen; ZHAI, Yikui; LIU, Yinhua. Deep self-taught learning for facial beauty prediction. *Neurocomputing*, v. 144, p. 295–303, 11 2014. Citado 3 vezes nas páginas 52, 80 e 82.

GANAIIE, Mudasir A; HU, Minghui; MALIK, AK; TANVEER, M; SUGANTHAN, PN. Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 115, p. 105151, 2022. Citado na página 39.

GAO, Wen; CAO, Bo; SHAN, Shiguang; CHEN, Xilin; ZHOU, Delong; ZHANG, Xiaohua; ZHAO, Debin. The cas-peal large-scale chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, v. 38, n. 1, p. 149 – 161, 01 2008. Citado na página 80.

GÉRON, Aurélien. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. [S.l.]: O'Reilly Media, 2019. Citado na página 57.

GHIMIRE, Deepak; LEE, Joonwhoan. Extreme learning machine ensemble using bagging for facial expression recognition. *Journal of Information Processing Systems*, v. 10, p. 443–458, 09 2014. Citado 2 vezes nas páginas 77 e 79.

GOELEVELN, Ellen; RAEDT, Rudi De; LEYMAN, Lemke; VERSCHUERE, Bruno. The karolinska directed emotional faces: a validation study. *Cognition and emotion*, Taylor & Francis, v. 22, n. 6, p. 1094–1118, 2008. Citado 2 vezes nas páginas 72 e 74.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. *Deep Learning*. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>. Citado 2 vezes nas páginas 55 e 56.

GOODFELLOW, Ian J; ERHAN, Dumitru; CARRIER, Pierre Luc; COURVILLE, Aaron; MIRZA, Mehdi; HAMNER, Ben; CUKIERSKI, Will; TANG, Yichuan; THALER, David; LEE, Dong-Hyun et al. Challenges in representation learning: A report on three machine learning contests. In: SPRINGER. *Neural Information Processing: 20th International Conference, ICONIP 2013, Daegu, Korea, November 3-7, 2013. Proceedings, Part III 20*. [S.l.], 2013. p. 117–124. Citado 2 vezes nas páginas 73 e 78.

GROSS, Ralph; MATTHEWS, Iain; COHN, Jeffrey; KANADE, Takeo; BAKER, Simon. Multi-pie. *Image and vision computing*, Elsevier, v. 28, n. 5, p. 807–813, 2010. Citado na página 75.

GROSSE, Roger B.; RAINA, Rajat; KWONG, Helen; NG, Andrew Y. Shift-invariance sparse coding for audio classification. In: *Proceedings of the Twenty-third Conference on Uncertainty in Artificial Intelligence (UAI)*. [S.l.: s.n.], 2007. Citado na página 83.

HAMESTER, Dennis; BARROS, Pablo; WERMTER, Stefan. Face expression recognition with a 2-channel convolutional neural network. In: . [S.l.: s.n.], 2015. p. 1–8. Citado na página 73.

HAPPY, SL; PATNAIK, Priyadarshi; ROUTRAY, Aurobinda; GUHA, Rajlakshmi. The indian spontaneous expression database for emotion recognition. *IEEE Transactions on Affective Computing*, IEEE, v. 8, n. 1, p. 131–142, 2015. Citado na página 75.

HARALICK, Robert M; SHANMUGAM, Karthikeyan; DINSTEN, Its' Hak. Textural features for image classification. *IEEE Transactions on systems, man, and cybernetics*, Ieee, n. 6, p. 610–621, 1973. Citado na página 40.

HASANUZZAMAN, Mohammed; DIAS, Gael; WAY, Andy. Demographic word embeddings for racism detection on twitter. In: . [S.l.]: Asian Federation of Natural Language Processing, 2017. Citado na página 84.

- HASTIE, Trevor; TIBSHIRANI, Robert; FRIEDMAN, Jerome. The elements of statistical learning. springer series in statistics. *New York, NY, USA*, 2001. Citado na página 80.
- HATEREN, J. Van; SCHAAF, A. van der. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proceedings: Biological Sciences*, v. 265, n. 1394, p. 359–366, Mar 1998. Citado na página 81.
- HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado na página 128.
- HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Identity mappings in deep residual networks. In: SPRINGER. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*. [S.l.], 2016. p. 630–645. Citado na página 72.
- HE, Peilin; JIA, Pengfei; QIAO, Siqi; DUAN, Shukai. Self-taught learning based on sparse autoencoder for e-nose in wound infection detection. *Sensors*, MDPI, v. 17, n. 10, p. 2279, 2017. Citado 2 vezes nas páginas 80 e 86.
- HERSH, William; BUCKLEY, Chris; LEONE, TJ; HICKAM, David. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In: SPRINGER. *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*. [S.l.], 1994. p. 192–201. Citado na página 84.
- HICKSON, Steven; DUFOUR, Nick; SUD, Avneesh; KWATRA, Vivek; ESSA, Irfan. Eyemotion: Classifying facial expressions in vr using eye-tracking cameras. In: IEEE. *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. [S.l.], 2019. p. 1626–1635. Citado na página 45.
- HINTON, Geoffrey E; OSINDERO, Simon; TEH, Yee-Whye. A fast learning algorithm for deep belief nets. *Neural computation*, MIT Press, v. 18, n. 7, p. 1527–1554, 2006. Citado na página 54.
- HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long short-term memory. *Neural computation*, MIT press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 85.
- HOU, Cheng-An; YANG, Min-Chun; WANG, Yu-Chiang Frank. Domain adaptive self-taught learning for heterogeneous face recognition. In: . [S.l.: s.n.], 2014. p. 3068–3073. Citado na página 86.
- HOWARD, Andrew G; ZHU, Menglong; CHEN, Bo; KALENICHENKO, Dmitry; WANG, Weijun; WEYAND, Tobias; ANDREETTO, Marco; ADAM, Hartwig. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. Citado na página 71.
- HU, Han; PHAN, NhatHai; CHUN, Soon A; GELLER, James; VO, Huy; YE, Xinyue; JIN, Ruoming; DING, Kele; KENNE, Deric; DOU, Dejing. An insight analysis and detection of drug-abuse risk behavior on twitter with self-taught deep learning. *Computational Social Networks*, SpringerOpen, v. 6, n. 1, p. 1–19, 2019. Citado 2 vezes nas páginas 80 e 84.

- HUANG, Chen-Chun; WU, Yi-Leih; TANG, Cheng-Yuan. Human face sentiment classification using synthetic sentiment images with deep convolutional neural networks. In: IEEE. *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*. [S.l.], 2019. p. 1–5. Citado na página 76.
- HUANG, Gao; LIU, Zhuang; MAATEN, Laurens Van Der; WEINBERGER, Kilian Q. Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 4700–4708. Citado na página 128.
- HUANG, Gary B.; MATTAR, Marwan A.; LEE, Honglak; LEARNED-MILLER, Erik G. Learning to align from scratch. In: *NIPS*. [S.l.: s.n.], 2012. p. 764–772. Citado 2 vezes nas páginas 80 e 82.
- HUANG, Gary B.; RAMESH, Manu; BERG, Tamara; LEARNED-MILLER, Erik. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments*. [S.l.], 2007. Citado 4 vezes nas páginas 80, 101, 102 e 103.
- HUANG, Guang-Bin; ZHU, Qin-Yu; SIEW, Chee-Kheong. Extreme learning machine: theory and applications. *Neurocomputing*, Elsevier, v. 70, n. 1-3, p. 489–501, 2006. Citado na página 77.
- HUANG, Kaizhu; XU, Zenglin; KING, Irwin; LYU, Michael; CAMPBELL, Colin. Supervised self-taught learning: Actively transferring knowledge from unlabeled data. In: . [S.l.: s.n.], 2009. p. 1272–1277. Citado 2 vezes nas páginas 52 e 84.
- HULL, Jonathan J. A database for handwritten text recognition research. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 16, n. 5, p. 550–554, 1994. Citado na página 84.
- IRGENS, Peter; BADER, Curtis; LÉ, Theresa; SAXENA, Devansh; ABABEI, Cristinel. An efficient and cost effective fpga based implementation of the viola-jones face detection algorithm. *HardwareX*, v. 1, 03 2017. Citado na página 47.
- JAGADEESH, M; BARANIDHARAN, B. Facial expression recognition of online learners from real-time videos using a novel deep learning model. *Multimedia Systems*, Springer, v. 28, n. 6, p. 2285–2305, 2022. Citado na página 46.
- JAIN, Deepak Kumar; DUTTA, Ashit Kumar; VERDÚ, Elena; ALSUBAI, Shtwai; SAIT, Abdul Rahaman Wahab. An automated hyperparameter tuned deep learning model enabled facial emotion recognition for autonomous vehicle drivers. *Image and Vision Computing*, Elsevier, v. 133, p. 104659, 2023. Citado na página 45.
- JAIN, Deepak Kumar; SHAMSOLMOALI, Pourya; SEHDEV, Paramjit. Extended deep neural network for facial emotion recognition. *Pattern Recognition Letters*, Elsevier, v. 120, p. 69–74, 2019. Citado na página 71.
- JINGJING, He. Self-taught learning features for human action recognition. In: . [S.l.: s.n.], 2016. p. 611–615. Citado na página 85.
- JOLLIFFE, Ian T; CADIMA, Jorge. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, The Royal Society Publishing, v. 374, n. 2065, p. 20150202, 2016. Citado 4 vezes nas páginas 54, 74, 80 e 85.

- JR, Alceu S Britto; SABOURIN, Robert; OLIVEIRA, Luiz ES. Dynamic selection of classifiers—a comprehensive review. *Pattern recognition*, Elsevier, v. 47, n. 11, p. 3665–3680, 2014. Citado na página 64.
- KAMATH, Uday; LIU, John; WHITAKER, James. *Deep learning for NLP and speech recognition*. [S.l.]: Springer, 2019. v. 84. Citado 2 vezes nas páginas 54 e 80.
- KANADE, Takeo; COHN, Jeffrey F; TIAN, Yingli. Comprehensive database for facial expression analysis. In: IEEE. *Proceedings fourth IEEE international conference on automatic face and gesture recognition (cat. No. PR00580)*. [S.l.], 2000. p. 46–53. Citado na página 75.
- KANADE, Takeo; TIAN, Ying li; COHN, Jeffrey F. Comprehensive database for facial expression analysis. *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, p. 46–53, 2000. Citado 2 vezes nas páginas 78 e 81.
- KANAN, Christopher; COTTREL, Garrison. Robust classification of objects, faces, and flowers using natural image statistics. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. Citado na página 82.
- KARIMPOULI, Sadegh; TAHMASEBI, Pejman. Segmentation of digital rock images using deep convolutional autoencoder networks. *Computers & geosciences*, Elsevier, v. 126, p. 142–150, 2019. Citado na página 59.
- KARTHEEK, Mukku Nisanth; PRASAD, Munaga VNK; BHUKYA, Raju. Radial mesh pattern: a handcrafted feature descriptor for facial expression recognition. *Journal of Ambient Intelligence and Humanized Computing*, Springer, v. 14, n. 3, p. 1619–1631, 2023. Citado 2 vezes nas páginas 74 e 131.
- KAZEMI, Vahid; SULLIVAN, Josephine. One millisecond face alignment with an ensemble of regression trees. In: . [S.l.: s.n.], 2014. Citado na página 48.
- KHAN, Salman; RAHMANI, Hossein; SHAH, Syed Afaq Ali; BENNAMOUN, Mohammed. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, Morgan & Claypool Publishers, v. 8, n. 1, p. 1–207, 2018. Citado 3 vezes nas páginas 58, 59 e 60.
- KIM, Mooseop; LEE, Deok Gyu; KIM, Ki-Young. System architecture for real-time face detection on analog video camera. *International Journal of Distributed Sensor Networks*, v. 2015, p. 1–11, 05 2015. Citado na página 48.
- KING, Davis. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, v. 10, p. 1755–1758, 07 2009. Citado na página 48.
- KITTLER, J.; HATER, M.; DUIN, Robert. On combining classifiers. In: . [S.l.: s.n.], 1996. v. 20, p. 897 – 901 vol.2. Citado 2 vezes nas páginas 65 e 76.
- KO, Albert HR; SABOURIN, Robert; JR, Alceu Souza Britto. From dynamic classifier selection to dynamic ensemble selection. *Pattern recognition*, Elsevier, v. 41, n. 5, p. 1718–1731, 2008. Citado 2 vezes nas páginas 64 e 65.

- KOELSTRA, Sander; MUHL, Christian; SOLEYMANI, Mohammad; LEE, Jong-Seok; YAZDANI, Ashkan; EBRAHIMI, Touradj; PUN, Thierry; NIJHOLT, Anton; PATRAS, Ioannis. Deap: A database for emotion analysis; using physiological signals. *IEEE transactions on affective computing*, IEEE, v. 3, n. 1, p. 18–31, 2011. Citado na página 71.
- KOLA, Durga G Rao; SAMAYAMANTULA, Srinivas K. Facial expression recognition using singular values and wavelet-based lgc-hd operator. *IET Biometrics*, Wiley Online Library, v. 10, n. 2, p. 207–218, 2021. Citado 2 vezes nas páginas 74 e 131.
- KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 6 vezes nas páginas 57, 71, 72, 76, 78 e 85.
- KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, AcM New York, NY, USA, v. 60, n. 6, p. 84–90, 2017. Citado 2 vezes nas páginas 71 e 128.
- KUNCHEVA, Ludmila I; WHITAKER, Christopher J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, Springer Nature BV, v. 51, n. 2, p. 181, 2003. Citado na página 111.
- KUO, Bor-Chen; HO, Hsin-Hua; LI, Cheng-Hsuan; HUNG, Chih-Cheng; TAUR, Jin-Shiuh. A kernel-based feature selection method for svm with rbf kernel for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, IEEE, v. 7, n. 1, p. 317–326, 2013. Citado na página 62.
- LANG, Ken. Newsweeder: Learning to filter netnews. In: *Machine learning proceedings 1995*. [S.l.]: Elsevier, 1995. p. 331–339. Citado na página 84.
- LANGNER, Oliver; DOTSCHE, Ron; BIJLSTRA, Gijsbert; WIGBOLDUS, Daniel HJ; HAWK, Skyler T; KNIPPENBERG, AD Van. Presentation and validation of the radboud faces database. *Cognition and emotion*, Taylor & Francis, v. 24, n. 8, p. 1377–1388, 2010. Citado na página 75.
- LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998. Citado na página 84.
- LECUN, Yann et al. Generalization and network design strategies. *Connectionism in perspective*, Elsevier Amsterdam, v. 19, p. 143–155, 1989. Citado na página 57.
- LEE, Daniel D; SEUNG, H Sebastian. Learning the parts of objects by non-negative matrix factorization. *Nature*, Nature Publishing Group, v. 401, n. 6755, p. 788–791, 1999. Citado 2 vezes nas páginas 54 e 83.
- LEE, Honglak; GROSSE, Roger; RANGANATH, Rajesh; NG, Andrew Y. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: *Proceedings of the 26th annual international conference on machine learning*. [S.l.: s.n.], 2009. p. 609–616. Citado na página 80.
- LEE, Honglak; GROSSE, Roger; RANGANATH, Rajesh; NG, Andrew Y. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, ACM New York, NY, USA, v. 54, n. 10, p. 95–103, 2011. Citado na página 52.

- LEE, Honglak; RAINA, Rajat; TEICHMAN, Alex; NG, Andrew. Exponential family sparse coding with application to self-taught learning with text documents. 01 2008. Citado na página 52.
- LEE, Honglak; RAINA, Rajat; TEICHMAN, Alex; NG, Andrew. Exponential family sparse coding with application to self-taught learning. In: . [S.l.: s.n.], 2009. p. 1113–1119. Citado na página 84.
- LEE, Seung Ho; BADDAR, Wissam J; RO, Yong Man. Collaborative expression representation using peak expression and intra class variation face images for practical subject-independent emotion recognition in videos. *Pattern Recognition*, Elsevier, v. 54, p. 52–67, 2016. Citado 2 vezes nas páginas 76 e 131.
- LENG, Biao; GUO, Shuang; ZHANG, Xiangyang; XIONG, Zhang. 3d object retrieval with stacked local convolutional autoencoder. *Signal Processing*, Elsevier, v. 112, p. 119–128, 2015. Citado 2 vezes nas páginas 57 e 60.
- LEWIS. *Reuters-21578 Dataset*. 1987. Disponível em: <<http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html>>. Citado na página 84.
- LEWIS, David D. Naive (bayes) at forty: The independence assumption in information retrieval. In: SPRINGER. *European conference on machine learning*. [S.l.], 1998. p. 4–15. Citado na página 85.
- LI, Rui; REN, Chao; ZHANG, Xiaowei; HU, Bin. A novel ensemble learning method using multiple objective particle swarm optimization for subject-independent eeg-based emotion recognition. *Computers in biology and medicine*, Elsevier, v. 140, p. 105080, 2022. Citado na página 71.
- LI, Shan; DENG, Weihong. Deep facial expression recognition: A survey. 04 2018. Citado na página 71.
- LI, Shan; DENG, Weihong. Deep facial expression recognition: A survey. *IEEE transactions on affective computing*, IEEE, v. 13, n. 3, p. 1195–1215, 2020. Citado 2 vezes nas páginas 39 e 48.
- LI, Sheng; LI, Kang; FU, Yun. Self-taught low-rank coding for visual learning. *IEEE Transactions on Neural Networks and Learning Systems*, PP, p. 1–12, 01 2017. Citado na página 80.
- LI, Stan; YI, Dong; LEI, Zhen; LIAO, Shengcai. The casia nir-vis 2.0 face database. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2013. p. 348–353. Citado na página 86.
- LIENHART, Rainer; MAYDT, Jochen. An extended set of haar-like features for rapid object detection. In: . [S.l.: s.n.], 2002. v. 1, p. I–900. Citado na página 46.
- LIU, Kuang; ZHANG, Mingmin; PAN, Zhigeng. Facial expression recognition with cnn ensemble. In: IEEE. *2016 international conference on cyberworlds (CW)*. [S.l.], 2016. p. 163–166. Citado 2 vezes nas páginas 78 e 79.

- LIU, Ling; WEI, Wenqi; CHOW, Ka-Ho; LOPER, Margaret; GURSOY, Emre; TRUEX, Stacey; WU, Yanzhao. Deep neural network ensembles against deception: Ensemble diversity, accuracy and robustness. In: IEEE. *2019 IEEE 16th international conference on mobile ad hoc and sensor systems (MASS)*. [S.l.], 2019. p. 274–282. Citado na página 88.
- LIU, Ping; HAN, Shizhong; MENG, Zibo; TONG, Yan. Facial expression recognition via a boosted deep belief network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 1805–1812. Citado na página 72.
- LIU, Tongliang; YANG, Qiang; TAO, Dacheng. Understanding how feature structure transfers in transfer learning. In: *IJCAI*. [S.l.: s.n.], 2017. p. 2365–2371. Citado na página 54.
- LONG, F.; WU, T.; MOVELLAN, J. R.; BARTLETT, M. S.; LITTLEWORT, G. Learning spatiotemporal features by using independent component analysis with application to facial expression recognition. *Neurocomputing*, v. 93, p. 126–132, 9 2012. Citado 3 vezes nas páginas 81, 82 e 112.
- LOWE, David G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Springer, v. 60, p. 91–110, 2004. Citado 2 vezes nas páginas 40 e 77.
- LU, Ying; LUO, Lingkun; HUANG, Di; WANG, Yunhong; CHEN, Liming. Knowledge transfer in vision recognition: A survey. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 53, n. 2, p. 1–35, 2020. Citado 2 vezes nas páginas 53 e 80.
- LUCEY, Patrick; COHN, Jeffrey; KANADE, Takeo; SARAGIH, Jason; AMBADAR, Zara; MATTHEWS, Iain. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In: . [S.l.: s.n.], 2010. p. 94 – 101. Citado 2 vezes nas páginas 104 e 105.
- LUCEY, Patrick; COHN, Jeffrey Kenneth F.; PRKACHIN, Kenneth M.; SOLOMON, Patricia E.; MATTHEWS, Iain A. Painful data: The unbc-mcmaster shoulder pain expression archive database. *Face and Gesture*, p. 57–64, 2011. Citado 8 vezes nas páginas 71, 72, 73, 74, 75, 76, 77 e 81.
- LUTTRELL, Joseph; ZHOU, Zhaoxian; ZHANG, Yuanyuan; ZHANG, Chaoyang; GONG, Ping; YANG, Bei; LI, Runzhi. A deep transfer learning approach to fine-tuning facial recognition models. In: IEEE. *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. [S.l.], 2018. p. 2671–2676. Citado na página 54.
- LYONS, M.J.; AKAMATSU, S.; KAMACHI, M.; GYOBA, J.; BUDYNEK, J. The japanese female facial expression (jaffe) database. In *Database of digital images*, 1997. Citado na página 81.
- LYONS, Michael; AKAMATSU, Shigeru; KAMACHI, Miyuki; GYOBA, Jiro. Coding facial expressions with gabor wavelets. In: . [S.l.: s.n.], 1998. p. 200 – 205. ISBN 0-8186-8344-9. Citado 9 vezes nas páginas 71, 72, 73, 74, 75, 77, 78, 104 e 105.
- LYONS, Michael; AKAMATSU, Shigeru; KAMACHI, Miyuki; GYOBA, Jiro. Coding facial expressions with gabor wavelets. In: IEEE. *Proceedings Third IEEE international conference on automatic face and gesture recognition*. [S.l.], 1998. p. 200–205. Citado 3 vezes nas páginas 74, 75 e 78.

- LYONS, Michael J; BUDYNEK, Julien; AKAMATSU, Shigeru. Automatic classification of single facial images. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 21, n. 12, p. 1357–1362, 1999. Citado na página 77.
- MACQUEEN, James et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297. Citado 2 vezes nas páginas 84 e 85.
- MANDAL, Murari; VERMA, Monu; MATHUR, Sonakshi; VIPPARTHI, Santosh Kumar; MURALA, Subrahmanyam; KUMAR, Deveerasetty Kranthi. Regional adaptive affinitive patterns (radap) with logical operators for facial expression recognition. *IET Image Processing*, Wiley Online Library, v. 13, n. 5, p. 850–861, 2019. Citado 2 vezes nas páginas 74 e 131.
- MARINAI, Simone; FUJISAWA, Hiromichi. *Machine learning in document analysis and recognition*. [S.l.]: Springer, 2007. v. 90. Citado na página 66.
- MARKOV, Konstantin; MATSUI, Tomoko. Music genre classification using self-taught learning via sparse coding. In: IEEE. *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2012. p. 1929–1932. Citado 2 vezes nas páginas 80 e 83.
- MARKOV, Konstantin; MATSUI, Tomoko. Nonnegative matrix factorization based self-taught learning with application to music genre classification. In: IEEE. *2012 IEEE International Workshop on Machine Learning for Signal Processing*. [S.l.], 2012. p. 1–5. Citado na página 83.
- MARKOV, Konstantin; MATSUI, Tomoko. High level feature extraction for the self-taught learning algorithm. *EURASIP Journal on audio, speech, and music processing*, Springer, v. 2013, n. 1, p. 1–11, 2013. Citado 2 vezes nas páginas 52 e 54.
- MARTINEZ, Aleix; ROBERT, Benavente. *The AR Face Database*. [S.l.], 1998. Citado na página 82.
- MASCI, Jonathan; MEIER, Ueli; CIREŞAN, Dan; SCHMIDHUBER, Jürgen. Stacked convolutional auto-encoders for hierarchical feature extraction. In: SPRINGER. *International conference on artificial neural networks*. [S.l.], 2011. p. 52–59. Citado na página 60.
- MAVADATI, S Mohammad; MAHOOR, Mohammad H; BARTLETT, Kevin; TRINH, Philip; COHN, Jeffrey F. Disfa: A spontaneous facial action intensity database. *IEEE Transactions on Affective Computing*, IEEE, v. 4, n. 2, p. 151–160, 2013. Citado na página 75.
- MOHANDÉS, Mohamed; DERICHE, Mohamed; ALIYU, Salihu O. Classifiers combination techniques: A comprehensive review. *IEEE Access*, IEEE, v. 6, p. 19626–19639, 2018. Citado na página 66.
- MONEKOSSO, Dorothy; REMAGNINO, Paolo; KUNO, Yoshinori. *Intelligent Environments*. [S.l.]: Springer, 2009. Citado na página 66.

MØRUP, Morten; SCHMIDT, Mikkel N; HANSEN, Lars K. Shift invariant sparse coding of image and music data. *Submitted to Journal of Machine Learning Research*, v. 5, 2008. Citado na página 83.

MOUSAVI, Reza; EFTEKHARI, Mahdi. A new ensemble learning methodology based on hybridization of classifier ensemble selection approaches. *Applied Soft Computing*, Elsevier, v. 37, p. 652–666, 2015. Citado na página 76.

MUELLER, John Paul; MASSARON, Luca. *Machine learning for dummies*. [S.l.]: John Wiley & Sons, 2016. Citado na página 61.

MUTNEJA, Vikram; SINGH, Satvir. Modified viola–jones algorithm with gpu accelerated training and parallelized skin color filtering-based face detection. *Journal of Real-Time Image Processing*, Springer, v. 16, n. 5, p. 1573–1593, 2019. Citado na página 46.

NG, Andrew et al. Sparse autoencoder. *CS294A Lecture notes*, v. 72, n. 2011, p. 1–19, 2011. Citado na página 55.

OBAIDAT, Muath; ALEXANDROU, Alex; SANACORE, Samantha. Machine learning stacking ensemble model for predicting heart attacks. In: *The Eighth International Conference on Big Data, Small Data, Linked Data and Open Data*. [S.l.: s.n.], 2022. p. 8–14. Citado na página 68.

OJALA, Timo; PIETIKÄINEN, Matti; HARWOOD, David. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, Elsevier, v. 29, n. 1, p. 51–59, 1996. Citado 5 vezes nas páginas 40, 73, 74, 75 e 77.

OLMOS, Adriana; KINGDOM GARRISON, F. A. A. A biologically inspired algorithm for the recovery of shading and reflectance images. *Perception*, v. 33, p. 1463 – 1473, 2004. Citado na página 82.

OLSHAUSEN, Bruno A; FIELD, David J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, Nature Publishing Group, v. 381, n. 6583, p. 607–609, 1996. Citado 2 vezes nas páginas 54 e 83.

PAN, Sinno Jialin; YANG, Qiang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 10 2010. Citado 2 vezes nas páginas 51 e 52.

PANTIC, Maja; VALSTAR, Michel; RADEMAKER, Ron; MAAT, Ludo. Web-based database for facial expression analysis. In: IEEE. *2005 IEEE international conference on multimedia and Expo*. [S.l.], 2005. p. 5–pp. Citado 2 vezes nas páginas 75 e 76.

PAPAMARTZIVANOS, Dimitrios; MÁRMOL, Félix Gómez; KAMBOURAKIS, Georgios. Introducing deep learning self-adaptive misuse network intrusion detection systems. *IEEE Access*, IEEE, v. 7, p. 13546–13560, 2019. Citado na página 53.

PARK, Jooyoung; SANDBERG, Irwin W. Universal approximation using radial-basis-function networks. *Neural computation*, MIT Press, v. 3, n. 2, p. 246–257, 1991. Citado na página 87.

PEDREGOSA, Fabian; VAROQUAUX, Gaël; GRAMFORT, Alexandre; MICHEL, Vincent; THIRION, Bertrand; GRISEL, Olivier; BLONDEL, Mathieu; PRETTENHOFER, Peter; WEISS, Ron; DUBOURG, Vincent et al. Scikit-learn: Machine learning in python.

*Journal of machine learning research*, v. 12, n. Oct, p. 2825–2830, 2011. Citado na página 99.

PENG, Min; WU, Zhan; ZHANG, Zhihao; CHEN, Tong. From macro to micro expression recognition: Deep learning on small datasets using transfer learning. In: IEEE. *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*. [S.l.], 2018. p. 657–661. Citado na página 128.

PRAKASH, Jothi; NITHYA, Dr. A survey on semi-supervised learning techniques. *International Journal of Computer Trends and Technology*, v. 8, 02 2014. Citado na página 51.

PUTHANIDAM, Roshni Velluva; MOH, Teng-Sheng. A hybrid approach for facial expression recognition. In: *Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication*. [S.l.: s.n.], 2018. p. 1–8. Citado na página 45.

QIAO, Yantao; ZENG, Kai; XU, Lina; YIN, Xiaoyu. A smartphone-based driver fatigue detection using fusion of multiple real-time facial features. In: . [S.l.: s.n.], 2016. p. 230–235. Citado 2 vezes nas páginas 39 e 45.

QU, Yongzhi; HE, Miao; DEUTSCH, Jason; HE, David. Detection of pitting in gears using a deep sparse autoencoder. *Applied Sciences*, Multidisciplinary Digital Publishing Institute, v. 7, n. 5, p. 515, 2017. Citado 2 vezes nas páginas 55 e 56.

QURESHI, Aqsa Saeed; KHAN, Asifullah; SHAMIM, Nauman; DURAD, Muhammad Hanif. Intrusion detection using deep sparse auto-encoder and self-taught learning. *Neural Computing and Applications*, Springer, v. 32, p. 3135–3147, 2020. Citado 2 vezes nas páginas 80 e 87.

RAINA, Rajat; BATTLE, Alexis; LEE, Honglak; PACKER, Ben; NG, Andrew. Self-taught learning: Transfer learning from unlabeled data. *Proceedings of the Twenty-fourth International Conference on Machine Learning*, v. 227, 01 2007. Citado 6 vezes nas páginas 39, 48, 52, 79, 83 e 85.

RAINA, Rajat; MADHAVAN, Anand; NG, Andrew Y. Large-scale deep unsupervised learning using graphics processors. In: *Proceedings of the 26th annual international conference on machine learning*. [S.l.: s.n.], 2009. p. 873–880. Citado na página 52.

RAJAT, Raina. *Self-Taught Learning*. Tese (Doutorado) — Stanford University, 2019. Citado na página 52.

RAMAMURTHY, Sreenivasan Ramasamy; GHOSH, Indrajeet; GANGOPADHYAY, Aryya; GALIK, Elizabeth; ROY, Nirmalya. Star-lite: A light-weight scalable self-taught learning framework for older adults' activity recognition. *Pervasive and Mobile Computing*, Elsevier, v. 87, p. 101698, 2022. Citado 2 vezes nas páginas 80 e 86.

RANZATO, Marc'Aurelio; SUSSKIND, Joshua; MNIH, Volodymyr; HINTON, Geoffrey. On deep generative models with applications to recognition. In: IEEE. *CVPR 2011*. [S.l.], 2011. p. 2857–2864. Citado na página 72.

- RATHOUR, Navjot; SINGH, Rajesh; GEHLOT, Anita; AKRAM, Shaik Vaseem; THAKUR, Amit Kumar; KUMAR, Amit. The decadal perspective of facial emotion processing and recognition: A survey. *Displays*, Elsevier, p. 102330, 2022. Citado 2 vezes nas páginas 39 e 70.
- RENDA, Alessandro; BARSACCHI, Marco; BECHINI, Alessio; MARCELLONI, Francesco. Comparing ensemble strategies for deep learning: An application to facial expression recognition. *Expert Systems with Applications*, Elsevier, v. 136, p. 1–11, 2019. Citado 3 vezes nas páginas 78, 79 e 88.
- SCHALK, Job Van Der; HAWK, Skyler T; FISCHER, Agneta H; DOOSJE, Bertjan. Moving faces, looking places: validation of the amsterdam dynamic facial expression set (adfs). *Emotion*, American Psychological Association, v. 11, n. 4, p. 907, 2011. Citado na página 75.
- SHAO, Jie; QIAN, Yongsheng. Three convolutional neural network models for facial expression recognition in the wild. *Neurocomputing*, Elsevier, v. 355, p. 82–92, 2019. Citado 2 vezes nas páginas 73 e 128.
- SHU, Lei; MCISSAC, Kenneth; OSINSKI, Gordon; FRANCIS, Reena. Unsupervised feature learning for autonomous rock image classification. *Computers & Geosciences*, v. 106, 05 2017. Citado na página 85.
- SILL, Joseph; TAKÁCS, Gábor; MACKKEY, Lester; LIN, David. Feature-weighted linear stacking. 11 2009. Citado na página 67.
- SIM, Terence; BAKER, Simon; BSAT, Maan. The cmu pose, illumination and expression database of human faces. *Carnegie Mellon University Technical Report CMU-RI-TR-OI-02*, 2001. Citado na página 75.
- SIMEONE, Osvaldo and. A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*, v. 4, n. 4, p. 648 – 664, 2018. Citado na página 50.
- SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado na página 71.
- SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado na página 128.
- SLATE, David. *Letter Recognition*. 1991. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C5ZP40>. Citado na página 84.
- SMOLENSKY, Paul. *Information processing in dynamical systems: Foundations of harmony theory*. [S.l.], 1986. Citado 2 vezes nas páginas 54 e 80.
- SONG, Yangqiu; ROTH, Dan. Machine learning with world knowledge: The position and survey. *arXiv preprint arXiv:1705.02908*, 2017. Citado na página 80.
- SZEGEDY, Christian; LIU, Wei; JIA, Yangqing; SERMANET, Pierre; REED, Scott; ANGUELOV, Dragomir; ERHAN, Dumitru; VANHOUCKE, Vincent; RABINOVICH, Andrew. Going deeper with convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 1–9. Citado na página 128.

- SZEGEDY, Christian; VANHOUCKE, Vincent; IOFFE, Sergey; SHLENS, Jon; WOJNA, Zbigniew. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2818–2826. Citado na página 71.
- TAN, Mingxing; LE, Quoc. Efficientnet: Rethinking model scaling for convolutional neural networks. In: PMLR. *International conference on machine learning*. [S.l.], 2019. p. 6105–6114. Citado na página 128.
- TAN, Xiaoyang; TRIGGS, Bill. Fusing gabor and lbp feature sets for kernel-based face recognition. In: SPRINGER. *International workshop on analysis and modeling of faces and gestures*. [S.l.], 2007. p. 235–249. Citado na página 77.
- TANG, Yan; ZHANG, Xingming; WANG, Haoxiang. Geometric-convolutional feature fusion based on learning propagation for facial expression recognition. *IEEE Access*, PP, p. 1–1, 07 2018. Citado 2 vezes nas páginas 77 e 79.
- TAUD, Hind; MAS, JF. Multilayer perceptron (mlp). *Geomatic approaches for modeling land change scenarios*, Springer, p. 451–455, 2018. Citado na página 81.
- TAVALLAEE, Mahbod; BAGHERI, Ebrahim; LU, Wei; GHORBANI, Ali A. A detailed analysis of the kdd cup 99 data set. In: IEEE. *2009 IEEE symposium on computational intelligence for security and defense applications*. [S.l.], 2009. p. 1–6. Citado na página 87.
- TONG, Ying; CHEN, Rui; CHENG, Yong. Facial expression recognition algorithm using lgc based on horizontal and diagonal prior principle. *Optik*, Elsevier, v. 125, n. 16, p. 4186–4189, 2014. Citado na página 74.
- TURABZADEH, Saeed; MENG, Hongying; SWASH, Rafiq M; PLEVA, Matus; JUHAR, Jozef. Real-time emotional state detection from facial expression on embedded devices. In: IEEE. *2017 Seventh International Conference on Innovative Computing Technology (INTECH)*. [S.l.], 2017. p. 46–51. Citado na página 54.
- TZANETAKIS, George; COOK, Perry. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, IEEE, v. 10, n. 5, p. 293–302, 2002. Citado na página 83.
- UEDA, Naonori; SAITO, Kazumi. Single-shot detection of multiple categories of text using parametric mixture models. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2002. Citado na página 83.
- UETZ, Rafael; BEHNKE, Sven. Large-scale object recognition with cuda-accelerated hierarchical neural networks. In: IEEE. *2009 IEEE international conference on intelligent computing and intelligent systems*. [S.l.], 2009. v. 1, p. 536–541. Citado 2 vezes nas páginas 101 e 102.
- ULLAH, Zia; QI, Lin; HASAN, Asif; ASIM, Muhammad. Improved deep cnn-based two stream super resolution and hybrid deep model-based facial emotion recognition. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 116, p. 105486, 2022. Citado na página 46.
- VALSTAR, Michel; PANTIC, M. Induced disgust, happiness and surprise: An addition to the mmi facial expression database. *Proc. Int'l Conf. Language Resources and Evaluation, Workshop EMOTION*, p. 65–70, 01 2010. Citado na página 78.

- VAPNIK, Vladimir. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013. Citado 8 vezes nas páginas 61, 71, 74, 77, 81, 83, 84 e 85.
- VEHLEN, Antonia; KELLNER, Antonia; NORMANN, Claus; HEINRICHS, Markus; DOMES, Gregor. Reduced eye gaze during facial emotion recognition in chronic depression: Effects of intranasal oxytocin. *Journal of Psychiatric Research*, Elsevier, v. 159, p. 50–56, 2023. Citado 2 vezes nas páginas 39 e 45.
- VIOLA, Paul; JONES, Michael. Robust real-time face detection. *International Journal of Computer Vision*, v. 57, p. 137–154, 05 2004. Citado 4 vezes nas páginas 6, 46, 47 e 105.
- WANG, Hua; HUANG, Heng; BASCO, Monica; LOPEZ, Molly; MAKEDON, Fillia. Self-taught learning via exponential family sparse coding for cost-effective patient thought record categorization. *Personal and Ubiquitous Computing*, v. 18, p. 27–35, 2012. Citado na página 83.
- WANG, Lukun. Recognition of human activities using continuous autoencoders with wearable sensors. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 16, n. 2, p. 189, 2016. Citado 2 vezes nas páginas 55 e 56.
- WANG, Shangfei; LIU, Zhilei; LV, Siliang; LV, Yanpeng; WU, Guobing; PENG, Peng; CHEN, Fei; WANG, Xufa. A natural visible and infrared facial expression database for expression recognition and emotion inference. *IEEE Transactions on Multimedia*, IEEE, v. 12, n. 7, p. 682–691, 2010. Citado na página 76.
- WANG, Xiaogang; TANG, Xiaoou. Face photo-sketch synthesis and recognition. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 31, n. 11, p. 1955–1967, 2008. Citado na página 86.
- WEN, Guihua; HOU, Zhi; LI, Huihui; LI, Danyang; JIANG, Lijun; XUN, Eryang. Ensemble of deep neural networks with probability-based fusion for facial expression recognition. *Cognitive Computation*, Springer, v. 9, n. 5, p. 597–610, 2017. Citado na página 76.
- WEN, Guihua; LI, Huihui; LI, Danyang. An ensemble convolutional echo state networks for facial expression recognition. In: . [S.l.: s.n.], 2015. p. 873–878. Citado 2 vezes nas páginas 78 e 79.
- WESTON, Jason; CHAPELLE, Olivier; VAPNIK, Vladimir; ELISSEEFF, André; SCHÖLKOPF, Bernhard. Kernel dependency estimation. *Advances in neural information processing systems*, v. 15, 2002. Citado na página 84.
- WESTON, Jason; WATKINS, Chris. *Multi-class support vector machines*. [S.l.], 1998. Citado na página 62.
- WOLPERT, David H. Stacked generalization. *Neural networks*, Elsevier, v. 5, n. 2, p. 241–259, 1992. Citado 2 vezes nas páginas 67 e 68.
- WU, Bing-Fei; LIN, Chun-Hsien. Adaptive feature mapping for customizing deep learning based facial expression recognition model. *IEEE access*, IEEE, v. 6, p. 12451–12461, 2018. Citado 2 vezes nas páginas 75 e 131.

- XIE, Siyue; HU, Haifeng; YIN, Ziyu. Facial expression recognition using intra-class variation reduced features and manifold regularisation dictionary pair learning. *IET Computer Vision*, Wiley Online Library, v. 12, n. 4, p. 458–465, 2018. Citado na página 75.
- XU, Lei; KRZYŻAK, Adam; SUEN, Ching Y. Methods of combining multiple classifiers and their applications to handwriting recognition. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 22, n. 3, p. 418–435, 1992. Citado na página 66.
- XU, Mao; CHENG, Wei; ZHAO, Qian; MA, Li; XU, Fang. Facial expression recognition based on transfer learning from deep convolutional networks. In: IEEE. *2015 11th International Conference on Natural Computation (ICNC)*. [S.l.], 2015. p. 702–708. Citado na página 128.
- YANG, Xiangli; SONG, Zixing; KING, Irwin; XU, Zenglin. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, 2022. Citado na página 51.
- YEUNG, Michael K. A systematic review and meta-analysis of facial emotion recognition in autism spectrum disorder: The specificity of deficits and the role of task characteristics. *Neuroscience & Biobehavioral Reviews*, Elsevier, v. 133, p. 104518, 2022. Citado 2 vezes nas páginas 39 e 45.
- YIN, Lijun; WEI, Xiaozhou; SUN, Yi; WANG, Jun; ROSATO, Matthew J. A 3d facial expression database for facial behavior research. In: IEEE. *7th international conference on automatic face and gesture recognition (FGR06)*. [S.l.], 2006. p. 211–216. Citado 2 vezes nas páginas 72 e 73.
- YIN, Yilong; LIU, Lili; SUN, Xiwei. Sdumla-hmt: A multimodal biometric database. In: SPRINGER. *Biometric Recognition: 6th Chinese Conference, CCBR 2011, Beijing, China, December 3-4, 2011. Proceedings 6*. [S.l.], 2011. p. 260–268. Citado na página 86.
- ZAVASCHI, Thiago HH; JR, Alceu S Britto; OLIVEIRA, Luiz ES; KOERICH, Alessandro L. Fusion of feature sets and classifiers for facial expression recognition. *Expert Systems with Applications*, Elsevier, v. 40, n. 2, p. 646–655, 2013. Citado 6 vezes nas páginas 40, 73, 74, 75, 77 e 79.
- ZENG, Nianyin; ZHANG, Hong; SONG, Baoye; LIU, Weibo; LI, Yurong; ABDULLAH, Dobaie. Facial expression recognition via learning deep sparse autoencoders. *Neurocomputing*, 09 2017. Citado na página 73.
- ZHANG, Jing; LI, Wanqing; OGUNBONA, Philip. Transfer learning for cross-dataset recognition: A survey. 05 2017. Citado 2 vezes nas páginas 51 e 54.
- ZHANG, Jing; LI, Wanqing; OGUNBONA, Philip; XU, Dong. Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 52, n. 1, p. 1–38, 2019. Citado 3 vezes nas páginas 54, 80 e 128.
- ZHENG, Wei-Long; ZHU, Jia-Yi; LU, Bao-Liang. Identifying stable patterns over time for emotion recognition from eeg. *IEEE Transactions on Affective Computing*, IEEE, v. 10, n. 3, p. 417–429, 2017. Citado na página 71.

---

ZHU, Xiaofeng; HUANG, Zi; YANG, Yang; SHEN, Heng; XU, Changsheng; LUO, Jiebo. Self-taught dimensionality reduction on the high-dimensional small-sized data. *Pattern Recognition*, v. 46, p. 215–229, 01 2013. Citado na página [84](#).

ZUE, Victor; SENEFF, Stephanie; GLASS, James. Speech database development at mit: Timit and beyond. *Speech communication*, Elsevier, v. 9, n. 4, p. 351–356, 1990. Citado na página [83](#).