

INSTITUTO FEDERAL

Ceará

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ

IFCE *CAMPUS* FORTALEZA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELECOMUNICAÇÕES

MESTRADO ACADÊMICO EM ENGENHARIA DE TELECOMUNICAÇÕES

YVES AUGUSTO LIMA ROMERO

**UM MÉTODO GLOBAL DE VERIFICAÇÃO FACIAL UTILIZANDO REDES
CONVOLUCIONAIS EXTRATORAS DE CARACTERÍSTICAS**

FORTALEZA

2025

YVES AUGUSTO LIMA ROMERO

UM MÉTODO GLOBAL DE VERIFICAÇÃO FACIAL UTILIZANDO REDES
CONVOLUCIONAIS EXTRATORAS DE CARACTERÍSTICAS

Dissertação apresentada ao curso de Mestrado Acadêmico em Engenharia de Telecomunicações do Programa de Pós-Graduação em Engenharia de Telecomunicações do *Campus* Fortaleza do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial à obtenção do título de mestre em Processamento de Sinais. Área de Concentração: Inteligência artificial e Aprendizado de Máquinas aplicados à Engenharia.

Orientador: Prof. Dr. Ajalmar Rêgo da Rocha Neto.

FORTALEZA

2025

YVES AUGUSTO LIMA ROMERO

**UM MÉTODO GLOBAL DE VERIFICAÇÃO FACIAL UTILIZANDO REDES
CONVOLUCIONAIS EXTRATORAS DE CARACTERÍSTICAS**

Dissertação apresentada ao curso de Mestrado Acadêmico em Engenharia de Telecomunicações do Programa de Pós-Graduação em Engenharia de Telecomunicações do *Campus* Fortaleza do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial à obtenção do título de mestre em Processamento de Sinais. Área de Concentração: Inteligência artificial e Aprendizado de Máquinas aplicados à Engenharia.

Aprovado em: ____/____/____.

BANCA EXAMINADORA

Prof. Dr. Ajalmar Rêgo da Rocha Neto (Orientador)
Instituto Federal de Educação, Ciência e Tecnologia do Ceará

Prof. Dr. Ronaldo Fernandes Ramos
Instituto Federal de Educação, Ciência e Tecnologia do Ceará

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

R763m Romero, Yves Augusto Lima.

Um Método Global de Verificação Facial Utilizando Redes Convolucionais Extratoras de Características /
Yves Augusto Lima Romero. - 2025.
84 f

Dissertação (Mestrado) – Instituto Federal do Ceará, Mestrado em Engenharia de Telecomunicações, Campus
Fortaleza, 2025.

Orientação: Prof. Dr. Ajalmar Rêgo da Rocha Neto.

1. Verificação facial. 2. Reconhecimento facial. 3. Redes convolucionais. 4. Sistemas biométricos. 5.
Aprendizado de máquinas. I. Título.

CDD 621.382

A Deus.

Aos meus pais.

AGRADECIMENTOS

A Deus, por seu amor e bondade.

Aos meus pais, por acreditarem em mim.

RESUMO

Recentemente, a difusão de técnicas de Aprendizado Profundo, como as Redes Neurais Convolucionais, tem beneficiado o setor biométrico fornecendo modelos capazes de extrair características relevantes e discriminativas de imagens para tarefas de reconhecimento. No ramo do reconhecimento facial, redes convolucionais como ArcFace e FaceNet têm entregue um alto desempenho em conjunto de dados públicos, sendo capazes de produzir um espaço vetorial de características onde faces de uma mesma identidade são representadas como pontos próximos no espaço, onde se determina se duas faces pertencem à mesma identidade mediante aplicação de um limiar de distância. No presente trabalho, propõe-se dois métodos para manipular vetores de características produzidos por redes convolucionais, visando gerar classificadores que sejam aptos para efetuar a tarefa de verificação facial. Uma realiza o treinamento de um classificador para cada identidade, alcançando performance mais alta que a proposta do classificador único, apesar de ter a desvantagem de que novos classificadores precisam ser treinados quando novos indivíduos são cadastrados no sistema. A outra proposta produz um único classificador que é habilitado para realizar verificação facial em todos os indivíduos da base de dados, bem como em identidades não vistas durante a etapa de treinamento, mostrando-se uma solução escalável, de rápido treinamento e de desempenho competitivo, quando usada em associação com a FaceNet e a ArcFace; os modelos gerados por essa proposta apresentaram um baixo consumo de memória, o que os torna vantajosos para sistemas que possuam limitação de recursos.

Palavras-chave: Verificação Facial. Reconhecimento Facial. Redes Convolucionais. Sistemas Biométricos. Aprendizado de Máquinas.

ABSTRACT

Recently, deep learning techniques such as convolutional networks have improved biometric systems providing deep models. Deep convolutional models encode rich and discriminative image features for image recognition. In face recognition, convolutional networks, such as FaceNet and ArcFace, have reached high performances in public datasets. Such networks produce vector feature space where faces that belong to the same identity are represented as spatially close points. A thresholding distance rule is used to decide whether two faces belong to the same identity. The present work proposes two methods that manipulate face features extracted by convolutional networks to build classifiers that can perform face verification task. One of the proposals is a per-class strategy that trains one classifier per identity, achieving a higher performance than the global proposal; though, it suffers from the drawback that new classifiers must be trained when new identities are stored in the face recognition system. Both proposals achieved high recall rates while keeping low false positive rates in the test set when they used FaceNet and ArcFace features, showing notably better results than the thresholding rule. The other proposal produced a global and unique classifier that can be used to perform face verification on all identities in the database, and also unknown identities from other databases, not seen during the training stage, demonstrating to be a scalable solution that shows competitive performance when FaceNet and ArcFace are used as embedding sources; this proposal offers classifiers that are fast to train and show low memory consumption, making it an attractive solution for systems that need to operate in restricted-resource environments.

Keywords: Face Verification. Face Recognition. Convolutional Networks. Biometric Systems. Machine Learning.

LISTA DE FIGURAS

Figura 1 – Esquema da identificação facial.	23
Figura 2 – Esquema da verificação facial.	23
Figura 3 – Perceptron Simples	27
Figura 4 – Perceptron Multicamadas	28
Figura 5 – Aplicação dos filtros Sobel G_x e G_y	33
Figura 6 – Afunilamento da informação em rede convolucional	34
Figura 7 – Mapas de característica	34
Figura 8 – Mapas de característica de uma camada da FaceNet em resposta a um rosto humano	35
Figura 9 – Faces humanas codificadas como pontos em um espaço vetorial de características	39
Figura 10 – Refinamento do espaço de características ao longo do treinamento	40
Figura 11 – Distâncias euclidianas entre faces com valores meramente ilustrativos	41
Figura 12 – Projeções esféricas de <i>embeddings</i>	42
Figura 13 – Gráfico das fronteiras de decisão de SphereFace, CosFace e ArcFace	44
Figura 14 – Exemplo de Curva ROC	49
Figura 15 – Exemplo de Curva PR	51
Figura 16 – Representação gráfica unidimensional de distâncias entre embeddings	52
Figura 17 – Representação gráfica tridimensional de distâncias entre embeddings	53
Figura 18 – Inferência tradicional utilizando uma imagem de referência	53
Figura 19 – Inferência utilizando múltiplas imagens de referência	54
Figura 20 – Esquema da formação de um vetor de distâncias positivo	55
Figura 21 – Esquema da formação de um vetor de distâncias negativo	56
Figura 22 – Fluxo para construção de amostras de treinamento. Valores meramente ilustrativos.	58
Figura 23 – Esquema de divisão do mesmo conjunto de treinamento em 5 rodadas para busca de hiperparâmetros.	61
Figura 24 – Esquema de testes aplicável a ambas as propostas	62
Figura 25 – Curvas PR médias de estratégias de manipulação, utilizando vetores da FaceNet	69
Figura 26 – Curvas PR médias de estratégias de manipulação, utilizando vetores da ArcFace	70

Figura 27 – Curvas PR médias de estratégias de manipulação, utilizando vetores da
GhostFaceNet

LISTA DE TABELAS

Tabela 1 – Valores de hiperparâmetros predefinidos para a busca na SVM	61
Tabela 2 – Resultados com <i>Kernel</i> Linear , a um limiar de probabilidade igual a 0,5 . .	66
Tabela 3 – Resultados com <i>Kernel</i> RBF , a um limiar de probabilidade igual a 0,5 . . .	67
Tabela 4 – Resultados da Estratégia de Limiar Global	68
Tabela 5 – Resultados da Estratégia Proposta no Conjunto de Dados CFP	74

LISTA DE SIGLAS

FNR	Taxa de falsos negativos <i>False Negative Rate</i>
FPR	Taxa de falsos positivos <i>False Positive Rate</i>
AM	Aprendizado de Máquinas
RNA	Redes Neurais Artificiais
RNC	Redes Neurais Convolucionais
SVM	Máquina de Vetores Suporte (<i>Support Vector Machine</i>)
TNR	Taxa de verdadeiros negativos <i>True Negative Rate</i>
TPR	Taxa de verdadeiros positivos <i>True Positive Rate</i>

LISTA DE SÍMBOLOS

α	Alfa
δ	Delta
η	Eta
σ	Sigma
χ	Chi
θ	Ângulo (Theta)
e	Número de Neper
$\mathbb{R}^N$	Conjunto dos números reais em dimensão N
$\forall$	Quantificador universal – "para todo"
$\in$	Pertencimento a um conjunto
$\cup$	União de conjuntos
Σ	Somatório
$\ x\ $	Norma de x
$\ x\ _2^2$	Quadrado da norma euclidiana de x
$[\]$	Notação vetorial e matricial
$\cdot$	Produto escalar entre vetores
W^T	Transposição matricial
W_{ij}	Indexação matricial
$\cos$	Função cosseno
σ'	Derivada da função σ
∂	Símbolo de derivada parcial
$\hat{y}$	Inferência/previsão da variável y
$E(I)$	<i>Embedding</i> da imagem I (definido no trabalho)
$d(E(I_a), E(I_b))$	Distância entre embeddings das imagens I_a e I_b

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Sistemas Biométricos - Panorama Histórico	15
1.2	Redes Extratoras de Características de Face	17
1.3	Identificação vs Verificação	18
1.4	Objetivo	18
1.5	Organização do Trabalho	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	Tarefas de aprendizado de máquina	21
2.2	Tarefas de biometria facial	22
2.2.1	<i>Identificação vs. Verificação</i>	<i>22</i>
2.3	Aprendizado Supervisionado vs Aprendizado Não Supervisionado . . .	24
2.4	Redes Neurais Artificiais (RNA)	25
2.4.1	<i>Mudanças de Paradigma de Inteligência Artificial</i>	<i>25</i>
2.4.2	<i>Redes Neurais Clássicas</i>	<i>26</i>
2.4.2.1	<i>Equações de funcionamento</i>	<i>28</i>
2.4.2.2	<i>Retropropagação do erro e o treinamento de redes neurais</i>	<i>29</i>
2.4.3	<i>Redes Neurais Convolucionais</i>	<i>31</i>
2.4.3.1	<i>Aspectos das Redes Convolucionais</i>	<i>36</i>
2.5	Visão Histórica do Uso de Redes Convolucionais em Reconhecimento Facial	36
2.5.1	<i>A busca pela generalidade</i>	<i>37</i>
2.5.1.1	<i>A FaceNet Como Pioneira dos Modelos Embedders</i>	<i>38</i>
2.5.2	<i>Outras Iniciativas de Modelos Embedders</i>	<i>41</i>
2.6	Estudos de Aplicação de Modelos <i>Embedders</i>	45
2.7	Avaliação de Métodos para Verificação Facial	46
2.7.1	<i>Protocolos de Experimento</i>	<i>46</i>
2.7.2	<i>Métricas</i>	<i>47</i>
2.7.2.1	<i>Curva ROC</i>	<i>48</i>
2.7.2.2	<i>Curvas de Precisão e Recall</i>	<i>50</i>
3	PROPOSTAS E METODOLOGIA	52
3.1	Intuições Preliminares	52
3.2	Proposta 1	54

3.3	Proposta 2	57
3.3.1	<i>Amostras positivas do classificador global</i>	57
3.3.2	<i>Amostras negativas do classificador global</i>	57
3.3.3	<i>Treinamento e Testes</i>	58
3.4	Metodologia de testes	58
3.4.1	<i>Base de Dados</i>	59
3.4.2	<i>Redes Convolucionais Extratoras</i>	60
3.4.3	<i>Treinamento</i>	60
3.4.4	<i>Testes</i>	62
3.4.5	<i>Estratégias de Base para Comparação</i>	63
4	EXPERIMENTOS E DISCUSSÃO DE RESULTADOS	66
4.1	Resultados dos Experimentos	66
4.1.1	<i>Espaço ocupado em memória pelos modelos</i>	73
4.1.2	<i>Desempenho de métodos globais em base de dados com identidades não vistas</i>	74
5	CONCLUSÃO	76
	REFERÊNCIAS	77

1 INTRODUÇÃO

A inteligência artificial é uma área de pesquisa que se propõe a desenvolver algoritmos que simulem o funcionamento da inteligência humana, tendo vasta aplicabilidade na solução de problemas práticos. Recentemente, modelos de linguagem natural baseados na arquitetura *Transformer* (VASWANI *et al.*, 2017) têm alcançado desempenho expressivo em tarefas como aritmética, manipulação de dados tabulares e processamento de texto, obtendo grande notoriedade pública. Ademais, aplicações como reconhecimento de placas de carro e reconhecimento facial vêm se tornando cada vez mais comuns na rotina das grandes cidades. Tarefas anteriormente executadas predominantemente por operadores humanos passaram a ser parcialmente ou integralmente automatizadas por sistemas computacionais, e muitas vezes sem necessidade de supervisão constante.

O Aprendizado de Máquinas (AM) é uma subdivisão da inteligência artificial, e se concentra em produzir algoritmos que aprendem via exemplos. Diferentemente dos algoritmos de inteligência artificial mais antigos, que operaram a partir de esquemas de regras fixas, manualmente programadas, os algoritmos de aprendizado via exemplos descobrem as fronteiras de decisão que separam as classes dos exemplos fornecidos durante o processo de aprendizado (treinamento); dessa forma, para que o modelo aprenda os padrões dos dados, não é necessário o desenvolvedor programar manualmente as regras do modelo, pois o próprio algoritmo, direcionado pelos dados apresentados, efetua o aprendizado dessas regras. Uma grande vantagem desses algoritmos é sua capacidade de alcançar elevadas taxas de acerto, o que os habilita para serem aplicados em automações em vários setores da indústria. Dentre esses algoritmos, vale citar a Máquina de Vetores Suporte (*Support Vector Machine* — SVM), o K-Médias, e a Árvore de Decisão.

Historicamente, sistemas biométricos, como sistemas de reconhecimento facial, sempre se beneficiaram da aplicação conjunta de técnicas de aprendizado de máquina e visão computacional, mediante o treinamento de classificadores utilizando características de imagens codificadas por métodos de processamento de sinais. O ápice de performance de muitos sistemas biométricos foi alcançado com o advento do Aprendizado Profundo, i. e., as Redes Neurais Artificiais (RNA), que aperfeiçoaram a tecnologia de aprendizado iterativo; as Redes Neurais Convolucionais (RNC) são especializadas em extrair características relevantes de imagens, e possuem um papel de destaque entre os sistemas biométricos de face. A versatilidade das RNCs abriu espaço para uma série de arquiteturas robustas, capazes de representar padrões de imagem em diversos níveis

de abstração e posição espacial, o que favoreceu grandemente o surgimento de modelos de codificação de face como a FaceNet (SCHROFF; KALENICHENKO; PHILBIN, 2015).

Nesse contexto, o reconhecimento facial acompanhou a tendência observada em outras áreas da inteligência artificial, caracterizada pela crescente adoção de modelos baseados em aprendizado profundo.

1.1 Sistemas Biométricos - Panorama Histórico

Antes mesmo de a face humana ser utilizada para o reconhecimento biométrico, o desafio de identificar pessoas a partir da fisionomia de partes do corpo ocupava o centro de muitas discussões científicas na segunda metade do século XX. Dentre as muitas controvérsias que eclodiram nessa época, uma girou em torno das *fingerprints* — impressões digitais. Discutia-se se o desenho impresso nas mãos dos seres humanos possuía uma fisionomia única, cada pessoa tendo impressa em suas mãos uma geometria própria e singular, não havendo no mundo duas pessoas com os mesmos padrões (COHEN, 1994; COLE, 1999). Décadas depois, quando a polêmica já não estava tão em voga, e o foco dos pesquisadores se deslocou para uma atitude mais pragmática, visando antes aperfeiçoar os métodos de reconhecimento biométrico do que buscar uma solução conceitual ou filosófica para o problema da unicidade¹, diversos modelos de representação de impressões digitais foram elaborados. Isso foi um grande avanço, não só para uma melhor compreensão, por parte da comunidade científica, das impressões digitais, mas para os sistemas biométricos em geral, pois esses estudos tocam numa dimensão fundamental dos processos de reconhecimento biométrico: a representatividade. Nem tudo aquilo que faz parte de um rosto, de uma impressão digital, é relevante em termos comparativos. Para que duas entidades, e. g., duas faces humanas, sejam comparáveis entre si, é preciso, antes de tudo, que elas possuam atributos em comum. O nariz de um pode ser mais afilado que o do outro porque ambos possuem nariz. Nesse sentido, é preciso estabelecer um conjunto de características relevantes e discriminativas daquela parte do corpo humano sobre a qual o sistema biométrico vai levantar os pontos-chave de comparação, para que, ao final, seja possível computar uma pontuação de similaridade entre dois rostos.

Tendo isso em vista, os primeiros pesquisadores que se lançaram sobre as tarefas de biometria facial desenvolveram métodos para captura de padrões em faces, de modo que se pudesse razoavelmente representá-las. A eficácia desses métodos está bastante condicionada

¹ O problema da unicidade consiste em saber se as impressões digitais de duas pessoas quaisquer são sempre distintas, ou se pode haver repetição

à qualidade das características. Quanto mais representativo for o conjunto de atributos modelado, mais isso se refletirá em uma acurácia elevada. Num primeiro momento, as táticas propostas tinham um caráter holístico, e. g., EigenFaces (TURK; PENTLAND *et al.*, 1991), FisherFaces (BELHUMEUR; HESPANHA; KRIEGMAN, 1997), que abstraíam padrões da face como um todo, porém sendo vulneráveis a variações imprevisíveis da imagem, como a pose, o ângulo e a iluminação. Mais adiante, foram desenvolvidos métodos mais robustos, capazes de obter características que preservam seu valor discriminativo apesar das eventuais distorções do cenário, sendo, na medida do possível, invariantes a essas mudanças. Exemplos dessa categoria de algoritmos, ainda que nem todos tenham sido formulados com a intenção explícita de desempenhar as tarefas de reconhecimento facial, são os métodos Padrões Binários Locais (OJALA; PIETIKAINEN; MAENPAA, 2002), Wavelets de Gabor (GABOR, 1946), Características de Haar (VIOLA; JONES, 2001), bem como os extratores de pontos-chave *SIFT* (LOWE, 2004) e *SURF* (BAY; TUYTELAARS; GOOL, 2006).

Esses avanços na invariância das características começaram a ser desenvolvidos por volta dos anos 90, e representaram um avanço significativo no setor de visão computacional, atingindo níveis de melhora relevantes. Na época em que foram desenvolvidos métodos como o Padrões Binários Locais, já estava bastante difundido o uso de algoritmos de aprendizado de máquina para resolver problemas de classificação e regressão, como a SVM (BOSER; GUYON; VAPNIK, 1992) e o Gradient Boost (FRIEDMAN, 2001). Entretanto, o desempenho da máquina só veio a superar o desempenho humano nas tarefas de reconhecimento facial com o ingresso de técnicas de aprendizado profundo mediante arquiteturas de rede neural convolucional. Em 1986, o algoritmo de retropropagação que inspirou e, em grande parte, definiu o processo de treinamento das redes neurais modernas, já havia sido formalmente elaborado e publicado (RUMELHART; HINTON; WILLIAMS, 1986). Apesar disso, e de outros avanços no setor de redes neurais, o interesse acerca desse tema não estava tão em alta naquela década. Mas com o surgimento da internet, e a subsequente difusão das redes sociais e de outros serviços na rede, grandes volumes de dados ficaram disponíveis para treinar e testar arquiteturas. Isso alavancou o número de pesquisas e interessados em redes neurais, pois a partir de então houve volume de informações suficiente para novos e mais ambiciosos experimentos, já que essas redes se aperfeiçoam de forma mais refinada quando há muitos dados de exemplo para treinamento. Ao mesmo tempo, por volta dessa época, a indústria de computação deu início a grandes investimentos em hardwares, por exemplo, as GPUs, que são capazes de processar volumes massivos de dados em curto intervalo de tempo, o que também concorreu para que o tema de redes neurais voltasse para o foco das

discussões.

Em virtude do avanço das pesquisas, dentre as quais se pode destacar um trabalho que propôs uma solução para o problema do desaparecimento do gradiente, o desenvolvimento de arquiteturas de redes neurais passou por um estágio de amadurecimento, que culminou na popularização de redes neurais convolucionais profundas. O marco principal foi a publicação do trabalho da AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), uma rede neural convolucional com milhões de parâmetros, treinada em uma base de dados contendo milhões de imagens, que atingiu desempenhos superiores a diversas estratégias de reconhecimento de imagem até então conhecidas.

Essas redes convolucionais combinam técnicas de visão computacional e de redes neurais comuns, utilizando o recurso de aprendizado de parâmetros para descobrir os filtros que melhor representam características das imagens, no intuito de classificá-las. Em visão computacional, os filtros são desenvolvidos conceitualmente para extrair as características específicas desejadas pelo arquiteto, e. g., o filtro Sobel. O que as redes convolucionais fazem é aprender dinamicamente, durante o treinamento, os melhores filtros para gerar características discriminativas das imagens do banco de dados. Os filtros são inicializados com valores aleatórios, sendo ajustados conforme o algoritmo de retropropagação que efetua o ajuste dos parâmetros. O aprendizado desses filtros, bem como dos demais parâmetros, é armazenado em matrizes que podem ser exportadas para uso em outras máquinas.

1.2 Redes Extratoras de Características de Face

O grande sucesso da FaceNet (SCHROFF; KALENICHENKO; PHILBIN, 2015), uma rede convolucional montada pelo Google, popularizou o *one-shot learning*, uma técnica que aprimora o uso comum de redes convolucionais para classificação, estendendo o poder de representação da rede para amostras (rostos) que não estavam nos bancos de dados originais de treinamento e de teste. A inovação trazida pelo *one-shot learning* não é exatamente um modelo novo, mas uma metodologia, uma forma de treinar a rede convolucional, que a torna suficientemente genérica para ser utilizada, no caso do reconhecimento facial, para gerar vetores de características de boa qualidade, mesmo quando são obtidos de imagens de outros bancos de dados, ou de identidades não vistas durante o treinamento — e isso sem necessidade de um refinamento do modelo sobre dos dados novos. A função de custo proposta pelos autores é a *Triplet Loss*, e a tecnologia de treinamento desse trabalho consiste em formar triplas de imagens e

propagar ao longo da rede o erro calculado em cima da tripla — esse aspecto será detalhado mais adiante neste trabalho. O resultado final, obtido após a transmissão dos dados da face ao longo das camadas do modelo, seria um vetor de N elementos, que pode ser encarado como um ponto em um espaço de dimensão N . A este espaço se dá o nome de **espaço de características**, que é o espaço vetorial que abrange todos os possíveis padrões numéricos para vetores que representam os rostos, sendo cada um destes vetores um ponto com componentes nas N dimensões. Após um treinamento bem-sucedido do modelo, as fotos das faces de um mesmo indivíduo tendem a ser representadas como pontos próximos dentro deste espaço de características, enquanto as de indivíduos diferentes tendem a apresentar-se como pontos distantes.

O intuito dos autores da FaceNet — esta afirmação pode se estender, com maior ou menor exatidão, aos demais modelos de *one-shot learning* usados em biometria — é transformar o problema da verificação facial no problema de estabelecer um valor de distância mínimo — i. e., um limiar — entre dois vetores de características, abaixo do qual se possa afirmar acertadamente que os vetores pertencem à mesma pessoa. Outras redes convolucionais também foram desenvolvidas para este propósito, como a ArcFace (DENG *et al.*, 2018), a SphereFace (LIU *et al.*, 2017) e a GhostFaceNet (ALANSARI *et al.*, 2023).

1.3 Identificação vs Verificação

A identificação facial é a tarefa de classificar a identidade de um rosto desconhecido, carimbando-o com uma das classes do catálogo de identidades já conhecidas, o que engloba a possibilidade de carimbar como não pertencente a nenhuma delas. Já a verificação consiste em verificar se um rosto desconhecido pertence a uma identidade específica do catálogo, devolvendo uma resposta binária: pertence ou não pertence. Muitos sistemas de reconhecimento facial incluem algoritmos feitos para ambas as tarefas.

1.4 Objetivo

A despeito do grande avanço promovido pelas redes convolucionais, bem como da sua capacidade de representar características discriminativas de faces humanas, o uso de um limiar de distância como regra para determinar se duas faces pertencem à mesma identidade pode levar a uma taxa alta de falsos positivos. Em cenários assim, pode-se implementar algoritmos para descobrir um limiar ótimo, que pode ser um limiar global, aplicado a todas as identidades, ou uma lista de limiares, um para cada identidade específica da base de dados. Contudo, o uso de

algoritmos de aprendizado de máquina, como a SVM, para aprender essas fronteiras de decisão tem se mostrado bastante promissor.

Diante disso, a meta deste trabalho é propor duas estratégias de manipulação desses vetores, bem como da relação espacial existente entre eles, explorando algoritmos de aprendizado de máquina para gerar sistemas de verificação facial mais robustos, visando sobretudo manter a taxa de falsos positivos em níveis baixos, e conservar certa independência em relação à rede neural usada como fonte extratora de características. A primeira proposta, publicada recentemente (ROMERO; Rocha Neto, 2022), consiste na criação de um classificador SVM para cada identidade do conjunto de dados de treinamento, utilizando um protocolo de um contra todos (*one-vs-all*) para a verificação facial. A segunda proposta consiste na geração de um classificador SVM único, capaz de desempenhar a tarefa de verificação facial em todas as identidades de um conjunto de dados, sendo capaz, quando associado à FaceNet ou à ArcFace, de realizar essa tarefa até mesmo em identidades que não foram vistas durante o treinamento, suprimindo a necessidade de retreinamento de classificadores ou a produção de novos classificadores para identidades de cadastro recente.

As propostas e os métodos utilizados como base comparativa foram testados utilizando vetores de características das redes FaceNet, ArcFace e GhostFaceNet, sendo utilizado o valor da área da curva de Precisão-Recall como critério aferidor de desempenho. Ao final dos testes, ambas as propostas atingiram alto desempenho na tarefa de verificação facial, alcançando valores próximos de 1,0 na AUC da curva de Precisão-Recall em muitos dos testes. As propostas deste trabalho atingiram um desempenho competitivo e aceitável em comparação com os métodos da base comparativa, apresentando, contudo, uma vantagem em termos de consumo de memória e pouca demanda de processamento para efetuar o treinamento dos modelos, o que torna a solução adequada para sistemas que precisam operar em ambientes de poucos recursos.

Observa-se que a primeira proposta possui um desempenho mais alto, em virtude de gerar vários classificadores individuais que se concentram nas fronteiras de decisão específicas de cada classe; em contrapartida, essa proposta apresenta a desvantagem de que novos usuários cadastrados no sistema vão demandar o treinamento de novos classificadores. Já a segunda proposta, apesar de ter uma queda de desempenho em relação à primeira, gera um modelo único que, durante os testes realizados, não precisou ser retreinado quando novas identidades (não vistas em treinamento) foram inseridas no banco, o que confere mais conveniência e escalabilidade à solução, sobretudo quando a FaceNet e a ArcFace são utilizadas como extratoras de características.

1.5 Organização do Trabalho

No Capítulo 2, serão esclarecidos os conceitos de verificação, reconhecimento e identificação facial, bem como serão descritas de forma mais fundamentada as métricas e tecnologias que foram o ponto de partida para o desenvolvimento deste trabalho, como as redes neurais, clássicas e convolucionais, e a técnica de *one-shot learning*.

No Capítulo 3, será descrita a proposta deste trabalho, os métodos que serviram de base comparativa para os experimentos, bem como a metodologia seguida nos experimentos para gerar os resultados.

No Capítulo 4, são apresentados resultados e discussões dos experimentos realizados.

No Capítulo 5, são apresentadas as considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados temas que compõem a base teórica para uma plena compreensão das propostas desenvolvidas neste trabalho, incluindo:

- Tarefas de aprendizado de máquina: classificação e regressão
- Tarefas de biometria facial: diferenças entre identificação e verificação
- Tipos de aprendizado
- Fundamentos de Redes Neurais Artificiais e Redes Neurais Convolucionais (RNC)
- Aplicação de RNCs em tarefas de reconhecimento facial
- Iniciativas recentes que combinam modelos de AM e extração de características via RNCs
- Modos de avaliação de sistemas de verificação facial, protocolos para experimentação e critérios numéricos de aferição de desempenho (métricas)

2.1 Tarefas de aprendizado de máquina

Algoritmos de aprendizado de máquina geralmente são utilizados para realizar tarefas de classificação, ou regressão. A classificação consiste em rotular um dado de entrada com base em sua similaridade com dados de classes pré-definidas. São problemas de classificação: reconhecimento facial, reconhecimento de impressões digitais etc. O que caracteriza as tarefas de classificação é que existe um conjunto de respostas(categorias) possíveis e previsíveis que se espera que o modelo ofereça em resposta a um dado de entrada.

Já a regressão consiste na inferência/estimativa de um valor contínuo, por exemplo, o preço de uma casa. Nesse caso, já não há um conjunto de respostas previsível e pré-definido, pois o preço da casa pode assumir qualquer valor dentro dos números reais. São problemas de regressão: estimativas de preço, detecção de coordenadas de objetos em imagens, previsão de número de vendas etc.

2.2 Tarefas de biometria facial

Na área de sistemas biométricos, especialmente no nicho de reconhecimento facial, existe uma diferença conceitual e prática entre as tarefas de identificação e verificação. Essa divisão nem sempre é claramente apresentada em publicidades de produtos biométricos, pois, geralmente, o produto vendido engloba tanto a verificação como a identificação embutidas. O reconhecimento facial é o processo biométrico, tomado como um todo, que pode se dar em tempo real, ou em processamento agendado.

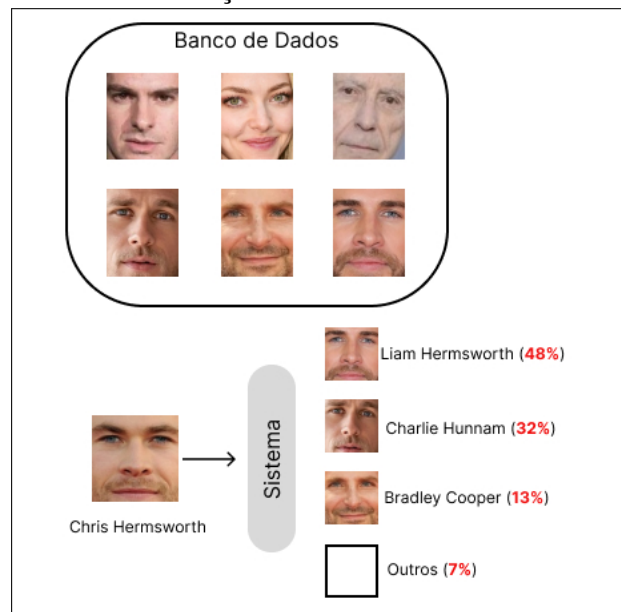
2.2.1 *Identificação vs. Verificação*

A identificação facial é o processo pelo qual se examina um rosto desconhecido e procura-se atribuir a ele uma identidade, que deve estar listada em uma base de identidades já conhecidas e previamente cadastradas. O caso de uso típico é o da empresa que cadastra a biometria de face de seus funcionários e instala catracas na entrada do prédio para que, quando alguém colocar o rosto diante da câmera, o sistema possa indicar, em meio às identidades conhecidas, a identidade mais provável daquele rosto. É uma tarefa, portanto, de classificação, e não de regressão.

A verificação facial é o processo que busca confirmar se um determinado rosto pertence, ou não, a uma identidade conhecida, previamente cadastrada, e informada no ato da verificação. Um exemplo disso é o acesso a contas bancárias via biometria facial. O usuário informa seu CPF, e o algoritmo aprova o acesso caso a face apresentada à câmera pertença à pessoa cujo CPF foi informado. Nesse caso, também se tem uma tarefa de classificação, assim como na identificação facial, com a diferença de que, na verificação, só existem duas classes, a saber, a classe positiva, que é a do titular do CPF, e a classe negativa, em que devem ser classificadas todas as outras pessoas.

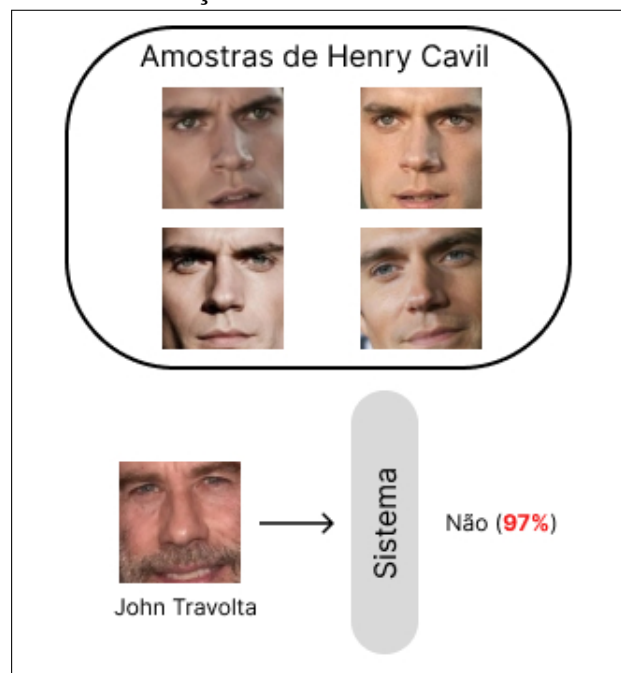
Em problemas de identificação facial onde o conjunto de dados é um **conjunto fechado**, todas as identidades que serão julgadas pelo modelo durante a fase de testes pertencem ao catálogo de identidades conhecidas. No entanto, quando o conjunto é um **conjunto aberto**, estão incluídas nele faces que não estão no catálogo (WANG; DENG, 2021). A identificação facial pode incluir, como uma etapa posterior ao ranqueamento das identidades mais prováveis, um mecanismo de rejeição de amostra, estabelecendo algum limite a partir do qual um rosto do conjunto de dados aberto será classificado como desconhecido, ou seja, não pertencente a nenhuma das classes do catálogo. Tal artifício é necessário pois, se a pessoa que está com o rosto apontado para a câmera não estiver no catálogo de identidades conhecidas, o ranqueamento vai

Figura 1 – Esquema da identificação facial.



Fonte: Elaborado pelo autor

Figura 2 – Esquema da verificação facial.



Fonte: Elaborado pelo autor

atribuir-lhe a identidade do empregado com fisionomia facial mais semelhante à sua. Caso não haja uma proteção que rejeite certos padrões, o sistema fica vulnerável a falsos positivos.

Desse modo, a verificação facial, sendo um método dedicado a desempenhar uma confirmação de identidade, pode ser aplicada pela identificação após o processo de ranqueamento, para conferir se a identidade do rosto analisado realmente coincide com algum dos padrões do catálogo, ou não. Nas Figuras 1 e 2, estão apresentados os esquemas de funcionamento desses dois processos, ilustrando a ênfase que se dá ao processo biométrico em cada uma dessas tarefas. Na Figura 1, ilustra-se a identificação facial em um caso onde o rosto que está sendo analisado não se encontra no catálogo de identidades conhecidas. Neste exemplo, o indivíduo do catálogo mais semelhante ao ator Chris Hemsworth é seu irmão, Liam Hemsworth, que foi indicado com 48% de probabilidade. Se o sistema utilizar um limiar de 50% para rejeitar as amostras, o rosto será considerado desconhecido, pois nenhum dos candidatos supera o valor de 50%. Já na Figura 2, é apresentada ao sistema um rosto, e pergunta-se se o rosto pertence ao ator Henry Cavil. Como o rosto analisado nessa ocasião é do John Travolta, então o sistema dá uma resposta negativa com 97% de probabilidade. Portanto, a finalidade da verificação facial é descobrir se duas imagens são da mesma pessoa.

2.3 Aprendizado Supervisionado vs Aprendizado Não Supervisionado

Antes de aprofundar os algoritmos mais comumente aplicados em sistemas de biometria facial, cumpre esclarecer dois conceitos da área de aprendizado de máquina: aprendizado supervisionado e o não supervisionado.

Os algoritmos de aprendizado de máquina podem ser subdivididos em supervisionados e não supervisionados. Em ambos os casos, o algoritmo aprende via exemplos, sendo-lhe fornecida alguma informação para direcionar o aprendizado. No caso dos algoritmos de aprendizado supervisionado, a informação que direciona o aprendizado é o rótulo, que consiste na resposta correta que se espera que o modelo infira com base nos dados de entrada; nesse tipo de aprendizado, o rótulo é um referencial para que o algoritmo saiba se está acertando ou errando, permitindo que ele aprenda a mapear as entradas nas saídas desejadas, enquanto busca maximizar/minimizar sua função objetivo. Dentre os algoritmos pertencentes à classe dos supervisionados, os mais comumente citados são a SVM e as RNA, incluindo as recorrentes (RNAR) e as convolucionais (RNC). As RNA, em particular, utilizam o rótulo para realizar um processo de autocorreção de parâmetros, que se dá iterativamente, entre erros e acertos.

Já o aprendizado não supervisionado é direcionado por alguma função de similaridade que sirva de critério para agrupar amostras semelhantes e discriminar as que são muito diferentes. Nesse aprendizado não se utiliza rótulos, pois o algoritmo converge à medida que se aproxima de um mínimo/máximo da sua função objetivo, que pode ser explícita ou implicitamente definida. No aprendizado não supervisionado, a função objetivo normalmente busca minimizar as diferenças dentro de grupos e maximizar as diferenças entre grupos. Exemplos de algoritmos que utilizam esse aprendizado são o K-Médias e o PCA.

As redes neurais, que compõem a base deste trabalho, são algoritmos de aprendizado supervisionado iterativo que aprendem padrões em vários níveis de abstração, convergindo por meio de uma descida no gradiente da função objetivo, como será esclarecido mais adiante.

2.4 Redes Neurais Artificiais (RNA)

A estratégia para verificação facial proposta neste trabalho toma como ponto de partida modelos de redes neurais convolucionais como a FaceNet e a ArcFace, e sua capacidade para extrair características de rostos humanos de forma eficiente para o reconhecimento. No intuito de fundamentar o funcionamento dessas redes, é preciso, antes, estabelecer bem os conceitos que norteiam as redes neurais clássicas (RNA), que servirão de base para compreender o funcionamento das redes convolucionais.

2.4.1 Mudanças de Paradigma de Inteligência Artificial

Os métodos mais antigos de inteligência artificial eram, muitas vezes, dependentes de regras e modelagens de conhecimento manualmente definidas por especialistas. O recurso de cláusulas condicionais, bastante empregado por sistemas especialistas (BUCHANAN; SMITH, 1988), que fazia o programa navegar pelas diversas possibilidades de resposta, é um exemplo expressivo. Os métodos do ramo de Engenharia do Conhecimento tomavam nota do conhecimento técnico de especialistas de uma determinada área, elaborando representações e codificações desse conhecimento sob a forma de regras formais, com o intento de construir raciocínios a partir delas. Apesar de todos os resultados alcançados por tais técnicas, essa forma de abordar os problemas do mundo real carecia de flexibilidade e autonomia. Pouco a pouco essa classe de abordagem foi perdendo o atrativo em face de algoritmos marcados pelo paradigma do aprendizado via exemplos, grupo em que estão enquadrados os algoritmos clássicos de aprendizado de máquina, bem como as redes neurais.

Um dos traços fundamentais desses algoritmos é a capacidade de equacionar os problemas do mundo real em uma busca pelo conjunto ótimo de parâmetros internos que maximizam o acerto. Essa descoberta se dá iterativamente. No princípio do processo de descoberta, denominado de treinamento, o modelo erra bastante; contudo, esses erros são corrigidos pelo algoritmo de treinamento, para forçar o modelo a se ajustar e obter respostas mais acertadas. Um exemplo disso é a Regressão Linear iterativa, que busca, de erro em erro, o valor correto dos coeficientes angular e linear visando obter a reta em relação à qual a soma das distâncias quadráticas de um conjunto de pontos é a menor possível. Esse tipo de regressão tem a particularidade de que seus parâmetros podem ser descobertos mediante uma fórmula analítica fechada, não havendo necessidade de iteração. Esse fato não sucede à maioria dos algoritmos de aprendizado via exemplos, sobretudo as redes neurais, que podem possuir milhões de parâmetros internos, sendo inviável descobrir uma fórmula analítica que entregue prontamente a resposta ótima e exata de cada parâmetro, para um determinado conjunto de pontos usado para aprendizado.

2.4.2 *Redes Neurais Clássicas*

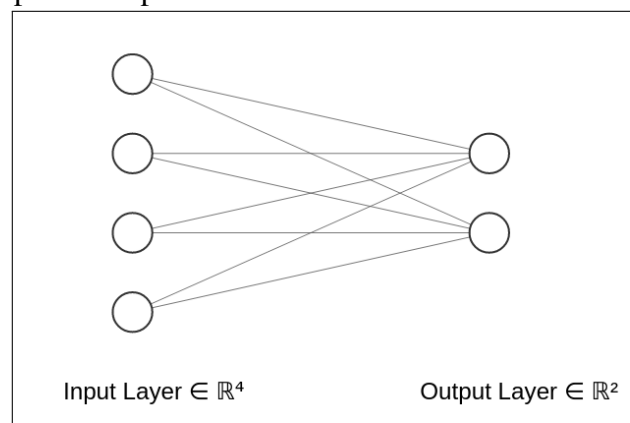
No final da década 1950, foi publicado o trabalho que definiu a primeira rede neural artificial, o Perceptron (ROSENBLATT, 1958), reconhecido como o ponto de partida para todos os avanços subsequentes. O Perceptron é um conjunto de neurônios em associação a um valor mutável de viés, juntamente com uma função de ativação que estabelece um critério de resposta, sendo cada neurônio vinculado a um parâmetro próprio (o peso do neurônio), e é responsável por representar o valor relativo de cada dimensão espacial do conjunto de pontos utilizado para o aprendizado. O Perceptron é treinado por um algoritmo que, de erro em erro, efetua a correção desses pesos em proporção ao impacto que cada dimensão da amostra surte na resposta final. Quando o modelo erra uma amostra, atribuindo-lhe uma classe à qual ela não pertence, a responsabilidade pelo erro é repartida de forma proporcionalizada, e o que o algoritmo de treinamento faz é corrigir mais severamente aqueles neurônios que tiveram maior parcela de responsabilidade pelo erro. Ao final, o que se espera obter é um conjunto de pesos, ou parâmetros, que minimizem a função de custo, que pode ser uma função de soma quadrática ou outras funções mais complexas.

Anos depois, foi publicado um trabalho de análise do Perceptron (MINSKY; PAPERT, 2017), destacando suas capacidades e limitações. Nesse trabalho, ficou provado que o Perceptron possui uma limitação intrínseca que o incapacita para resolver problemas com classes não

linearmente separáveis, como o problema do XOR, um caso geometricamente expressivo desta demonstração. Um dos efeitos, provavelmente não premeditados, desse trabalho, foi um crescente desinteresse, por parte de muitos pesquisadores, no ramo de redes neurais, pois ficara provado que o Perceptron não podia ir além das condições limitantes descritas por Minsky, que só muito raramente aparecem nos problemas do mundo real.

Na segunda metade da década de 1980, um trabalho acadêmico definiu o algoritmo de retropropagação utilizado nas redes neurais modernas (RUMELHART; HINTON; WILLIAMS, 1986). Esse algoritmo estabelece o roteiro de treinamento do Perceptron Multicamadas, que é um modelo de rede neural mais avançado que o Perceptron simples, sendo capaz de superar aquelas limitações apontadas por Minsky. Esse modelo mais complexo dá origem à ideia de interconexão entre neurônios, possibilitando a representação de padrões complexos em diversos níveis de abstração, gerando características mais ricas e mais apropriadas para solucionar problemas do mundo real. Se o Perceptron simples modelava o universo de dados como um conjunto de classes linearmente separáveis, expostas ao mundo externo por uma única camada de entrada dotada de pesos, o Perceptron Multicamadas realiza combinações lineares de várias camadas de neurônios para produzir figuras geométricas que ultrapassam o limite da linearidade. Uma alegoria bastante usual para explicar este comportamento é dizer que o PMC calcula e une um grupo de retas para formar uma figura que se assemelha à geometria dos dados.

Figura 3 – Perceptron Simples

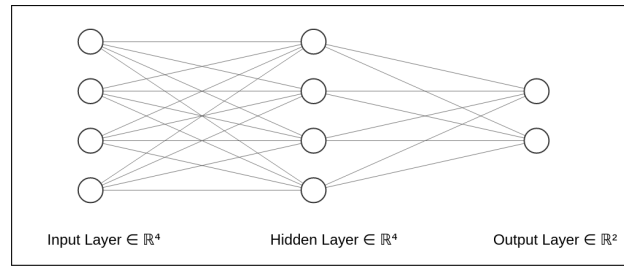


Fonte: Elaborado pelo autor

Na Figura 3, pode-se notar que há 4 neurônios na camada de entrada, que é uma camada única. Cada neurônio representa um atributo do conjunto de dados. No caso dessa figura, esses atributos poderiam ser altura, idade, peso e sexo, e a variável que se deseja prever é a existência, ou não, de doença cardíaca. O Perceptron vai multiplicar o valor de cada atributo pelo peso do neurônio que lhe corresponde; ao final, esses produtos serão somados e a função de ativação do

neurônio de saída determinará o resultado da previsão.

Figura 4 – Perceptron Multicamadas



Fonte: Elaborado pelo autor

Já na Figura 4, cada um dos 4 neurônios da camada de entrada estão conectados a outros 4 neurônios de uma camada intermediária, que é conhecida pelo nome camada escondida ou, como está redigido na figura, *hidden layer*. Cada neurônio da camada escondida armazena o resultado do produto escalar entre o vetor de entrada e o vetor de pesos que liga a camada anterior a este neurônio. Isso significa que, no Perceptron Multicamadas (PMC), a estrutura que interliga a camada de entrada com a camada escondida é uma matriz de pesos, o que também vale para a conexão entre as diversas camadas escondidas que possa haver no modelo projetado. Esse estilo de camada é chamada de *fully-connected*, ou seja, são camadas totalmente conectadas; é comum também chamar essas camadas de densas (*dense layer*), pois o volume de conexões tende a ser grande quando se adiciona muitos neurônios na rede. Essas camadas densas, quando combinadas em sequência, geram um modelo capaz de representar padrões de dados em diversos níveis de abstração, que se tornam mais e mais representativos e eficientes para tarefas do mundo real quanto mais essas matrizes de peso forem refinadas pelo algoritmo de treinamento.

2.4.2.1 Equações de funcionamento

No caso do Perceptron Simples, a equação de funcionamento, ou seja, a fórmula analítica que define o processamento dos dados de entrada até chegar à resposta do neurônio de saída, está descrita na Equação 2.1.

$$\hat{y} = \sigma \left(\sum_{j=1}^n W_j X_j + b \right) \quad (2.1)$$

$$I_1^{(1)} = \sigma_1 \left(\sum_{j=1}^m W_{ij}^{(1)} X_j + b_1^{(1)} \right) \quad (2.2)$$

$$I_N = \sigma_N \left(\sum_{i=1}^m W_{ij}^{(N)} I_{N-1} + b_i^{(N)} \right) \quad (2.3)$$

Essa equação mostra que basta efetuar um produto escalar entre o vetor de entrada X , e o vetor de pesos W , somar o resultado a um valor de viés b , e aplicar a função de ativação σ , para obter a saída do último neurônio. Já a equação de funcionamento do Perceptron Multicamadas é mais complexa; a equação que determina a trajetória dos dados, desde a camada de entrada para a primeira camada escondida, pode ser expressa nos termos da Equação 2.2. E a equação pela qual os estados de última camada escondida são transmitidos para a camada seguinte estão definidos na Equação 2.3. A passagem de uma camada a outra se dá mediante multiplicação de matrizes, multiplicando sempre a matriz de pesos W_{ij}^k da camada k pela saída $I_{(k-1)}$ da camada anterior, somando este produto com o vetor de viés; ao final, aplica-se a função σ_k da referida camada, que pode ser uma função linear, ou não, a depender da escolha do arquiteto da rede.

2.4.2.2 Retropropagação do erro e o treinamento de redes neurais

A princípio, tanto os valores do vetor de pesos do Perceptron simples, quanto os das matrizes de pesos das camadas do Perceptron Multicamadas são inicializados de forma aleatória. À medida que a rede faz inferências, visando acertar a classe, ou o valor, das amostras, o resultado da camada de saída é comparado com o valor real, ou esperado, da amostra, que funciona como uma espécie de gabarito (*ground truth*). Quando a inferência não bate com o resultado esperado para aquela amostra, é preciso corrigir os pesos do modelo de uma forma que as próximas inferências se aproximem mais do valor esperado. Contudo, antes de efetuar o ajuste, é preciso identificar os neurônios que mais influíram na resposta incorreta da rede e corrigir o valor de seus pesos em proporção ao impacto que surtiram na inferência final. O grande problema é que qualquer inferência da rede neural densa utiliza todas as matrizes de peso para gerar a resposta. Isso quer dizer que, quando um erro acontece, a maioria dos neurônios tem alguma parcela de responsabilidade, exceto aqueles cujo valor de peso é igual a 0, o que é raro. Para descobrir essa proporção entre o erro ocorrido na inferência, e a participação de cada neurônio nesse processo, foi proposto, em 1986, o *backpropagation* (retropropagação) (RUMELHART; HINTON; WILLIAMS, 1986), que é um algoritmo que computa as derivadas parciais da função de custo em relação ao valor atual de cada peso da rede. Para isso, é preciso calcular o gradiente da expressão final da inferência em relação ao vetor de pesos, no caso do Adaline (uma arquitetura

semelhante ao Perceptron), e em relação às matrizes de peso que interligam cada camada, no caso do Perceptron Multicamadas.

$$J = \sum_{k=1}^K \frac{1}{2} (y_k - \hat{y}_k)^2 \quad (2.4)$$

A função de custo definida na Equação 2.4 estabelece que, quanto menor a diferença entre os valores inferidos pela rede e os valores reais das amostras, menor é o erro médio, significando, assim, que os neurônios serão mais severamente corrigidos quando essa discrepância for alta. A correção dos pesos de uma rede neural, como o Perceptron e o Perceptron Multicamadas, é feita incrementalmente, ou seja, pela adição, ou subtração, de pequenos valores ao valor atual desses pesos. O Adaline, uma rede semelhante ao Perceptron e o Perceptron Multicamadas, como não possui camadas intermediárias, a equação de propagação é simplificada, e está expressa nas Equações 2.5 e 2.6.

$$W_j \leftarrow W_j - \eta \cdot \frac{\partial J}{\partial W_j} \cdot X_j \quad (2.5)$$

$$\frac{\partial J}{\partial W_j} = (y - \hat{y}) \cdot X_j \quad (2.6)$$

Nessa fórmula, nota-se que, para atualizar o valor do peso W_j , basta somar o valor anterior deste peso ao valor de $\frac{\partial J}{\partial W_j}$, que é o gradiente da função de custo da Equação 2.4 em relação a este peso vezes o j -ésimo atributo da amostra, multiplicado pela taxa de aprendizado η , que é um número menor do que 1, inserido na equação para evitar que os pesos sejam atualizados muito depressa. Com a atualização iterativa desse vetor, o modelo vai convergindo para uma solução mais próxima do conjunto de parâmetros ótimos, e os valores do gradiente vão se tornando cada vez mais insignificantes. No começo do treinamento, o vetor de pesos sofre mudanças mais bruscas; com o passar das iterações, esse vetor vai se estabilizando até adquirir a forma final que se cristaliza com o término do treinamento.

$$W_{ij}^{(L)} \leftarrow W_{ij}^{(L)} - \eta \cdot \frac{\partial J}{\partial W_{ij}^{(L)}} \quad (2.7)$$

$$\frac{\partial J}{\partial W_{ij}^{(L)}} = \delta_i^{(L)} I_j^{(L-1)} \quad (2.8)$$

$$\delta_i^{(L)} = \frac{\partial J}{\partial I_j^{(L)}} \sigma'(Z_i) \quad (2.9)$$

Já a equação de propagação do Perceptron Multicamadas possui uma característica recursiva. Como as saídas $I^{(L)}$ de uma determinada camada L são expressas analiticamente como dependentes das variáveis das outras camadas ($I^{(L)}$ depende de $W^{(L-1)}, I^{(L-1)}, W^{(L-2)}, I^{(L-2)}, \dots, W^{(1)}$ e X), descobrir a expressão do gradiente dos pesos $W^{(L)}$ dessa camada é uma tarefa que requer o cômputo de um extenso produto de derivadas parciais. Esse cômputo apresenta um padrão recorrente que está definido nas Equações 2.7, 2.8 e 2.9. Por meio dessa equação, é possível calcular o impacto que cada neurônio surtiu na resposta final, ainda que alguns desses neurônios estejam associados a camadas que, por estarem distantes da camada de saída, aparentam influir só remotamente no cálculo da inferência. Muitos fatores podem prejudicar ou aperfeiçoar esse processo de correção, como a escolha da função de custo, das funções de ativação, bem como o desenho da arquitetura da rede em si, que envolve a escolha do número de camadas e do número de neurônios de cada camada.

2.4.3 *Redes Neurais Convolucionais*

Em métodos de redes neurais clássicas, como o Perceptron Multicamadas, as amostras de treinamento já são esperadas em forma de vetor, em que cada elemento representa um atributo, e.g., altura, idade, peso, largura, estado civil etc. Em problemas de visão computacional, como reconhecimento de objetos em fotos e detecção de pessoas, é bastante complexo estabelecer um vetor que represente atributos de forma unívoca, pois a informação está espalhada ao longo da imagem, e um mesmo dado relevante pode aparecer em tamanhos, escalas, ângulos e regiões diferentes. A proposta das redes neurais convolucionais é reduzir a necessidade de engenharia manual de atributos pelo projetista, terceirizando essa etapa de extração de características e informações relevantes para a própria rede neural, orientada pelo algoritmo de treinamento. Ao final, a rede produzirá um vetor de características. Entretanto, a posição dos elementos desse vetor não carrega um significado inequívoco; ou seja, não se pode afirmar que cada índice representa um tipo de informação presente na imagem, por exemplo, que primeiro elemento representa o raio de curvatura médio dos contornos, enquanto o último representa o matiz médio das cores. Esse vetor final é bastante discriminativo, podendo servir para classificar objetos, porém pode carecer de interpretabilidade, sendo difícil para o projetista compreender

exatamente o que aquelas características representam no modelo. Diversos esforços foram feitos recentemente para mitigar o problema da pouca interpretabilidade dos mapas de característica das redes convolucionais (ZHANG; WU; ZHU, 2018; ZEILER; FERGUS, 2014). Isso, no entanto, parece ser um obstáculo bastante comum entre métodos de aprendizado profundo.

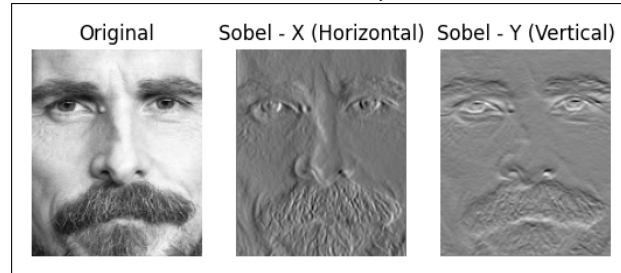
A proposta das redes convolucionais engajou uma série de esforços acadêmicos para resolver problemas de imagem propondo arquiteturas cada vez mais inovadoras e eficientes, lideradas pela pioneira AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), que deu pontapé inicial demonstrando um bom desempenho no problema da ImageNet, considerado, à época, um problema bastante desafiador. Em um espaço de poucos anos, muitas arquiteturas e metodologias de treinamento envolvendo redes convolucionais foram propostas para múltiplas classes de problema; algumas das mais conhecidas são a VGGNet (SIMONYAN; ZISSERMAN, 2014), a GoogLeNet (SZEGEDY *et al.*, 2015) e uma das mais relevantes: a ResNet (HE *et al.*, 2016), que foi responsável pela idealização da conexão residual, uma estrutura básica de conexão de redes convolucionais que basicamente adiciona à saída de algumas camadas o valor bruto do sinal de entrada, funcionando como um reforço de sinal, possibilitando que os arquitetos aumentem significativamente a profundidade das redes convolucionais sem que isso resulte em maiores perdas de informação ao longo da rede.

As redes convolucionais constituem um tipo muito particular de rede neural, dedicadas ao processamento de informações pluridimensionais, como imagens. A convolução, operação matemática que serviu de inspiração para essas arquiteturas, consiste, em termos computacionais, no janelamento completo dos dados, realizando o produto de um filtro com cada janela para extrair dos dados características relevantes, e desempenhar, em tempo reduzido, aquilo que, no modelo *Fully Connected*, é obtido a custo de muitas multiplicações de matrizes. A convolução é, na visão computacional, uma das operações que mais se aproxima da extração semi-automática de atributos. O que as redes neurais convolucionais fazem é tornar esse processo de extração totalmente automático e livre de intervenção humana.

Na área de visão computacional, os filtros são empregados sobre imagens a fim de obter informações de contorno em múltiplas direções, bem como segmentar regiões de acordo com sua heterogeneidade ou homogeneidade (o que, no domínio de Fourier, é conhecido como alta e baixa frequência). Esses filtros, no entanto, são projetados pelo cientista utilizando seu conhecimento prévio de processamento de imagens.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (2.10)$$

Figura 5 – Aplicação dos filtros Sobel G_x e G_y



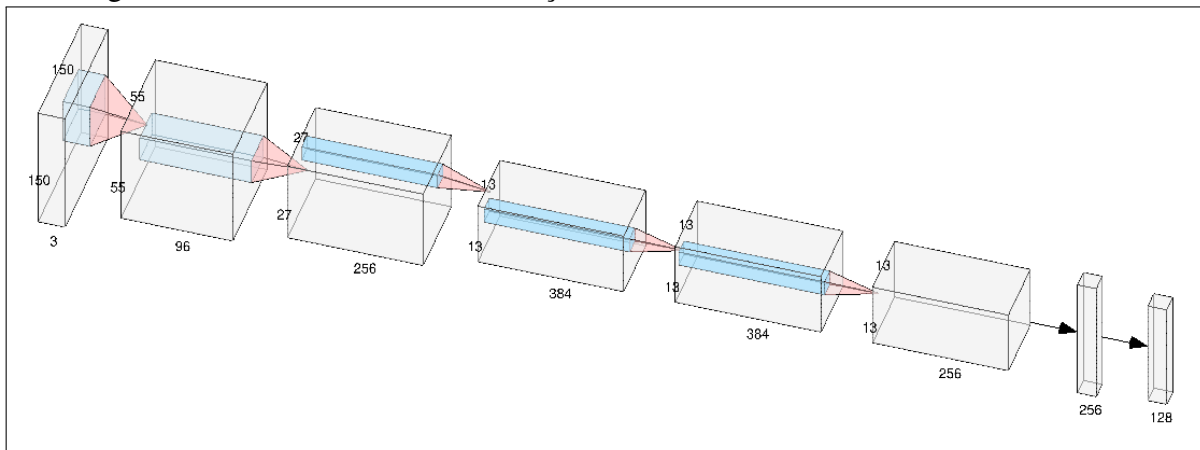
Fonte: Elaborado pelo autor

Exemplos de filtros são os filtros de Sobel, G_x e G_y (na Equação 2.10), que destacam na imagem respectivamente as variações nos eixos X e Y ; na Figura 5, estão apresentadas três imagens, a primeira é o rosto do ator Christian Bale, e as outras duas são o resultado da aplicação dos filtros G_x e G_y para marcar as variações direcionais dessa imagem.

Uma rede convolucional, semelhantemente, utiliza filtros para extrair informações de imagens. Em cada camada há uma grande quantidade de filtros, cada um especializado em extrair um tipo de informação. Um determinado filtro, quando aplicado à imagem pela convolução, produz uma espécie de máscara da imagem, onde são destacados os aspectos da informação que aquele filtro está concentrado em representar. Essas máscaras são comumente denominadas mapas de características (*feature maps*). A partir da primeira camada escondida convolucional, as convoluções são aplicadas sobre os mapas de características gerados na camada anterior, representando a informação da imagem em diversos níveis de abstração e qualidade. À medida que essas informações são transmitidas pela rede, também acontece um processo gradual de compactação e codificação, pelo qual se diminui a dimensionalidade dos dados. Por exemplo, a camada de entrada de uma rede convolucional espera receber imagens RGB com grandes dimensões, e. g. $150 \times 150 \times 3$, e entregar como resultado um vetor de características de tamanho reduzido, e. g. 256×1 . Esse vetor de características é acoplado a uma rede neural totalmente conectada, de camadas densas, que vai usar essas informações mais refinadas obtidas na convolução para efetuar eficientemente a classificação das imagens.

Uma ilustração dessa extração de características, bem como redução de dimensionalidade, está na Figura 6, onde se observa que ocorre um afunilamento da informação ao longo da rede,

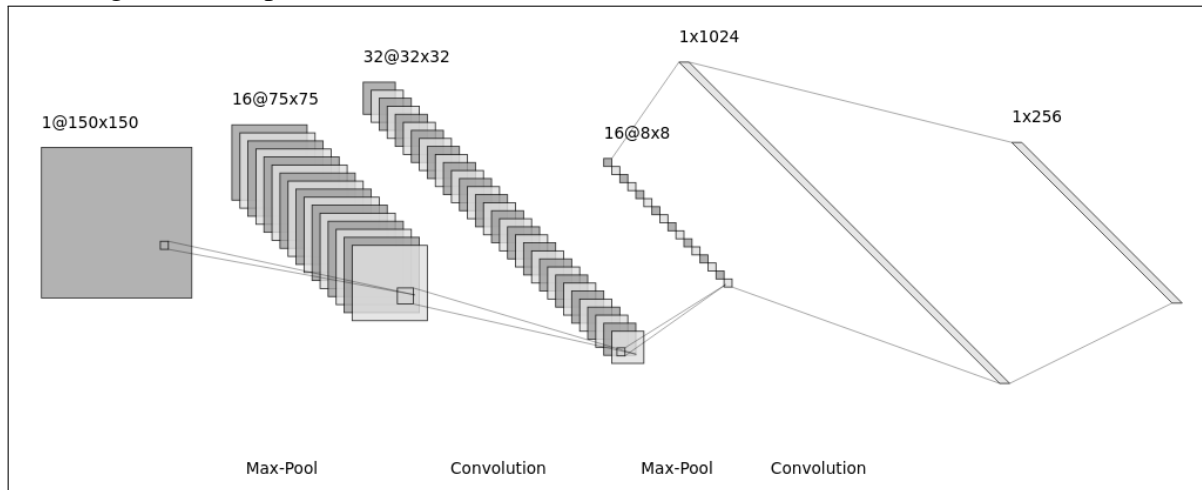
Figura 6 – Afunilamento da informação em rede convolucional



Fonte: Elaborado pelo autor

sugerindo que, com o processamento do sinal original de camada em camada, a informação que é retida pelo modelo é cada vez mais seletiva. Além da convolução, a rede possui outros mecanismos para reduzir a dimensionalidade dos dados, como o Pooling, que faz a varredura da imagem transformando janelas de pixels em um valor único, que pode ser a média da janela, o valor máximo etc., de modo a fazer o processamento focar nas partes mais relevantes.

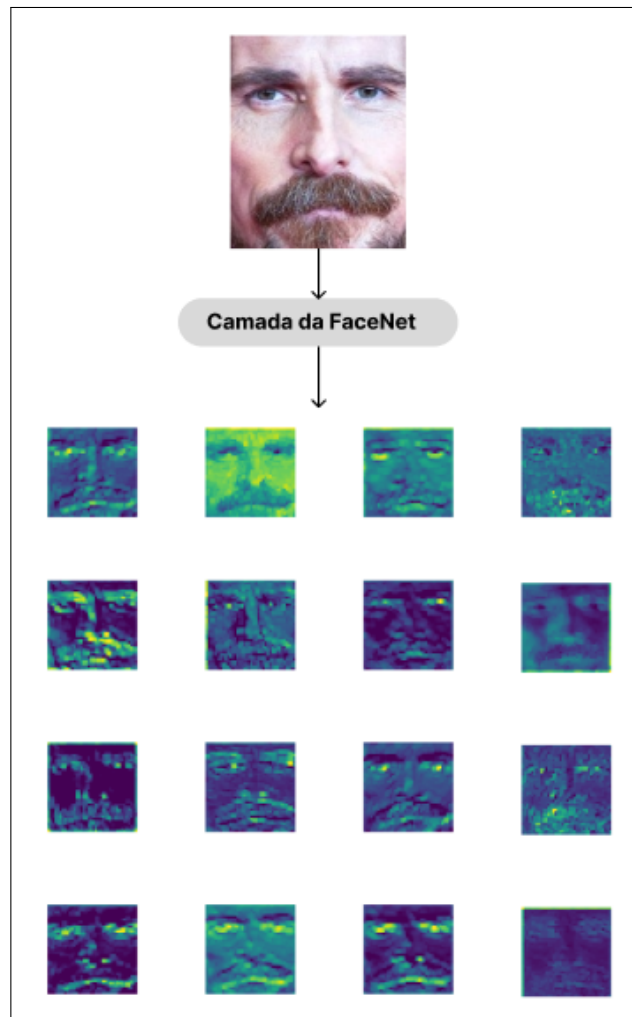
Figura 7 – Mapas de característica



Fonte: Elaborado pelo autor

Na Figura 7, está apresentado, sob outro ponto de vista, o que acontece na rede neural convolucional. Nela está explicitada a ideia de que, para uma imagem, são produzidos vários mapas de características, os quais são processados pela camada seguinte, gerando outros mapas de dimensão menor e com características mais ricas, até chegar à etapa de vetorização. Isso pode ser visto como um processo de codificação dos dados, mediante o qual o dado vai sendo refinado e alcançando representações mais concisas que concentram a informação relevante em estruturas cada vez menores.

Figura 8 – Mapas de característica de uma camada da FaceNet em resposta a um rosto humano



Fonte: Elaborado pelo autor

Nas tarefas de reconhecimento facial, utiliza-se características extraídas por essas camadas, geralmente as que ficam mais próximas do final da rede. Na Figura 8, mostra-se o resultado de uma das camadas intermediárias da FaceNet em resposta a um recorte do rosto do ator Christian Bale. A camada em questão possui 64 filtros e, assim, dá origem a 64 mapas de características, dos quais apenas 16 foram escolhidos para compor a ilustração. Observe-se que cada filtro marca com cores mais vivas certos padrões encontrados no rosto, e diminui a luminosidade daqueles pixels que não fazem parte de sua região de interesse. Tudo isso é aprendido ao longo do treinamento. Após a parte convolucional da arquitetura, acopla-se uma ou mais camadas densas, que recebem as informações sintetizadas pelas camadas de convolução e as utilizam como se fossem o vetor de entrada de um Perceptron Multicamadas.

2.4.3.1 Aspectos das Redes Convolucionais

As redes convolucionais apresentam três características que convém discutir brevemente (GOODFELLOW; BENGIO; COURVILLE, 2016), a saber, (1) Interação Esparsa, (2) Compartilhamento de Parâmetros e (3) Representatividade Equivariante.

A **interação esparsa** se refere ao fato de que, apesar de as imagens recebidas pela rede convolucional possuírem grandes dimensões e, portanto, representarem um grande volume de dados, ainda assim a rede consegue extrair dessas imagens características significantes utilizando filtros que ocupam uma área de poucos pixels, por exemplo: 3×3 ou 7×7 . Isso reduz bastante o número de parâmetros que precisam ser aprendidos pelo modelo, já que o número de pesos de um filtro é igual à quantidade de elementos na matriz que o define.

O **compartilhamento de parâmetros** diz respeito à reutilização dos pesos em várias partes da imagem. Nas redes neurais totalmente conectadas, como o Perceptron Multicamadas, cada neurônio só é utilizado uma vez para gerar a saída da rede. Já na rede convolucional, todos os poucos pesos dos filtros são usados em todas as partes do dado de entrada. Ou seja, ao invés de aprender um parâmetro para cada região da imagem, aprende-se um pequeno grupo de parâmetros para obter informações significativas da imagem inteira.

Outra característica importante é a da **equivariância das representações**, que é uma propriedade que deriva da própria natureza da convolução. Considere que uma imagem $I(x, y)$ é deslocada em um pixel para a direita ($I(x, y) \leftarrow I(x - 1, y)$). O resultado da convolução $G(I(x - 1, y))$ é igual ao próprio $G(I)$ deslocado em um pixel para a direita. Isso significa que a convolução possui um grande potencial para representar padrões que aparecem em diferentes regiões da imagem, por exemplo, contornos. Essa propriedade, no entanto, não se aplica às variações de escala e rotação.

2.5 Visão Histórica do Uso de Redes Convolucionais em Reconhecimento Facial

Em 2012, foi publicado um trabalho que utiliza redes convolucionais para aperfeiçoar o processo de verificação facial em indivíduos humanos (HUANG; LEE; LEARNED-MILLER, 2012). Esse trabalho apresenta um breve panorama do estado da arte, àquela altura, dos sistemas de reconhecimento facial. A maioria, segundo os autores, dependia de características *hand-crafted*, ou seja, projetadas com base em conhecimento humano prévio, dentre as quais se pode destacar a técnica de extração Padrões Binários Locais. O trabalho se concentra em combinar esses descritores de características tradicionais com o poder de extração das redes

convolucionais, para explorar a representação dos dados em diversos níveis, atingindo, assim, um alto desempenho em ambientes que não são controlados. Esse trabalho evidencia o escopo predominante da investigação na área naquele período. Com o passar do tempo, a preocupação começou a entrar no campo da generalidade e escalabilidade dos modelos de rede convolucional para as tarefas biométricas. Já não era mais suficiente propor técnicas que empregam redes convolucionais para obter bons resultados em um banco de dados específico. O foco de alguns estudiosos passou a ser a construção de modelos que, embora treinados em um banco de dados particular, alcançassem tal poder de generalização que pudessem ser utilizados em outros conjuntos de dados de rosto humano, com outras distribuições de sexo, idade e etnia.

2.5.1 A busca pela generalidade

Essa preocupação se manifestou singularmente na proposição das redes neurais DeepFace (TAIGMAN *et al.*, 2014), resultado de uma pesquisa envolvendo o Facebook e a Universidade de Tel Aviv, e DeepID (SUN; WANG; TANG, 2014), bem como na crescente apresentação de novos conjuntos de dados, mais volumosos e diversificados, buscando reproduzir o ambiente imprevisível do mundo real (KLARE *et al.*, 2015a; MILLER *et al.*, 2015) para induzir nos modelos convolucionais uma forma de representação mais generalizante e discriminativa. O trabalho da DeepFace apresenta uma rede neural convolucional treinada com um banco de dados famoso, o SFC (*Social Face Classification*), que conta com 4,4 milhões de imagens de 4030 indivíduos. Os autores utilizaram esses dados para treinar a rede neural na tarefa de identificação facial dos 4030 indivíduos, classificando cada imagem em uma dessas classes. Em seguida, utilizaram o poder generalizador do modelo resultante para realizar verificação facial em outros dois grandes conjuntos de dados (LFW (HUANG *et al.*, 2008) e YTF (WOLF; HASSNER; MAOZ, 2011)) que continham identidades não vistas no conjunto de treinamento; para isso, serviram-se do vetor de características (*embedding*) extraído pela rede para verificar se duas imagens pertenciam à mesma identidade, computando métricas de similaridade entre os dois vetores. Já o trabalho da DeepID traz uma abordagem parecida, porém desempenhando no treinamento a identificação facial de 10000 indivíduos, usando, em seguida, os *embeddings* da rede resultante para performar verificação facial em associação a abordagens supervisionadas, como *Joint Bayesian* e redes neurais densas, que determinam a probabilidade de dois vetores pertencerem à mesma identidade.

Essa mudança de foco, observada no reconhecimento facial, acompanhou uma tendência

mais ampla do aprendizado profundo voltada à generalização entre bases de dados. Concomitantemente, o *one-shot learning*, uma outra linha de pesquisa inspirada por motivações semelhantes, vinha se desenvolvendo já desde a década de 1990 (BROMLEY *et al.*, 1993), e passando por avanços significativos mediante a integração de redes neurais convolucionais (FEI-FEI; FERGUS; PERONA, 2006), incluindo também aplicações em verificação facial (CHOPRA; HADSELL; LECUN, 2005), até atingir um nível de maturidade significativo com a proposição das redes siamesas (KOCH *et al.*, 2015). O foco desses métodos de *one-shot learning* costuma ser a verificação, ou seja, dadas duas amostras, deseja-se saber se pertencem à mesma classe, o que coaduna bastante com a proposta da verificação facial. As redes siamesas, em particular, utilizam uma metodologia de treinamento com função de perda contrastiva, induzindo a rede neural a produzir *embeddings* espacialmente distintos para pessoas distintas, e o contrário para *embeddings* da mesma pessoa. O treinamento é realizado em pares de imagem, e a rede é ensinada a prever com um valor positivo os pares de imagem pertencentes ao mesmo indivíduo, independentemente de quem seja.

$$E((X_1, X_2), Y_{true}) = \frac{1 - Y_{true}}{2} \|f(X_1) - f(X_2)\|_2^2 + \frac{Y_{true}}{2} [\max(0, m - \|f(X_1) - f(X_2)\|_2)]^2 \quad (2.11)$$

Na equação 2.11, tem-se a fórmula da função de custo contrastiva utilizada na rede siamesa; note-se que, quando os rostos são da mesma pessoa, Y_{true} é igual a 1, anulando o primeiro membro da equação, e o contrário se dá quando os rostos são de pessoas distintas (Y_{true} igual a 0). Assim, a função pune *embeddings* ($f(X_1)$ e $f(X_2)$) muito próximos quando estes pertencem a identidades diferentes, e vice-versa.

2.5.1.1 A FaceNet Como Pioneira dos Modelos Embedders

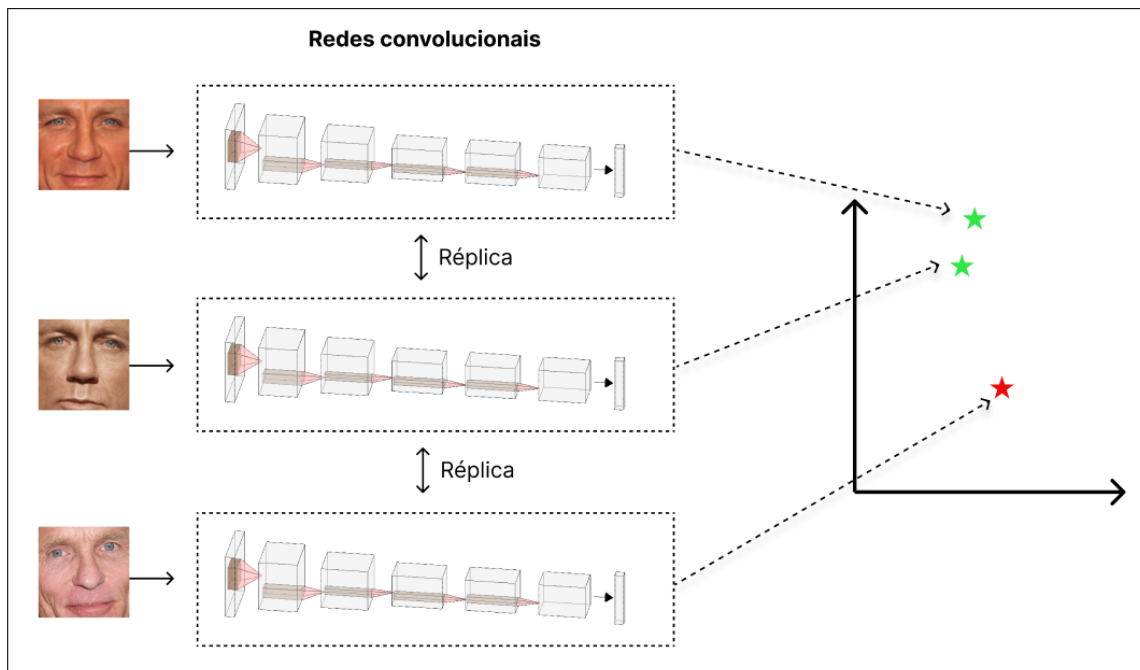
Em 2015, no mesmo ano da publicação das redes siamesas, uma equipe de pesquisadores publicou um trabalho que gerou grande impacto na academia, em que divulgaram a FaceNet (SCHROFF; KALENICHENKO; PHILBIN, 2015), uma rede neural extratora de características de faces humanas treinada por meio da função de custo *triplet loss*. Da mesma forma que as redes siamesas, a metodologia de treinamento da FaceNet foi desenhada tendo em vista o processo de verificação facial. Para isso, utilizou-se uma arquitetura de rede convolucional inspirada nas redes Inception (SZEGEDY *et al.*, 2015) (em versões mais maduras do trabalho empregou-se a InceptionResNetV1 (SZEGEDY *et al.*, 2017) para aperfeiçoar as conexões da

rede), guiada por uma tecnologia de treinamento cujo dado de entrada consiste numa tripla de imagens, duas pertencentes a um mesmo usuário A, e a outra pertencente a um determinado usuário B; a rede convolucional efetua a extração de características, codificando os padrões de cada uma das três faces em *embeddings* separados, de tamanho fixo.

$$E(X_{ancora}, X_{pos}, X_{neg}) = \max(\|f(X_{ancora}) - f(X_{pos})\|_2^2 - \|f(X_{ancora}) - f(X_{neg})\|_2^2 + \alpha, 0) \quad (2.12)$$

A função de custo, descrita na Equação 2.12, calcula a distância euclidiana entre os *embeddings* do usuário A — $f(X_{ancora})$ e $f(X_{pos})$ —, bem como a distância entre o *embedding* âncora do usuário A com o do usuário B — $f(X_{ancora})$ e $f(X_{neg})$; subtraem-se esses dois valores, posteriormente adiciona-se ao resultado o valor de uma constante *alpha* e, em seguida, anulam-se os valores inferiores a zero com a função de máximo. Note-se que essa função pune severamente os pesos da rede quando o *embedding* âncora é euclidianamente mais próximo do *embedding* negativo que do positivo, induzindo, assim, a rede convolucional a produzir padrões que, no espaço vetorial de características, tendem a ser próximos quando pertencentes a uma mesma identidade, e distantes quando oriundos de indivíduos distintos.

Figura 9 – Faces humanas codificadas como pontos em um espaço vetorial de características



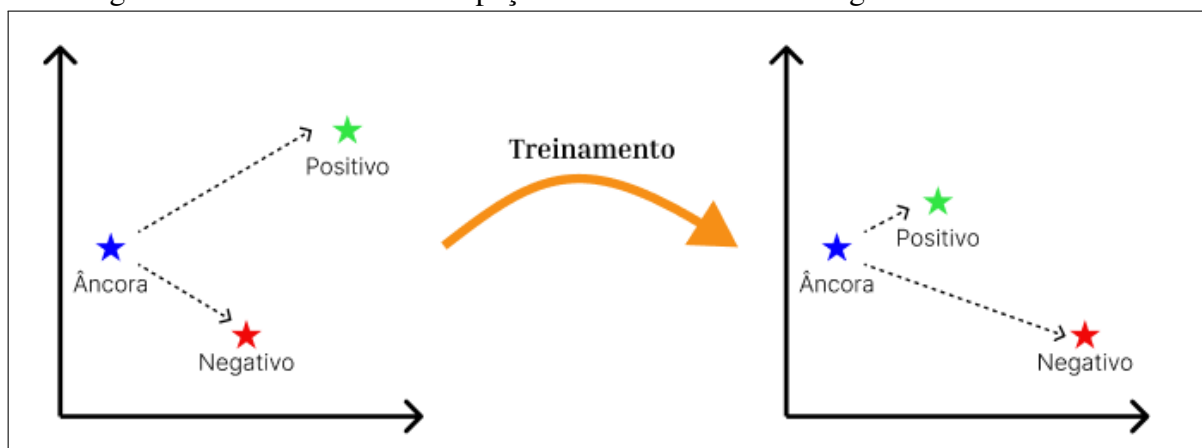
Fonte: Elaborado pelo autor

Essa dinâmica de treinamento engloba uma apreciação da dificuldade das triplas, pois triplas de imagens contendo indivíduos que, embora diferentes, têm fisionomia parecida, tendem a

ser mais desafiadoras para o modelo julgar corretamente. Foi preciso desenvolver um ecossistema de descoberta de triplas que fosse capaz de equilibrar a dificuldade dos casos a serem inseridos no treinamento, formando triplas desafiadoras o bastante para estimular o aprendizado, mas não tão difíceis a ponto de paralisar o treinamento. A esse processo dá-se o nome de Mineração de Triplas Semi-Difíceis — *semi-hard triplet mining* (SCHROFF; KALENICHENKO; PHILBIN, 2015). Alguns trabalhos posteriores apresentaram táticas alternativas para realizar essa mineração, não se limitando à formação de triplas (GE, 2018; HARWOOD *et al.*, 2017; SHI *et al.*, 2016), e outros discutiram os fundamentos e a importância dessa mineração no resultado final (HERMANS; BEYER; LEIBE, 2017).

Na Figura 9, mostra-se o estágio final da rede convolucional após o treinamento, indicando que faces de uma mesma identidade são mapeadas como pontos próximos no espaço de características; o aspecto bidimensional do gráfico dessa figura é apenas ilustrativo, pois os *embeddings* das faces são, na realidade, mapeados em um espaço de dimensão N , onde N é o tamanho fixo do vetor, escolhido pelo arquiteto da rede neural antes do treinamento. Na Figura 10 mostra-se como o processo de treinamento induz a rede neural convolucional a realizar projeções discriminativas. No início do treinamento, a rede faz um mapeamento inconsistente das faces no espaço vetorial de características; com o passar do tempo, a capacidade representativa do modelo vai sendo refinada, à medida que aprende a separar padrões de identidades distintas nesse espaço.

Figura 10 – Refinamento do espaço de características ao longo do treinamento

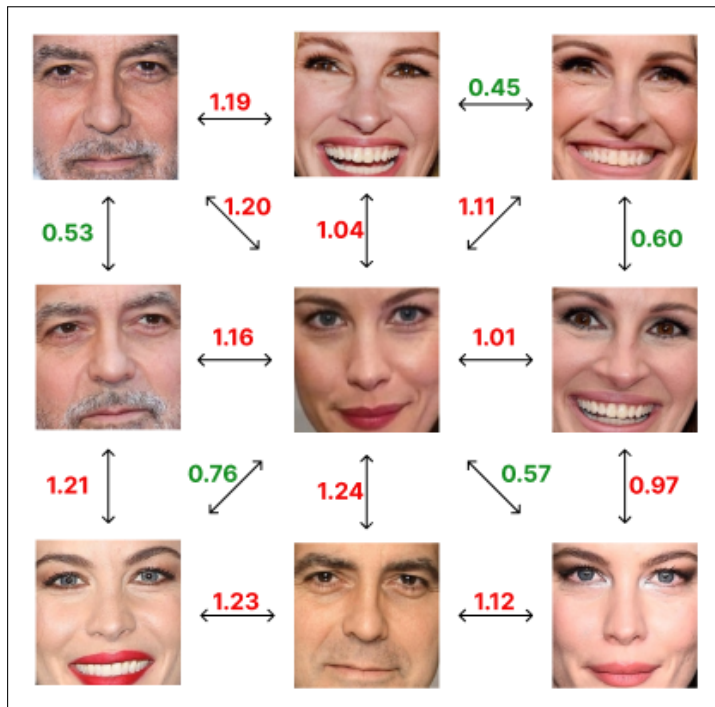


Fonte: Elaborado pelo autor

A iniciativa da FaceNet colocou o foco da metodologia na criação do espaço de características mediante o treinamento de uma rede neural convolucional, e resumiu o problema da verificação facial ao problema de encontrar um limiar de distância euclidiana abaixo do qual se

possa considerar que dois vetores são da mesma pessoa. Um exemplo ilustrativo se encontra na Figura 11, onde se vê que as distâncias entre fotos das mesmas pessoas giram em torno de uma faixa de valores mais baixos, ao passo que as distâncias entre pessoas diferentes apresentam valores elevados. Para escolher o valor de limiar que resulta no maior número de acertos, os autores da FaceNet dividiram o conjunto de treinamento em 10 subdivisões, e aplicaram validação cruzada sobre todos os pares de imagem possíveis do subconjunto de treinamento (ao total, foram 100 mil por 100 mil pares).

Figura 11 – Distâncias euclidianas entre faces com valores meramente ilustrativos



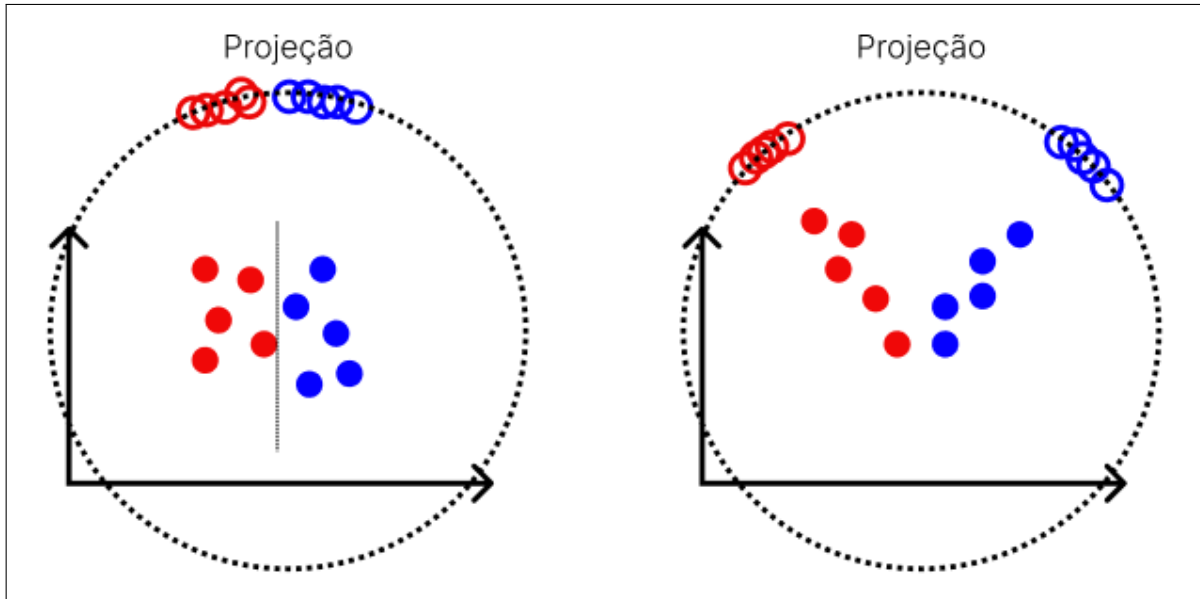
Fonte: Elaborado pelo autor

2.5.2 Outras Iniciativas de Modelos Embedders

Depois do impacto inicial da publicação da FaceNet, a metodologia proposta por esse trabalho começou a ser assimilada por estudiosos da área, que passaram a notar e explorar aspectos da construção do modelo extrator que não necessariamente estavam no campo de visão dos publicadores da FaceNet. Em um curto intervalo de tempo, observou-se uma profusão de iniciativas inovadoras que buscavam produzir modelos extratores mais discriminativos (WEN *et al.*, 2016), propondo mudanças de perspectiva geométrica na construção do espaço de características (LIU *et al.*, 2017; WANG *et al.*, 2018; DENG *et al.*, 2019), aperfeiçoando táticas de mineração das amostras (HUANG *et al.*, 2020), bem como adaptando as funções de custo para contornar problemas de qualidade de imagem (MENG *et al.*, 2021; KIM; JAIN; LIU, 2022).

Muitos modelos que efetuam mudanças de perspectiva geométrica se dedicaram a explorar a distância angular, produzindo espaços de características onde *embeddings* do mesmo indivíduo ocupam pontos próximos de uma projeção esférica. Na Figura 12, tem-se um exemplo ilustrativo do que acontece com o espaço de características produzido pela rede convolucional quando se utiliza funções de custo que forçam a separabilidade angular entre as classes. Note-se que os pontos do espaço pertencentes a uma mesma classe estão aproximadamente alinhados em uma mesma direção, o que resulta em projeções que ocupam um pequeno arco do círculo.

Figura 12 – Projeções esféricas de *embeddings*



Fonte: Elaborado pelo autor, inspirado em uma imagem do artigo da SphereFace

$$Y_i = \frac{e^{-(E(X) \cdot W_i^T + b)}}{\sum_{j=0}^N e^{-(E(X) \cdot W_j^T + b)}} \quad (2.13)$$

Similarmente à DeepFace e à DeepID, duas redes convolucionais recentes, a SphereFace (LIU *et al.*, 2017) e a CosFace (WANG *et al.*, 2018), também foram treinadas para efetuar a tarefa de identificação facial, e a verificação é efetuada mediante uso dos *embeddings* da rede resultante. No entanto, elas superam limitações da função de ativação *softmax*, utilizada comumente para identificação facial, mediante uma mudança na equação dessa função, alterando o lugar geométrico que caracteriza a fronteira decisória. A fórmula analítica da saída de uma rede que utiliza a função *softmax* original na camada de saída está expressa na Equação 2.13, onde $E(X)$ é o *embedding* gerado pela rede, e N é o número de classes da identificação; o vetor Y representará, em cada elemento Y_i , a probabilidade de o rosto X pertencer à classe i . A fronteira

de decisão, ou seja, o lugar geométrico que representa o conjunto de valores para os quais Y_i e Y_j têm probabilidades iguais é $(W_1 - W_2) E(X) + (b_1 - b_2) = 0$

A SphereFace insere na função *softmax* uma inovação para gerar um espaço de características angularmente separável. Para isso, efetua-se a normalização da matriz de pesos W , assegurando que $\|W\|_2^2 = 1$, anulam-se os vieses ($b = 0$); isto faz com que o produto escalar $E(X) \cdot W_j$ seja proporcional ao ângulo entre esses dois vetores; como $\|W_j^T\| = 1$, a expressão do produto $E(X) \cdot W_j^T$ se resume a $\|E(X)\| \cos(\theta_j)$, onde θ_j é o ângulo formado por $E(X)$ e W_j^T .

$$Y_i^{SphereFace} = \frac{e^{-(\|E(X)\| \cos(m\theta_i))}}{e^{-(\|E(X)\| \cos(m\theta_i))} + \sum_{j \neq i}^N e^{-(\|E(X)\| \cos(\theta_j))}} \quad (2.14)$$

Adicionando um fator de escala $m \geq 2$ apenas ao ângulo da componente i da função *softmax* ($\theta_i \leftarrow m\theta_i$), a nova formulação da função passa a ser descrita pela Equação 2.14. A nova fronteira de decisão é definida, num caso onde só há duas classes, pela fórmula $\|E(X)\| (\cos(m\theta_1) - \cos(\theta_2))$. Assim, para classificar corretamente uma face, o modelo é forçado a produzir *embeddings* que satisfaçam a relação $\cos(m\theta_1) \geq \cos(\theta_2)$, e dessa forma o algoritmo de treinamento o induz a gerar uma separação angular maior entre vetores de classes distintas.

$$Y_i^{CosFace} = \frac{e^{-(\|E(X)\| (\cos(\theta_i) - m))}}{e^{-(\|E(X)\| (\cos(\theta_i) - m))} + \sum_{j \neq i}^N e^{-(\|E(X)\| \cos(\theta_j))}} \quad (2.15)$$

Uma mudança sutilmente diferente da promovida pela SphereFace, é a que foi implementada na CosFace, cuja publicação se deu no mesmo ano, e que modifica a *softmax* (Equação 2.15) de modo a produzir fronteiras de decisão do tipo $\cos(\theta_1) \geq \cos(\theta_2) + m$, que é outra maneira de induzir separabilidade angular aos vetores de características produzidos pela rede neural convolucional.

Quadro 1 – Fronteiras de Decisão por Rede Convolucional

Rede Convolucional	Fronteira de Decisão
SphereFace	$\ E(X)\ (\cos(m\theta_1) - \cos(\theta_2))$
CosFace	$\ E(X)\ (\cos(\theta_1) - \cos(\theta_2) - m)$
ArcFace	$\ E(X)\ (\cos(\theta_1 + m) - \cos(\theta_2))$

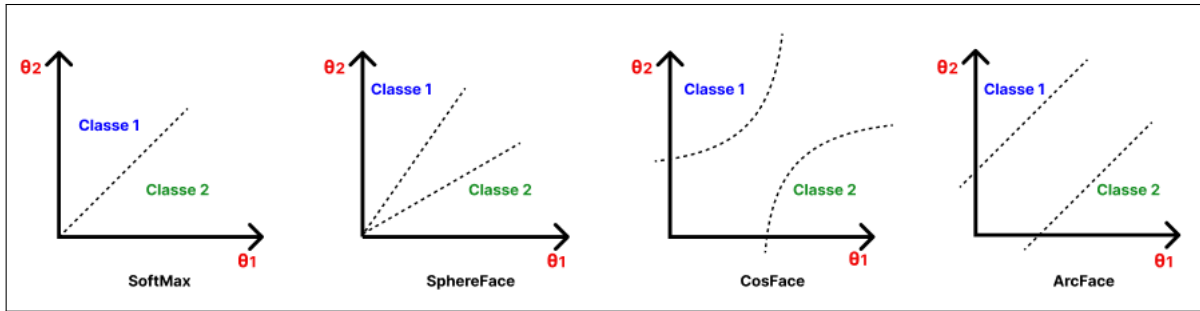
Fonte: Elaborado pelo autor

Uma rede mais conhecida, a ArcFace (DENG *et al.*, 2018), também publicada em 2018, introduz uma mudança semelhante, adicionando o fator m ao próprio ângulo, alterando a

fórmula da função de ativação de acordo com a Equação 2.16, que induz a produção de margens propriamente angulares.

$$Y_i^{ArcFace} = \frac{e^{-(\|E(X)\| \cos(\theta_i + m))}}{e^{-(\|E(X)\| \cos(\theta_i + m))} + \sum_{j \neq i}^N e^{-(\|E(X)\| \cos(\theta_j))}} \quad (2.16)$$

Figura 13 – Gráfico das fronteiras de decisão de SphereFace, CosFace e ArcFace



Fonte: Elaborado pelo autor, inspirado em uma imagem do artigo da ArcFace

Apesar da semelhança com os outros dois modelos, a margem produzida pela ArcFace é equivalente à distância geodésica, o que lhe confere melhor separabilidade. As fronteiras de decisão dessas redes estão expressas graficamente na Figura 13. A largura das margens são meramente ilustrativas.

Note-se que a ArcFace é a única rede em que as margens da fronteira de decisão estão distantes entre si por um valor constante, o que torna o processo de convergência do algoritmo de treinamento mais estável, bem como induz o modelo a separar as classes umas das outras com aproximadamente a mesma folga. A SphereFace, por exemplo, que multiplica o ângulo pelo fator m , precisou de uma supervisão no início do treinamento para enfraquecer a penalização multiplicativa, pois estava sofrendo de divergência no algoritmo (LIU *et al.*, 2017; DENG *et al.*, 2018).

Por fim, destacam-se as GhostFaceNets (ALANSARI *et al.*, 2023), uma série de redes convolucionais que mantém o uso da função de custo definida no trabalho da ArcFace, porém alterando a arquitetura subjacente, tornando-as mais leves mediante o uso de módulos da GhostNet (HAN *et al.*, 2020), realizando o treinamento no conjunto de dados MS-Celeb-1M (GUO *et al.*, 2016). Existem múltiplas versões desse conjunto de redes, e a utilizada neste trabalho para testes foi *v1.2*.

2.6 Estudos de Aplicação de Modelos *Embedders*

Nesta seção se apresentará o estado atual de discussões sobre modelos *embedders* para verificação facial, passando por trabalhos comparativos e exploratórios, e outros que focam em aplicações práticas de redes convolucionais extratoras de características de face. Tudo isso servirá como base para compreender como a metodologia de testes e as técnicas propostas nesta dissertação se enquadram no panorama acadêmico geral da tarefa de verificação facial.

Nos últimos anos, o uso de modelos *embedders* capazes de produzir espaços de características discriminativos no campo da biometria facial tem se difundido. Alguns estudos aplicam esses modelos para resolver problemas de reconhecimento facial em conjuntos de dados não vistos no treinamento original (DENG; ZAFERIRIOU, 2019; WILLIAM *et al.*, 2019; CAHYONO; WIRAWAN; RACHMADI, 2020; QI *et al.*, 2022; LUCENA *et al.*, 2022; MONTERO *et al.*, 2022; PAPANTONIOU *et al.*, 2024); outros, dedicam-se a comparar a qualidade dos vetores produzidos por essas redes (ROSTUM; VÁSÁRHELYI, 2024), mostrando que, apesar dos aperfeiçoamentos de separação angular promovidos pela ArcFace, a FaceNet segue como uma rede competitiva, demonstrando, em alguns trabalhos, desempenho bastante superior (IYPE; SEBASTIAN, 2023; FIRMANSYAH; KUSUMASARI; ALAM, 2023). Em meio a esses trabalhos, destaca-se um que evidencia que os vetores produzidos por um certo modelo *embedder* podem ser comparados aos vetores de um outro modelo após sofrer rotação ou alguma outra transformação linear (MCNEELY-WHITE *et al.*, 2022), sem grandes perdas de performance.

Com foco no aprendizado de máquina associado ao uso de vetores de modelos *embedders*, destaca-se a SVM (SANCHEZ-MORENO *et al.*, 2021), que tem sido bastante utilizada para efetuar classificação mediante manipulação dos *embeddings* desses modelos (RYANDO *et al.*, 2024; GHADA *et al.*, 2024). É frequente o uso de Máquinas de Vetores Suporte que usam como amostra o próprio *embedding* gerado pela rede; outros propõem que se calcule o valor de distância de vários pares de imagem positivos (da mesma pessoa) e negativos (de pessoas distintas) e se utilize esses valores como amostra do classificador SVM, criando um modelo por indivíduo para desempenhar verificação facial. Um desses trabalhos utiliza uma rede treinada para identificação e sugere usar o algoritmo PCA para diminuir a discrepância entre os domínios de problema antes de treinar (XING *et al.*, 2017).

Ademais, o estudo da DeepFace (TAIGMAN *et al.*, 2014) propôs, juntamente com o modelo *embedder* que caracteriza a entrega principal desse trabalho, uma tática de *one-shot learning*, a qual consiste em fazer um classificador neural aprender a julgar se dois *embeddings*

pertencem à mesma pessoa alimentando-o com a diferença ponto a ponto entre esses *embeddings*, construindo, dessa forma, um classificador global, que serve para todos os usuários. Basicamente, depois de treinado o classificador, toma-se o *embedding* de um rosto que se deseja validar, e um *embedding* do rosto da pessoa alvo, para a rede neural inferir a resposta com base no vetor diferença.

$$\chi^2(x_1, x_2) = \frac{(x_1 - x_2)^2}{x_1 + x_2} \quad (2.17)$$

O algoritmo utilizado como classificador pelos autores da DeepFace foi o SVM, e a diferença entre os vetores não é a mera diferença ponto a ponto, mas é uma diferença χ^2 , expressa pela Equação 2.17. Um dos métodos que será utilizado como *baseline* para testar o método proposto nesta dissertação é inspirado nessa estratégia de manipulação proposta no trabalho da DeepFace, sofrendo algumas adaptações para ajustar-se a um ambiente de recursos computacionais pouco abundantes. No entanto, nesse trabalho da DeepFace não se informou a forma de escolher as imagens de uma pessoa, quando há muitas disponíveis no banco. No trabalho da DeepID (SUN; WANG; TANG, 2014), é implementada uma tática semelhante, utilizando probabilidades conjuntas para testar os pares de faces.

2.7 Avaliação de Métodos para Verificação Facial

Após apresentar o panorama histórico e tecnológico da tarefa de verificação facial, dedica-se aqui um espaço para discutir as formas de quantificar a qualidade de um sistema que pretenda desempenhar essa tarefa. Logo, serão apresentados protocolos de experimento, métricas de avaliação e gráficos como a curva ROC e a curva de Precisão-Recall.

2.7.1 Protocolos de Experimento

Em um sistema de verificação facial, em que se busca confirmar se uma determinada face realmente pertence a uma identidade, o protocolo de testagem das faces pode ser **global** ou **individual**. A verificação de protocolo global se dá mediante o cálculo de métricas de avaliação sobre **todas as amostras** de verificação independentemente do indivíduo que está sendo validado. O trabalho da FaceNet (SCHROFF; KALENICHENKO; PHILBIN, 2015) utiliza essa tática de validação quando computa a acurácia com base nas inferências da rede para todos os pares possíveis do conjunto de testes. Constroem-se 100.000 x 100.000 pares de imagens; em seguida,

a rede infere se as duas imagens integrantes de cada par pertencem à mesma pessoa. As métricas são calculadas a partir do confronto entre essas inferências e o resultado esperado.

Já o protocolo individual de experimento concentra a análise no indivíduo (na classe) do conjunto de dados, calculando as métricas de avaliação considerando o contexto de inferência de cada indivíduo em particular; posteriormente, agrega-se os resultados mediante aplicação de funções estatísticas, por exemplo, a média. Nesse protocolo, cada indivíduo é testado contra todos os outros indivíduos da base de dados, e se espera que o sistema rejeite como incorretas as amostras que resultaram da comparação entre faces desse indivíduo e as faces de teste de outros indivíduos, e acolha como positivas as amostras que foram formadas pela comparação das faces registradas desse indivíduo no sistema e as faces teste desse mesmo indivíduo. Ao final, a quantificação da qualidade do sistema é uma média do desempenho do sistema por indivíduo. Esse protocolo é bastante útil para identificar disparidades de desempenho devido à etnia, sexo, quantidade de imagens de um indivíduo na base etc., bem como para os casos em que é necessário estabelecer um limiar de confiabilidade por usuário.

2.7.2 Métricas

No campo da inteligência artificial, é comum utilizar métricas de avaliação como a acurácia, que consiste em calcular o percentual de amostras corretamente inferidas em relação ao total de inferências realizadas. No entanto, essa métrica pode não expressar fielmente seu desempenho em problemas do mundo real. Considere um cenário em que se deseja inferir a presença de uma doença rara que acomete uma pessoa a cada dez mil. Se a inteligência artificial inferir que ninguém tem a doença, ou seja, dar um diagnóstico negativo a todos, a acurácia será de 99,99%, o que, à primeira vista, pode parecer uma elevada taxa de acerto; porém, a verdade é que o modelo acertou 100% dos casos em que os pacientes realmente não tinham a doença, mas errou 100% dos casos em que havia de fato uma manifestação da doença rara, o que demonstra que o modelo está bem abaixo do necessário para ser utilizado em produção.

Diante disso, pode-se, a princípio, deslocar o foco da análise da acurácia global para avaliar os acertos e erros que acontecem em cada uma das classes do problema. Considerando, por simplificação, um cenário binário, a acurácia pode ser quebrada em duas componentes, a *sensibilidade (recall)*, e a *especificidade*, que em cenários binários é intitulada *True Negative Rate*.

No Quadro 2, são indicadas as diversas categorias de acerto e erro possíveis. O TP

Quadro 2 – Matriz de Confusão

	Classe Positiva	Classe Negativa
Inferência Positiva	TP	FP
Inferência Negativa	FN	TN

Fonte: Elaborado pelo autor

(*True Positive*) e o FN (*False Negative*), por exemplo, são, respectivamente, a quantidade de amostras positivas corretamente classificadas como positivas e a quantidade de amostras positivas incorretamente classificadas como negativas.

$$TPR = \frac{TP}{TP + FN} \quad FNR = \frac{FN}{TP + FN} \quad (2.18)$$

$$TNR = \frac{TN}{TN + FP} \quad FPR = \frac{FP}{FP + TN} \quad (2.19)$$

$$TPR + FNR = 1 \quad TNR + FPR = 1 \quad (2.20)$$

Nas Equações 2.18, 2.19, estão definidas as expressões de TPR (*True Positive Rate*), TNR (*True Negative Rate*), FPR (*False Positive Rate*) e FNR (*False Negative Rate*). A Equação 2.20 estabelece uma relação de complementariedade entre elas.

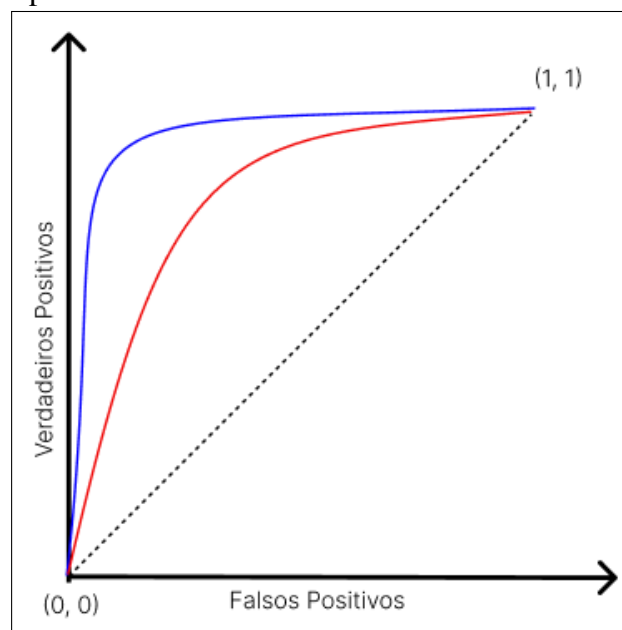
2.7.2.1 Curva ROC

No contexto de biometria facial com protocolo de experimento individual, esses conceitos são bastante úteis para compreender o comportamento das inferências do modelo, sobretudo porque nos cenários reais de aplicação de verificação facial o conjunto de dados é aberto e extremamente desbalanceado: só há um rosto que coincide com a identidade de um indivíduo específico, mas há bilhões de outros rostos que devem ser rejeitados pelo modelo como não pertencentes a esse indivíduo. Para medir a capacidade de rejeitar amostras negativas, utiliza-se a TNR (ou, de forma equivalente, a FPR, quando se quer quantificar o erro de aceitação). Para medir a capacidade de aceitar amostras positivas, utiliza-se a TPR (recall) (ou a FNR, quando se quer quantificar o erro de rejeição). Em geral, busca-se TNR/TPR altas e FPR/FNR baixas.

No entanto, o comportamento de sistemas biométricos de verificação costuma apresentar uma certa oposição entre um bom nível de rejeição de amostras negativas, e níveis aceitáveis de aceitação de amostras positivas. Quando um modelo é muito competente em rejeitar faces

que não são de uma determinada identidade, ele tende a ficar mais rígido e propenso a rejeitar até mesmo as faces dessa mesma identidade que, em virtude de pequenas distorções do cenário, são ligeiramente diferentes das faces armazenadas no banco de dados; por outro lado, quando o modelo aceita bem as faces do indivíduo, mesmo quando apresentam diferenças significativas em relação às cadastradas no sistema, ele tende a aceitar como pertencentes a esse indivíduo imagens de face que não são dele; assim, o modelo pode apresentar um critério mais ou menos rígido de aceitação, e essa rigidez pode ser controlada modificando o nível de confiança necessário para inferir uma amostra como positiva. Quando se assume níveis de confiança muito altos, o modelo exige um alto grau de probabilidade para considerar a amostra como positiva, rejeitando tudo que não satisfaz a esse critério; nesse caso, será um TPR muito baixo e um FNR elevado, pois ele rejeitará imagens do próprio usuário. Todavia, quando o nível de confiança é baixo, o FNR será baixo, visto que aceitará todas as imagens que realmente pertencem ao usuário; em contrapartida, pode acabar aceitando como positivas algumas imagens que não são do usuário, elevando o TPR.

Figura 14 – Exemplo de Curva ROC



Fonte: Elaborado pelo autor

Para descrever como o sistema se comporta quando se varia o nível de confiança, calcula-se a curva ROC, que traz no eixo X os valores FPR, e os valores de TPR no eixo Y. Na Figura 14, mostra-se um exemplo típico de curva ROC comparando dois modelos: um representado pela linha azul, e outro representado pela linha vermelha. Note-se que, à medida que o número de verdadeiros positivos (TPR) aumenta, também aumenta a quantidade de falsos positivos. O

modelo cuja curva fica mais acima, é justamente aquele que consegue atingir mais rapidamente o 100% de verdadeiros positivos mantendo uma taxa de falsos positivos baixa; no caso da Figura 14, esse modelo é representado pela linha azul, indicando um desempenho melhor. Na prática, para quantificar o desempenho, ou seja, quão rapidamente ele é capaz de atingir o pico de TPR, calcula-se a área sob a curva (AUC, *Area Under Curve*) da linha que representa o modelo; quanto mais a área se aproximar de 1, mais próximo do ideal é o modelo. Nessa mesma figura, há uma linha pontilhada com inclinação de 45 graus que engloba uma área igual a 0.5, representando o modelo que faz inferências aleatoriamente, servindo como um ponto de referência para as outras curvas do gráfico.

2.7.2.2 Curvas de Precisão e Recall

Quando se trata de problemas em que há um desbalanceamento extremo no número de amostras das classes, o que coincide com o problema de verificação facial no mundo real, a curva ROC pode não ser capaz de expressar o real desempenho do sistema, embora já seja um grande avanço em relação ao uso simples e direto de métricas como acurácia, TPR, FPR e outras. Para compreender isso, é preciso entender o conceito de **precisão**, que é uma métrica que pode ser usada em associação com a métrica TPR (*recall*) para destacar aspectos do comportamento do sistema: o espaço amostral da precisão é o conjunto de inferências positivas do sistema, e calcula-se o percentual de amostras realmente positivas nesse conjunto; já o espaço amostral do TPR é o conjunto de amostras realmente positivas, e é calculado o percentual de inferências positivas que o sistema fez. O foco da precisão está nas inferências do sistema; uma precisão alta indica que, quando o sistema inferiu que uma amostra é positiva, muito provavelmente ela era realmente positiva; já uma TPR alta indica que, se uma amostra é positiva, provavelmente o sistema a classificará como positiva. A fórmula da precisão está na Equação 2.21.

$$Prec = \frac{TP}{TP + FP} \quad (2.21)$$

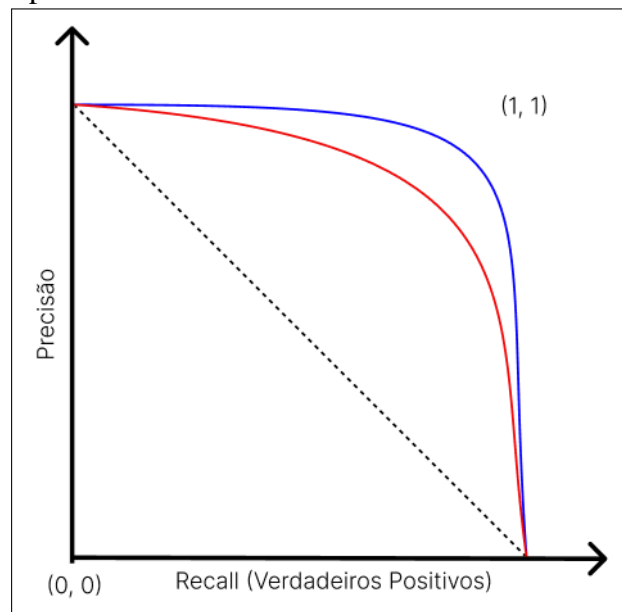
Quadro 3 – Matriz de Confusão de Verificação Biométrica

	Rosto pertence à pessoa	Rosto não pertence à pessoa
Inferência Positiva	4	3
Inferência Negativa	1	122

Fonte: Elaborado pelo autor

Considere o exemplo de um sistema de verificação biométrica no Quadro 3. Usando as

Figura 15 – Exemplo de Curva PR



Fonte: Elaborado pelo autor

Equações 2.18 e 2.19, descobre-se que o valor de TPR é 80% ($\frac{4}{4+1}$), e de FPR é 2,4% ($\frac{3}{3+122}$). Como se pode ver, a taxa de falsos positivos é relativamente baixa, e a de verdadeiros positivos é razoavelmente alta. Porém, quando se mede a precisão do modelo, cujo valor é 57,14% ($\frac{4}{3+4}$), que é uma precisão baixa, indicando que, neste caso, se a taxa de falsos positivos é baixa, não é porque o modelo está bem treinado, mas porque há um viés que o torna propenso a inferir a classe negativa (rejeitar), o que reduz FPR em bases desbalanceadas. Como se pode ver, essa informação só é trazida à tona pela métrica de precisão, que avalia a qualidade das predições positivas do modelo, punindo a ocorrência de falsos positivos mais severamente que a própria taxa de falsos positivos FPR. Isso é bastante útil para sistemas de verificação biométrica, pois o falso positivo é o mais grave erro de sistema, que implica na não rejeição de um impostor.

Da mesma forma que a curva ROC descreve o comportamento das taxas TPR e FPR, variando o nível de confiança da inferência, existe uma curva, chamada Curva de Precisão e Recall, que descreve esse comportamento colocando a precisão no eixo Y e o *recall* no eixo X . As curvas que representam os sistemas começam no ponto mais alto. Terá melhor desempenho a curva que conseguir manter a precisão alta por mais tempo enquanto, avança ao longo do eixo de verdadeiros positivos, o que é quantificado também pelo cálculo da área sob a curva (AUC). No caso da Figura 15, a linha azul é a que atinge, no eixo X , o ponto mais distante da origem sem sofrer uma queda brusca no eixo Y .

3 PROPOSTAS E METODOLOGIA

Nas seções anteriores, estabeleceram-se os fundamentos que norteiam os processos de verificação facial envolvendo modelos *embedders*, bem como resumos técnicos das propostas e discussões recentes que estão alinhadas com esse caminho de pesquisa. Nesta seção serão apresentadas as estratégias propostas para manipular os *embeddings* de modelos de rede convolucional e a metodologia de testes adotada para validar os experimentos.

3.1 Intuições Preliminares

Tome-se como ponto de partida o fato de que as distâncias entre pares de *embeddings* que representam o rosto de uma mesma pessoa assumem valores baixos, quando os padrões faciais são representados em um espaço de características rico e discriminativo. Disso resulta que uma representação gráfica que mostra distâncias entre *embeddings* de classes idênticas e de classes diferentes terá a aparência de dois conjuntos visualmente separáveis no espaço. No entanto, quando se utiliza apenas uma dimensão para efetuar essa representação gráfica, pode-se, ainda assim, obter resultados abaixo do desejado; quando se avança para mais dimensões, o aspecto dos dois conjuntos vai ganhando uma separação mais nítida. Considere o caso de duas dimensões: para cada face no banco de dados, constrói-se um par de valores de distância que resultam da comparação do *embedding* dessa face com os *embeddings* de outras imagens da mesma pessoa; repete-se esse processo com duas faces de identidades diferentes daquela primeira. Aplicando isso a todos os usuários da base, obter-se-á um gráfico bidimensional em que a zona divisória que separa os conjuntos positivo e negativo é mais bem definida, em comparação à que é gerada pelos valores unidimensionais de distância resultantes da primeira comparação. Dessa forma, à medida que se utilizam valores de distância para representar pontos em espaços vetoriais de dimensões maiores, tende-se a promover melhor separabilidade.

Figura 16 – Representação gráfica unidimensional de distâncias entre embeddings

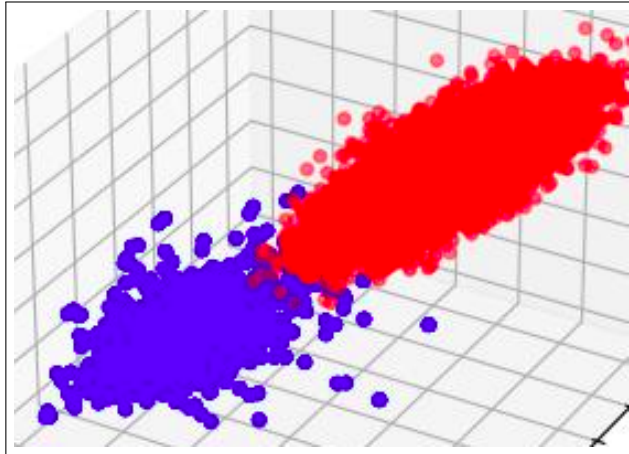


Fonte: Elaborado pelo autor

Na Figura 16, estão representadas as distâncias euclidianas em uma só dimensão: o conjunto vermelho foi colocado acima do azul, para demonstrar mais claramente a intersecção entre os dois conjuntos, acusando a existência de uma margem de separação pouco discriminativa

entre as classes.

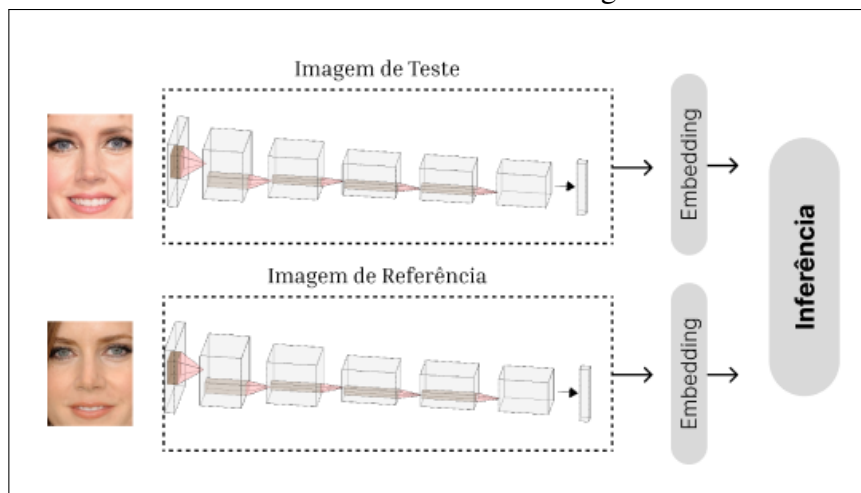
Figura 17 – Representação gráfica tridimensional de distâncias entre embeddings



Fonte: Elaborado pelo autor

Já na Figura 17, em que se utilizou três dimensões, observa-se que os dois conjuntos se distanciaram. Isso significa que, embora a grande melhoria promovida pelos algoritmos de *one-shot learning* seja minimizar a necessidade de se possuir muitas amostras de uma mesma classe, a presença de múltiplos exemplos não precisa ser descartada, mas pode ser utilizada em favor dos algoritmos de verificação facial. As distintas abordagens estão ilustradas nas Figuras 18 e 19.

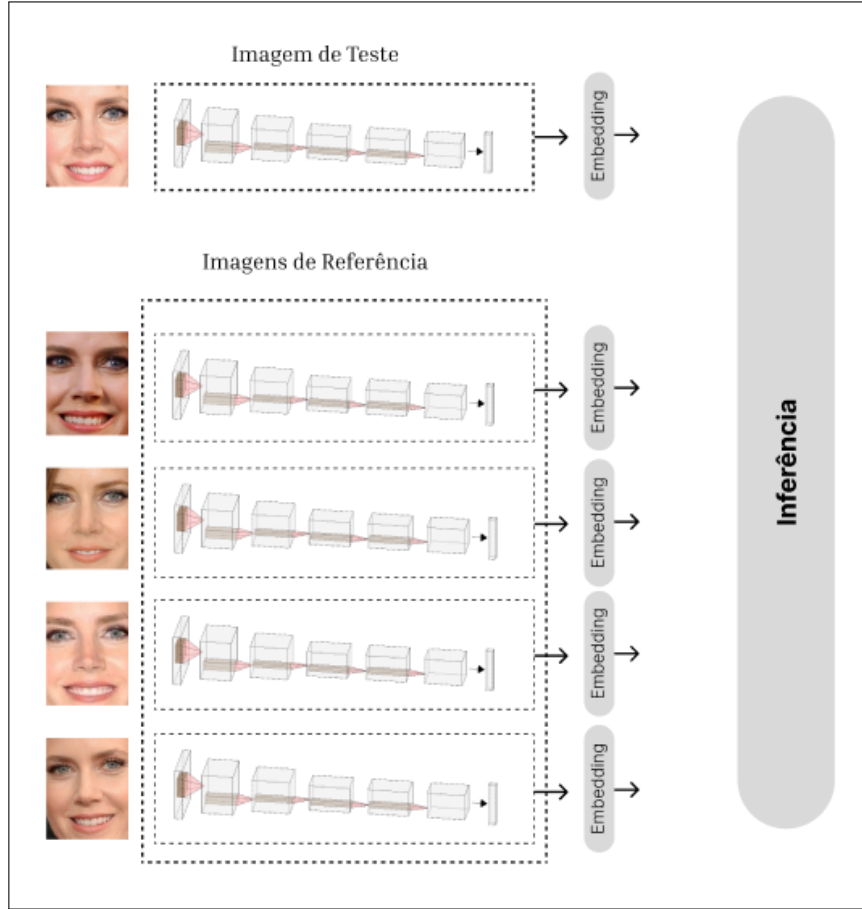
Figura 18 – Inferência tradicional utilizando uma imagem de referência



Fonte: Elaborado pelo autor

Há casos em que a aplicação simples e direta de um limiar sobre um valor de distância resultaria em falsos positivos. Muitas vezes, essas inferências podem ser corrigidas se houver uma estratégia que leve em consideração mais de um valor de distância. Aumentando o número de imagens de referência, os algoritmos que podem ser usados para efetuar a verificação facial

Figura 19 – Inferência utilizando múltiplas imagens de referência



Fonte: Elaborado pelo autor

não ficariam presos a um único exemplo, diminuindo a chance de se prenderem a aspectos muito específicos da amostra dada. Nesse sentido, com o aumento de amostras de referência, a tendência é de reduzir a variância do sistema, tornando-o menos suscetível a apresentar mudanças bruscas na inferência em resposta a pequenas alterações nos dados de entrada, fazendo-o, assim, encontrar fronteiras de decisão mais estáveis.

3.2 Proposta 1

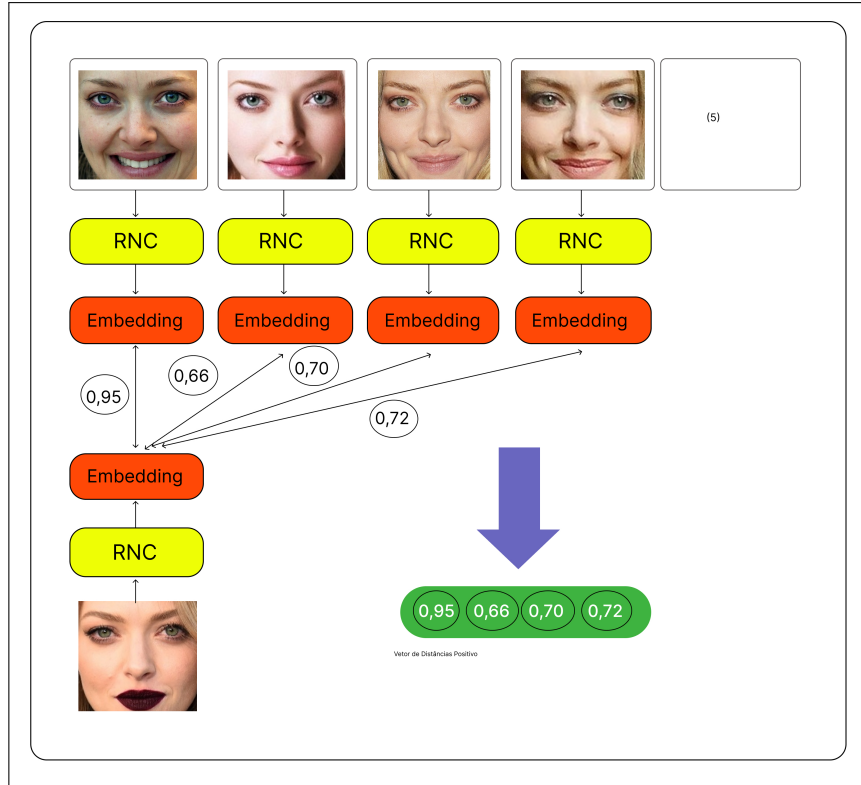
Na Equação 3.1, considere uma função de *embeddings* E que transforma $I_i^{(A)}$, a i -ésima imagem de referência da identidade A , em um vetor de características $E(I_i^{(A)})$.

$$P_i^{(A)} = \left[d(E(I_i^A), E(I_j^A)) \right]_{i \neq j} \in \mathbb{R}^{N-1} \quad (3.1)$$

O i -ésimo vetor de distância calculado no contexto da identidade A é representado por $P_i^{(A)}$, o qual é preenchido pelos valores de distância resultantes da comparação entre o *embedding* âncora $E(I_i^{(A)})$ e os demais *embeddings* pertencentes a essa mesma identidade A , excetuando

o próprio *embedding* âncora, que não é comparado a si mesmo. Assim, formam-se N vetores de distância positivos, cada qual possuindo tamanho igual a $N - 1$; nesse caso, N é o número de imagens de referência da identidade A no conjunto de dados de treinamento. A união desses vetores, que resulta em uma matriz $N \times (N - 1)$, é o conjunto de amostras positivas ($P^{(A)}$) do classificador que será treinado para esta identidade. A Figura 20 representa o processo de construção de um vetor de distâncias positivo, com valores meramente ilustrativos.

Figura 20 – Esquema da formação de um vetor de distâncias positivo

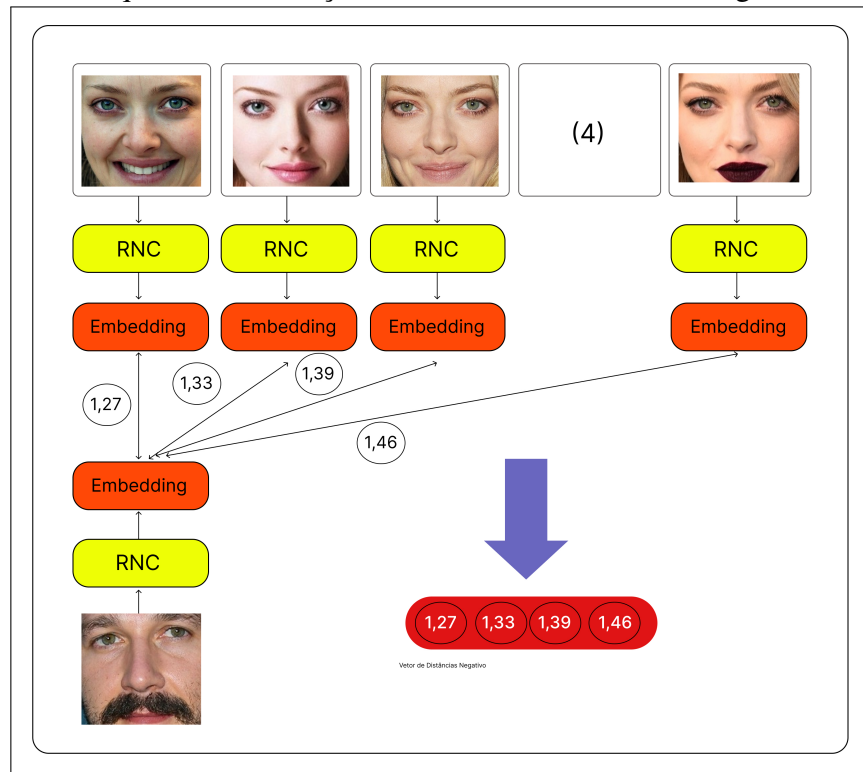


Fonte: Elaborado pelo autor

Na Equação 3.2, o i -ésimo vetor de distâncias negativo da identidade A é representado por $N_i^{(A)}$, que é definido pela comparação de um *embedding* âncora $E(I_i^B)$, pertencente à identidade B , com $N - 1$ *embeddings* da identidade A , onde N é o número de imagens de referência pertencentes a A no conjunto de dados de treinamento. A união desses vetores, que resulta em uma matriz $N \times (N - 1)$, é o conjunto de amostras negativas ($N^{(A)}$) do classificador que será treinado para esta identidade.

$$N_i^{(A)} = \left[d(E(I_i^B), E(I_j^A)) \right]_{j \neq i} \in \mathbb{R}^{N-1}, \quad k \in 1, 2, \dots, N \quad (3.2)$$

Figura 21 – Esquema da formação de um vetor de distâncias negativo



Fonte: Elaborado pelo autor

A Figura 21 representa o processo de construção de um vetor de distâncias negativo, com valores meramente ilustrativos.

Propõe-se o treinamento de uma SVM para cada indivíduo, cujas amostras positivas (classe +1) são os vetores de distância positivos calculados em relação a este indivíduo, e as amostras negativas (classe -1) são os vetores de distância negativos calculados nesse contexto. Para o treinamento, são usadas todas as amostras positivas de P^A . Contudo, apenas um subconjunto de N^A (que é bem maior que P^A) é utilizado, mantendo uma proporção de 1 amostra positiva para cada 2,3 negativas.

Após o treinamento dos classificadores, o processo de verificação facial acontece de acordo com os seguintes passos:

- Toma-se o *embedding* $E(I_\alpha)$ de um rosto desconhecido I_α , desejando verificar se este rosto pertence à identidade A ;
- Tomando o *embedding* $E(I_\alpha)$ como vetor âncora, utiliza-se a Equação 3.2 para calcular o vetor de distâncias;
- Utiliza-se o vetor de distâncias como dado de entrada para o classificador treinado para a identidade A ;

- Caso o classificador sinalize um valor de probabilidade superior ao limiar pré-fixado (0,5), considera-se que o rosto I_α , até então desconhecido, pertence à identidade A .

3.3 Proposta 2

Propõe-se gerar um classificador SVM único para realizar verificação facial em todas as identidades da base, bem como em identidades situadas fora da base, com reduzida necessidade de retreinamento. Para isso, tomam-se, como amostras positivas (classe +1), a união de todas as amostras positivas calculadas no contexto de cada identidade; semelhante processo é feito para o caso negativo.

3.3.1 Amostras positivas do classificador global

Considere que $P^{(A_i)}$ é o conjunto de amostras positivas da identidade A_i , que é calculado utilizando a Equação 3.1. O conjunto de amostras positivas para treinar o classificador único (P) é definido, na Equação 3.3, como a união de cada P^{A_i} , $\forall i \leq M$, onde M é o número de identidades da base.

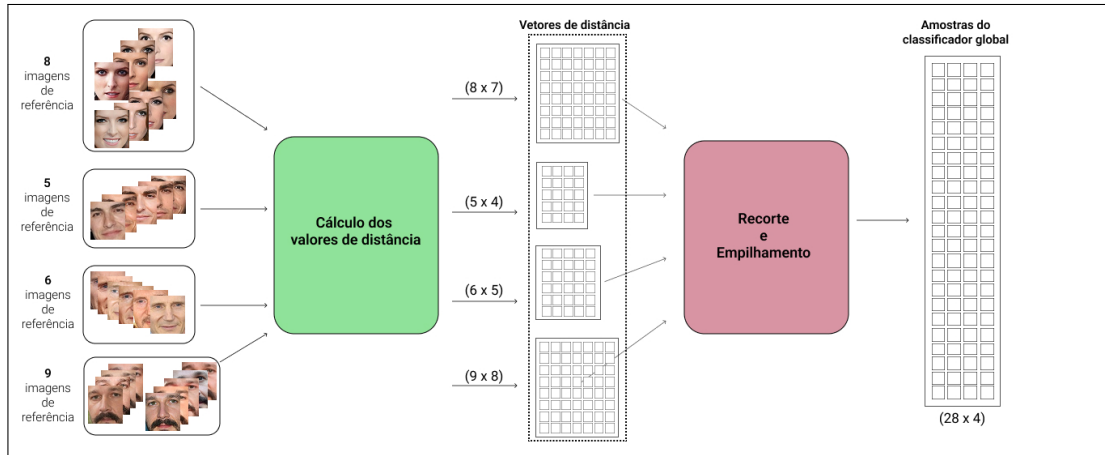
$$P = P^{(A_1)} \cup P^{(A_2)} \cup \dots P^{(A_M)} \quad (3.3)$$

A enunciação dessa ideia já levanta, por si, o problema da dimensionalidade, pois, se uma identidade A possui K imagens na base de treinamento, serão geradas, para esta identidade, K vetores de tamanho $K - 1$. Como as identidades da base possuem diferentes quantidades de imagens faciais, é necessário cortar o conjunto P , a fim de padronizar o tamanho das amostras, pois assim elas podem ser levadas ao mesmo classificador único. Para isso, seleciona-se apenas $L - 1$ valores de cada vetor de distância, onde L é o número de imagens da identidade menos frequente da base, conforme ilustrado na Figura 22. Note-se, nessa Figura, que as amostras das diferentes identidades foram empilhadas, resultando numa matriz com mais linhas, porém com apenas 4 colunas, visto que, em tal Figura, a identidade menos frequente possui apenas 5 imagens, e gerou vetores de tamanho 4.

3.3.2 Amostras negativas do classificador global

Semelhantemente, as amostras negativas são a união dos conjuntos de amostras negativas calculados para cada classe, segundo a Equação 3.4, onde M é o número de identidades da base.

Figura 22 – Fluxo para construção de amostras de treinamento. Valores meramente ilustrativos.



$$N = N^{(A_1)} \cup N^{(A_2)} \cup \dots N^{(A_M)} \quad (3.4)$$

3.3.3 Treinamento e Testes

Para o treinamento, todas as amostras positivas foram aproveitadas, ao passo que, dos conjuntos de amostras negativas, foi utilizado apenas um subconjunto, respeitando a proporção de 1 para 2,3 adotada na Proposta 1.

Após o treinamento do classificador, o processo de verificação facial acontece de acordo com os seguintes passos:

- Toma-se o *embedding* $E(I_\alpha)$ de um rosto desconhecido I_α , desejando verificar se este rosto pertence à identidade A ;
- Tomando o *embedding* $E(I_\alpha)$ como vetor âncora, utiliza-se a Equação 3.2 para calcular o vetor de distâncias;
- Utiliza-se o vetor de distâncias como dado de entrada para o classificador único;
- Caso o classificador sinalize um valor de probabilidade superior ao limiar pré-fixado (0,5), considera-se que o rosto I_α , até então desconhecido, pertence à identidade A .

3.4 Metodologia de testes

Nesta seção, serão descritas as táticas empregadas para validar os métodos propostos, desde o uso de um banco de imagens de faces frontalmente recortadas, até os detalhes dos

processos utilizados para comparar a estratégia manipuladora de *embeddings* com outras formas de usar essas redes convolucionais para verificação facial.

3.4.1 Base de Dados

Para validar métodos de reconhecimento facial, muitos pesquisadores dedicaram grandes esforços para coletar imagens de faces humanas e as armazená-las em grandes bancos de dados. A primeira tentativa internacionalmente bem-sucedida resultou na criação do LFW (*Labeled Faces In The Wild*) (HUANG *et al.*, 2008), um conjunto de dados que contém milhares de celebridades. Este conjunto de dados foi utilizado como referência para os primeiros métodos de verificação facial construídos sobre *embeddings* discriminativos, como a FaceNet. Com o passar do tempo, sentiu-se a necessidade de colocar os algoritmos à prova em cenários mais realistas e desafiadores; tal necessidade foi parcialmente atendida pelo grande conjunto de dados MegaFace, contendo 1 milhão de imagens, que teria tido mais repercussão não fosse o problema ético da coleta de dados (MILLER *et al.*, 2015); em seguida, o MS-CELEB-1M (GUO *et al.*, 2016), com 10 milhões de imagens, ganhou espaço nas pesquisas, sofrendo, no entanto, dos mesmos problemas éticos do MegaFace, em virtude da coleta de faces humanas sem consentimento dos indivíduos fotografados. Além desses conjuntos de dados, há também a série de bancos de imagens levantados pela IARPA Janus (KLARE *et al.*, 2015b; WHITELAM *et al.*, 2017; MAZE *et al.*, 2018), que buscam submeter os algoritmos de reconhecimento facial a cenários bastante severos, envolvendo vídeos, variações de pose, iluminação e expressão. Vale citar o conjunto de dados CFP (*Celebrities In Frontal Profile*) (SENGUPTA *et al.*, 2016), que visa comparar o desempenho de algoritmos em faces frontais e de perfil; por fim, um banco bastante utilizado é o CASIA-WebFace (YI *et al.*, 2014), dotado de cerca de 10 mil indivíduos, cada um com múltiplas imagens.

Como o objetivo do trabalho era simular um ambiente controlado, no qual houvesse moderada abundância de exemplos de face por pessoa, além de modesta variação de pose, iluminação e ângulo, utilizou-se o conjunto de dados publicado no mesmo trabalho da Proposta 1, chamado OFPF (*Open Famous People Faces*) (ROMERO; Rocha Neto, 2022), que se encontra pública e gratuitamente disponível no Kaggle ¹. Este conjunto de dados, cuja produção contou com a participação dos autores desta dissertação, foi montado mediante *web scraping*, em que se pesquisou exaustivamente na *internet* por imagens de pessoas famosas, aplicando ao conteúdo encontrado um algoritmo de detecção de objetos chamado SSD (*Single-Shot Detection*) (LIU

¹ <https://www.kaggle.com/datasets/yveslr/open-famous-people-faces>

et al., 2016), utilizando-o na extração de faces; após a detecção e armazenamento das faces extraídas, utilizou-se um algoritmo de mapeamento de *landmarks* faciais, para encontrar a posição de órgãos da face, como olhos, nariz e boca, a fim de realizar um recorte centralizado no rosto, excluindo regiões que poderiam carregar informação pouco relevante. Ao final, o conjunto de dados, com faces alinhadas e recortadas, atingiu o número de 258 usuários, com um total de 3024 imagens, sendo o usuário menos frequente portador de apenas 6 imagens, e o mais populoso possuindo 29. Há imagens em preto e branco, bem como uma grande variedade de qualidade e tamanho entre as imagens.

Para demonstrar que a Proposta 2, desenvolvida nesta dissertação, pode ser utilizada para validar identidades não contidas no conjunto de dados usado no treinamento (OFPF), utilizou-se o conjunto de dados *Celebrities in Frontal-Profile* (SENGUPTA *et al.*, 2016), o qual possui 500 identidades, cada qual com 10 imagens frontais e 4 imagens de perfil. Excluindo as identidades que aparecem no conjunto de dados OFPF, sobram 442 identidades, das quais se tomaram 4 imagens para uso de referência (comparações) e 6 para validação, por identidade da base.

3.4.2 *Redes Convolucionais Extratoras*

Os métodos comparados são todos de manipulação de *embeddings*, e para isso foi necessário escolher redes convolucionais para gerá-los. Escolheu-se a FaceNet, que produz *embeddings* de tamanho 128; a ArcFace, que produz *embeddings* de tamanho 512; e a GhostFaceNet (versão 1.2), que também produz *embeddings* de tamanho 512. O espaço de características da FaceNet é um espaço euclidiano, enquanto o das redes ArcFace e GhostFaceNet é um espaço angular.

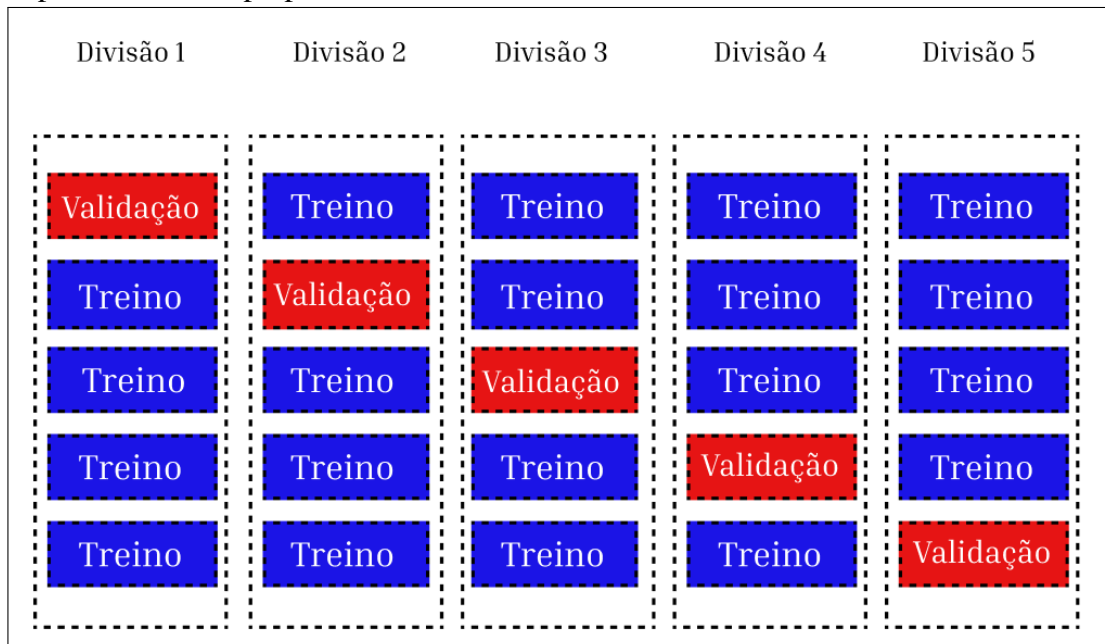
3.4.3 *Treinamento*

Utilizando as 258 classes do conjunto de dados OFPF, foram treinados classificadores para a Proposta 1 e um classificador único para a Proposta 2, cujas amostras foram construídas de acordo com a técnica descrita nas seções acima.

O conjunto de dados de treinamento e teste foi aleatoriamente obtido a partir da divisão do conjunto total em 5 rodadas distintas de partição. Sendo assim, foram gerados 5 conjuntos de treinamento e 5 conjuntos de teste. Depois de obter os cinco conjuntos de treinamento e teste, foram treinados classificadores em 5 rodadas, e calculou-se a média das performances.

Em cada rodada, fez-se uma busca de hiperparâmetros utilizando *Grid-Search* no conjunto de treinamento da rodada.

Figura 23 – Esquema de divisão do mesmo conjunto de treinamento em 5 rodadas para busca de hiperparâmetros.



Fonte: Elaborado pelo autor

Na Figura 23, está ilustrada a divisão dos conjuntos de treino em 5 partes, sempre uma delas sendo separada para validação; ao final de 5 treinamentos com os pedaços azuis e validações com os pedaços vermelhos, é calculada a média das métricas associadas a uma combinação específica de hiperparâmetros. Isso é feito para todas as combinações. Por fim, considerou-se como critério de desempenho o valor da área da curva ROC (AUC). Os pontos dessa curva ROC são, na verdade, a média ponto a ponto dos gráficos de curva ROC calculados para cada identidade individualmente. Ou seja, para cada combinação de hiperparâmetros, todas as identidades da base foram avaliadas de forma independente das outras, cada qual tendo suas métricas calculadas, incluindo a AUC da curva ROC; o que definiu o desempenho de uma combinação de hiperparâmetros foi a AUC da curva ROC média, calculada como agregação ponto a ponto das curvas ROC individuais.

Tabela 1 – Valores de hiperparâmetros predefinidos para a busca na SVM

Hiperparâmetro	Valores	Kernel
C	$[2^0 \quad 2^{0.75} \quad \dots \quad 2^{7.5}]$	Linear & RBF
Gamma	$\left[\frac{1}{2^0} \quad \frac{1}{2^{0.5}} \quad \dots \quad \frac{1}{2^{4.5}} \right]$	RBF

Fonte: Elaborado pelo autor

O método de busca utilizado para o classificador deste trabalho foi a busca em grade (*grid-search*), que consiste em montar todas as combinações possíveis de hiperparâmetros com

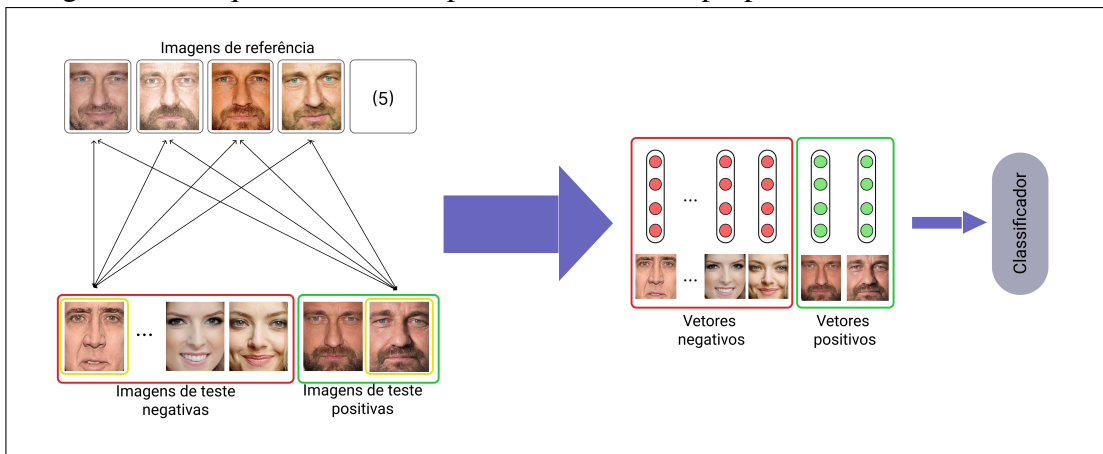
base em uma lista de valores predefinidos pelo projetista. Os valores utilizados estão listados na Tabela 1. A SVM possui um parâmetro C , que define o grau de rigidez do modelo, e este parâmetro é comum aos *kernels* linear e RBF do modelo; já o parâmetro *Gamma* é exclusivo do *kernel* RBF e controla o alcance de influência de cada ponto do conjunto de treinamento no espaço de características gerado.

3.4.4 Testes

Para os testes, foi adotado o protocolo individual², em que cada classe é testada isoladamente, sendo considerados como exemplos negativos todas as amostras que não pertencem à classe. Assim, tem-se um grande desbalanceamento de amostras entre as classes positiva e negativa, reproduzindo um cenário mais semelhante ao do mundo real.

Sendo assim, considerando que o número de imagens no conjunto de teste é N_T , cada classe será validada com todas essas imagens. Se, em meio a essas N_T imagens de teste, há N_P imagens de um determinado indivíduo, ele terá N_P amostras de teste que devem ser classificadas como positivas, e $N_T - N_P$ amostras que se espera que o modelo infira como negativas. Das N_P amostras, as que forem incorretamente classificadas como negativas, ou seja, não pertencentes ao indivíduo, serão computadas nos casos de falso negativo; e das $N_T - N_P$, as que forem incorretamente classificadas como positivas, serão computadas como falsos positivos.

Figura 24 – Esquema de testes aplicável a ambas as propostas



Fonte: Elaborado pelo autor

A Figura 24 ilustra visualmente tal processo. Vale notar que, neste trabalho, adotou-se um N_T bem maior que N_P , para simular um ambiente real. Dessa maneira, todas as métricas são calculadas individualmente para cada classe na fase de testes, tomada como oponente de todas

² Ver, no capítulo de Fundamentação Teórica, a seção Protocolo de Experimento

as outras classes, e ao final se calcula uma média das métricas de cada classe para compor as pontuações finais.

No intuito de promover maior clareza sobre os resultados, foram utilizadas as curvas de precisão e recall³. Essas curvas foram construídas, durante os testes, mediante a média ponto a ponto, das curvas de cada classe, semelhantemente ao que foi feito no procedimento de validação cruzada. A escolha da curva de precisão e recall (curva PR), em detrimento da mais conhecida curva ROC, pode ser explicada pelo fato de que a área abaixo da curva PR, cujo valor representa a eficiência global do modelo, tem a característica de diminuir drasticamente quando há muitos falsos positivos; isso acaba expondo aspectos do desempenho do modelo que podem passar despercebidos pela análise da curva ROC. No entanto, para a realização da validação cruzada, escolheu-se a combinação de hiperparâmetros que resultasse na maior área da curva ROC média descrita acima. Empiricamente, observou-se que encontrar os hiperparâmetros que maximizam a área desta curva levou a resultados mais consistentes, não só no método proposto, mas em todos os outros métodos que foram utilizados como bases experimentais comparativas. Contudo, isso não exclui a contribuição da curva PR para uma melhor compreensão da performance, o que será explicado mais claramente na discussão dos resultados.

3.4.5 Estratégias de Base para Comparação

Tendo em vista a validação da estratégia de manipulação de *embeddings* proposta no presente trabalho, escolheram-se outras estratégias de manipulação (dentre as quais está incluída a forma tradicional de utilizar modelos de *one-shot learning*, que consiste em aplicar um limiar de distância simples) para servir de base para comparações.

Proposta 1 é uma das propostas, publicada em um trabalho recente (ROMERO; Rocha Neto, 2022), e consiste em treinar uma SVM para cada classe do conjunto de dados.

Proposta 2 é uma das propostas, e consiste na criação de um classificador SVM único utilizando vetores de distâncias.

Limiar Global é o método que consiste na aplicação de uma regra de limiar de distância para diferenciar se dois *embeddings* pertencem à mesma identidade. Como esse método originalmente não tem uma formulação metodológica idêntica à apresentada neste trabalho, foi preciso aplicar-lhe os mesmos procedimentos de validação empregados para as propostas desta dissertação, para assegurar a integridade da comparação. Tomando como exemplo a FaceNet: essa rede foi testada utilizando 100 mil x 100 mil pares de imagens com regra de limiar de

³ Ver o capítulo de Fundamentação Teórica

distância, e a partir dos resultados foram calculadas as métricas; no presente trabalho, ao validar a regra de limiar, utilizou-se a abordagem individual, onde cada classe é considerada isoladamente.

VetoresEmbeddings é um método que é utilizado como base de comparação pelo artigo propositor do método VetoresIndividuais. As duas abordagens são similares, com a diferença de que, no VetoresEmbeddings, as amostras positivas são dos próprios *embeddings* da classe, e as negativas são *embeddings* de outras classes escolhidas aleatoriamente.

DeltaVetores, na seção de resultados, é um método que foi inspirado na proposta do trabalho da DeepFace (TAIGMAN *et al.*, 2014). A estratégia **DeltaVetores** consiste em treinar um classificador global, ou seja, que pode realizar verificação facial em todas as classes, usando como amostras positivas vetores de diferença absoluta, que consistem na subtração dos *embeddings* de uma classe em relação à média ponto a ponto dos *embeddings* desta classe; as amostras negativas são um vetor de diferença absoluta que é resultado da comparação entre *embeddings* de várias classes e a média ponto a ponto dos vetores *embeddings* da classe em questão. Após levantar amostras positivas e negativas por classe, todas são reunidas para treinar um classificador só. Essa estratégia não é idêntica à proposta pelo artigo da DeepFace, pois possui, em relação a ela, certas diferenças fundamentais:

- Os vetores são de diferença absoluta, enquanto no trabalho da DeepFace é utilizada a diferença χ^2 . Essa adaptação foi feita, pois, no caso da DeepFace, os vetores *embeddings* não possuem valores negativos, ao passo que os vetores *embeddings* das redes aqui testadas possuem tais valores.
- Não foram utilizados todos os vetores de diferença gerados por cada classe. Em virtude de um limite de recursos computacionais, aplicou-se uma estratégia de agrupamento para escolher, do conjunto de vetores diferença, os mais representativos dos dados, a fim de que o classificador treinado apresente um desempenho similar ao classificador que seria obtido caso se treinasse com todo o conjunto. Essa semelhança parece ser reforçada pelo fato de que a SVM não precisa de muitas amostras, desde que as poucas amostras utilizadas para o treinamento representem bem a fronteira de decisão. O agrupamento foi realizado da seguinte forma: os vetores diferença de cada classe, tanto os que constituem amostras positivas, como os que constituem as negativas, foram agrupados, por distância, em cinco grupos, dos quais foram escolhidos, aleatoriamente, 80 vetores de amostras positivas e 184 de amostras negativas, obedecendo uma proporção de 1 positivo para 2,3 negativos.

A proporção de 1 positivo para cada 2,3 negativos foi utilizada em todas as estratégias de manipulação apresentadas na seção de resultados, para manter a coerência dos testes.

4 EXPERIMENTOS E DISCUSSÃO DE RESULTADOS

Após a devida condução, realizada neste trabalho, passando por aspectos teóricos que fundamentam esta dissertação, bem como pela descrição detalhada da tecnologia que se decidiu propor para efetuar verificação facial, esta seção apresenta os experimentos realizados e a discussão dos resultados obtidos.

Cada uma das abordagens aqui comparadas foi executada utilizando *embeddings* de redes neurais convolucionais diferentes, dentre as quais se pode destacar a FaceNet e a ArcFace. Esta variedade de extratores enriquece a análise de resultados, e pode fornecer intuições acerca dos pontos fortes de cada estratégia de manipulação desses *embeddings*. Em cada estratégia validada, desenharam-se as curvas ROC e PR médias, para facilitar o exame do desempenho real.

4.1 Resultados dos Experimentos

Tabela 2 – Resultados com *Kernel Linear*, a um limiar de probabilidade igual a 0,5

Extrator	Estratégia	FP		FN		Precisão		Recall	
		Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
FaceNet	VetoresEmbeddings	1,27	0,86	0,49	2,84	34,47	18,68	99,50	2,84
	Proposta 1	0,19	0,26	0,98	3,90	79,02	19,71	99,01	3,90
	Proposta 2	0,41	0,47	1,90	7,61	61,54	23,47	98,09	7,61
	DeltaVetores	0,29	0,32	1,01	4,15	67,17	21,80	98,98	4,15
ArcFace	VetoresEmbeddings	1,01	1,07	0,58	2,59	45,27	25,01	99,41	2,59
	Proposta 1	0,19	0,30	0,91	3,33	80,10	20,26	99,08	3,33
	Proposta 2	0,32	0,36	1,31	5,52	67,51	24,07	98,68	5,52
	DeltaVetores	0,81	1,01	3,91	8,52	56,50	29,85	96,08	8,52
GhostFace	VetoresEmbeddings	2,08	1,80	3,69	6,04	27,83	18,97	96,30	6,04
	Proposta 1	1,91	1,75	8,98	10,69	32,38	22,47	91,01	10,69
	Proposta 2	4,22	4,95	12,78	18,36	28,15	26,39	87,21	18,36
	Delta Vetores	1,74	14,51	9,7	10,80	29,33	22,59	90,29	10,80

Fonte: Elaborado pelo autor

Na Tabela 2, são mostrados os resultados dos testes de cada estratégia utilizando como classificador a SVM, utilizando o *kernel* linear. Nessa tabela, são apresentadas as métricas que cada estratégia obteve em diferentes modelos *embedders*. Para a FaceNet, utilizou-se a distância euclidiana como função para gerar os vetores de distância, pois essa rede foi originalmente treinada com uma função de custo que induzia os parâmetros internos do modelo a separar espacialmente os *embeddings* de acordo com uma distância euclidiana no espaço de características; já para a GhostFace e para a ArcFace, utilizou-se a distância cosseno, refletindo o treinamento apresentado pelos trabalhos que publicaram essas redes. Aqui, cabe destacar que a

rede GhostFace apresentou baixo desempenho em todas as estratégias de manipulação, quando comparada às demais redes, indicando que seus *embeddings* foram pouco discriminativos para o conjunto de dados utilizado. Os testes realizados utilizando *embeddings* da GhostFace, uma rede recentemente proposta, não geraram bons classificadores.

Tabela 3 – Resultados com *Kernel RBF*, a um limiar de probabilidade igual a 0,5

Extrator	Estratégia	FP		FN		Precisão		Recall	
		Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
FaceNet	VetoresEmbeddings	0,20	0,25	1,11	4,68	76,95	20,06	98,88	4,68
	Proposta 1	0,20	0,25	1,30	5,43	77,08	20,78	98,69	5,43
	Proposta 2	0,36	0,44	1,89	7,52	63,79	23,18	98,10	7,52
	DeltaVetores	0,29	0,31	1,01	3,96	68,14	21,83	98,98	3,96
ArcFace	VetoresEmbeddings	0,03	0,08	6,41	8,29	95,07	9,53	93,58	8,29
	Proposta 1	0,18	0,25	0,84	3,65	79,59	19,70	99,16	3,65
	Proposta 2	0,28	0,32	1,23	5,23	69,90	23,35	98,76	5,23
	DeltaVetores	0,44	0,69	4,25	9,32	69,48	28,87	95,74	9,32
GhostFace	VetoresEmbeddings	0,0	0,0	99,89	1,09	0,38	4,11	0,10	0,10
	Proposta 1	1,70	1,59	12,43	14,42	32,69	22,45	87,56	14,42
	Proposta 2	3,21	3,93	12,40	17,17	33,46	29,07	87,60	17,17
	Delta Vetores	0,00	0,00	99,85	1,13	0,46	3,49	0,14	1,13

Fonte: Elaborado pelo autor

Alguns, inclusive, sofreram um viés muito forte para a classe majoritária, levando o modelo a classificar todas as amostras de validação como negativas, como está apresentado na Tabela 3. Um estudo recente (ARDIAWAN; NEGARARA, 2024) reportou um baixo desempenho da GhostFaceNet, quando colocada lado a lado com a FaceNet e a VGGFace. É possível que particularidades do conjunto de dados interfiram na eficiência da rede durante a extração de características e, portanto, prejudiquem a discriminação no espaço de características. No entanto, a constatação mais conhecida é de que a GhostFaceNet apresenta desempenho similar ao da FaceNet no conjunto de dados LFW; além disso, a GhostFaceNet possui uma arquitetura mais leve que a ArcFace, o que pode prejudicar a generalização para outros conjuntos de dados.

Ainda na Tabela 3, estão os resultados que os classificadores atingiram usando o *kernel RBF*. Nota-se que, utilizando a FaceNet como extrator de características, na tabela do *kernel RBF*, as taxas de falso positivo e falso negativo são similares entre as três estratégias de manipulação; contudo, apesar dessa semelhança, os métodos globais, ou seja, Proposta 2 e DeltaVetores, apresentaram uma precisão bastante inferior às outras. Isso acontece porque a precisão é muito sensível a falsos positivos quando a classe minoritária do conjunto de dados é a positiva. Sendo assim, ligeiras mudanças nas taxas podem afetar drasticamente a precisão; por essa razão, em um cenário em que deve haver bastante robustez contra falsos positivos, a precisão é um excelente

indicador.

Com relação à ArcFace, observa-se um padrão similar. No entanto, vale notar que, para os *embeddings* dessa rede, o método VetoresEmbeddings apresentou uma taxa de falso positivo muito baixa, e um volume maior de falsos negativos, com o *kernel* RBF; isso provavelmente sucedeu porque as Máquinas de Vetores Suporte treinadas para cada classe do conjunto de dados ficaram muito intolerantes a erros da classe positiva, aumentando a quantidade de erros da classe negativa. A Proposta 1, no *kernel* RBF, mostrou-se bastante competitiva, tendo desempenho superior em falsos positivos e precisão em relação a todos os métodos, quando testados com *embeddings* da FaceNet e da GhostFace.

Tabela 4 – Resultados da Estratégia de Limiar Global

Modelo	FP		FN		Precisão		Recall	
	Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
FaceNet	0,21	0,58	14,76	12,91	84,44	19,38	85,23	12,91
ArcFace	0,35	0,70	13,58	12,08	78,15	24,51	86,41	12,08
GhostFace	3,69	3,05	24,31	15,83	24,94	22,55	75,68	15,83

Fonte: Elaborado pelo autor

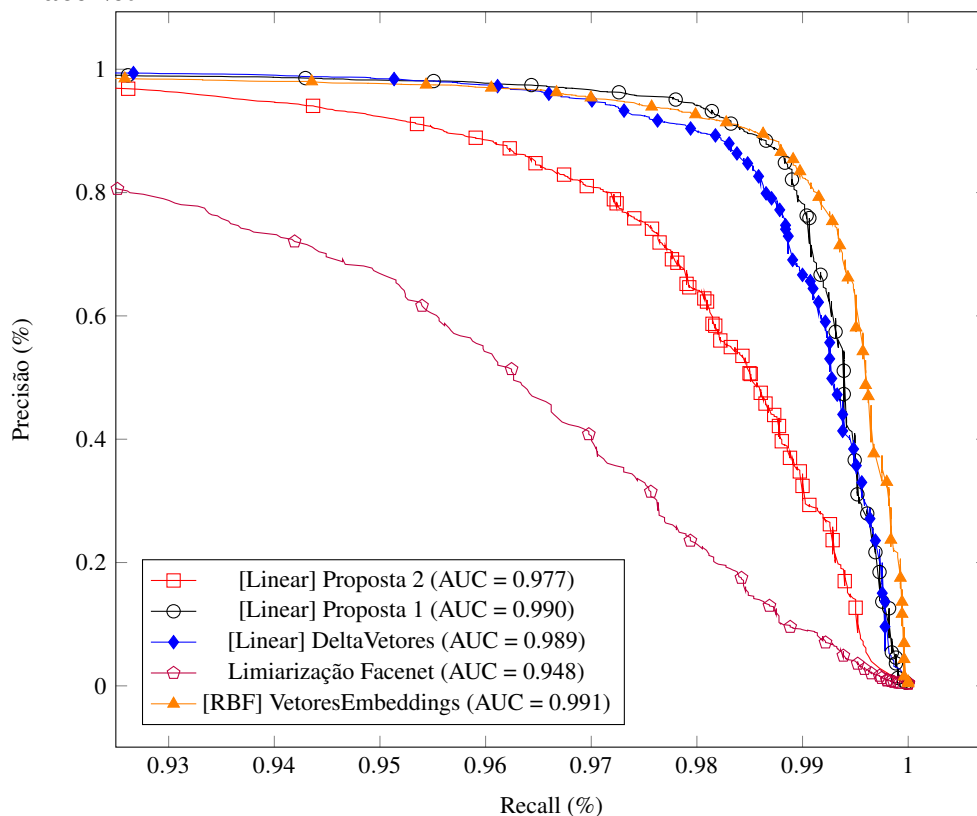
Na Tabela 4, mostram-se resultados da estratégia de limiarização com os *embeddings* das três redes convolucionais, em que se pode notar um destaque de desempenho das redes FaceNet e ArcFace em relação à GhostFaceNet, o que reforça a ideia de que tal modelo parece ter uma menor capacidade de produzir um espaço de características discriminativo.

Das tabelas referidas, pode-se concluir que os métodos que produzem um classificador por indivíduo na base de dados tendem a atingir desempenhos mais elevados. Isso ocorre, pois, no caso dos métodos globais, o classificador único precisa ocupar-se de achar uma fronteira de decisão que segregue corretamente padrões positivos e negativos oriundos de diversas identidades. Enquanto isso, os classificadores dos métodos individuais, podendo concentrar-se em padrões de vetores de distância associados a uma só identidade, acabam por realizar um melhor aprendizado da geometria desses padrões, elevando a performance. Em contrapartida, os métodos de criação individual de classificadores carregam a desvantagem de que, sempre que uma nova identidade é cadastrada no sistema de verificação facial, será preciso realizar um novo classificador com os padrões obtidos dessa identidade. Ao passo que, nesse mesmo contexto, os classificadores únicos gerados por métodos globais podem ser aplicados satisfatoriamente, sem necessidade de serem retreinados: quando, após o treinamento do classificador único, novas identidades são acrescentadas ao sistema, nada precisa ser feito, basta usar o mesmo classificador. O alcance

dessa afirmação será mais detalhadamente discutido mais adiante, ao serem abordados os testes de classificadores globais em identidades não vistas.

Os classificadores SVM que geraram os resultados das tabelas possuem um limiar interno que estabelece que, a partir de um determinado valor de probabilidade, considera-se que a amostra avaliada é positiva. Em virtude disso, é possível que um mesmo modelo pareça, ora superior, ora inferior em relação aos demais, a depender do limiar de probabilidade que se utilize para julgar as amostras. Diante disso, é preciso encontrar uma métrica de avaliação de classificadores que quantifique, em valor numérico, o desempenho global do classificador, já considerando os diversos limiares de probabilidade possíveis. Para isso, costuma-se utilizar a área de cobertura das curvas ROC e das curvas de Precisão-Recall.

Figura 25 – Curvas PR médias de estratégias de manipulação, utilizando vetores da FaceNet

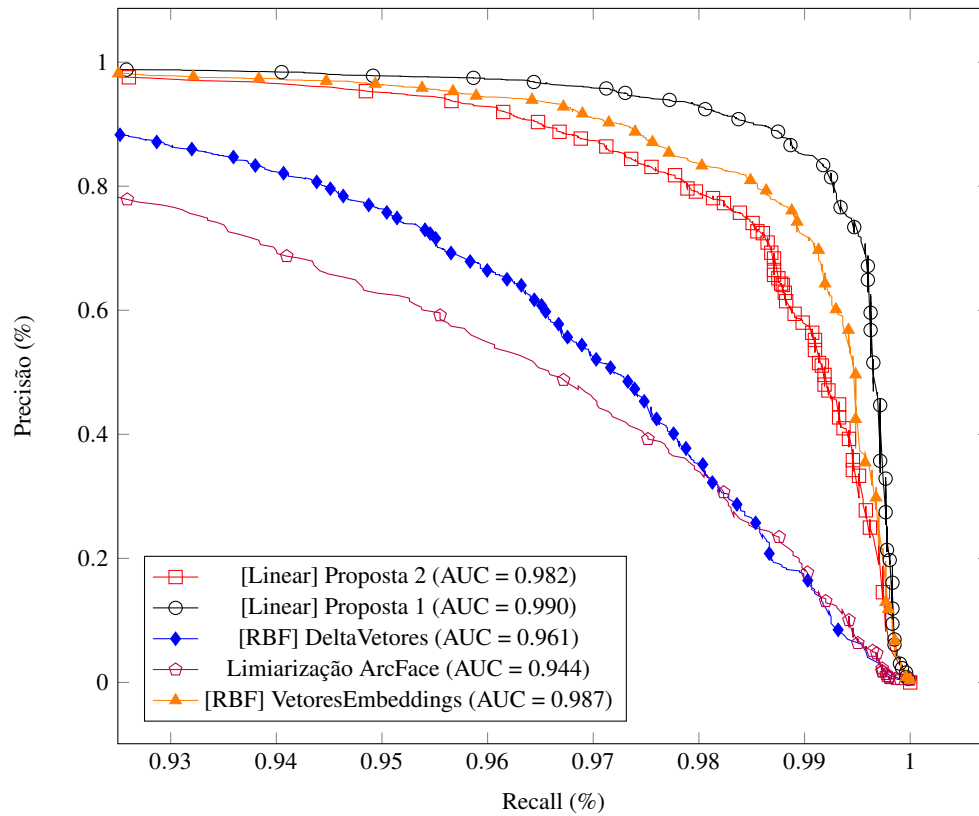


Fonte: Elaborado pelo autor

Nas Figuras 25, 26 e 27, estão exibidos os gráficos das curvas de Precisão-Recall, calculadas para cada método, com a legenda, no canto inferior esquerdo, mostrando o valor das áreas abaixo das curvas.

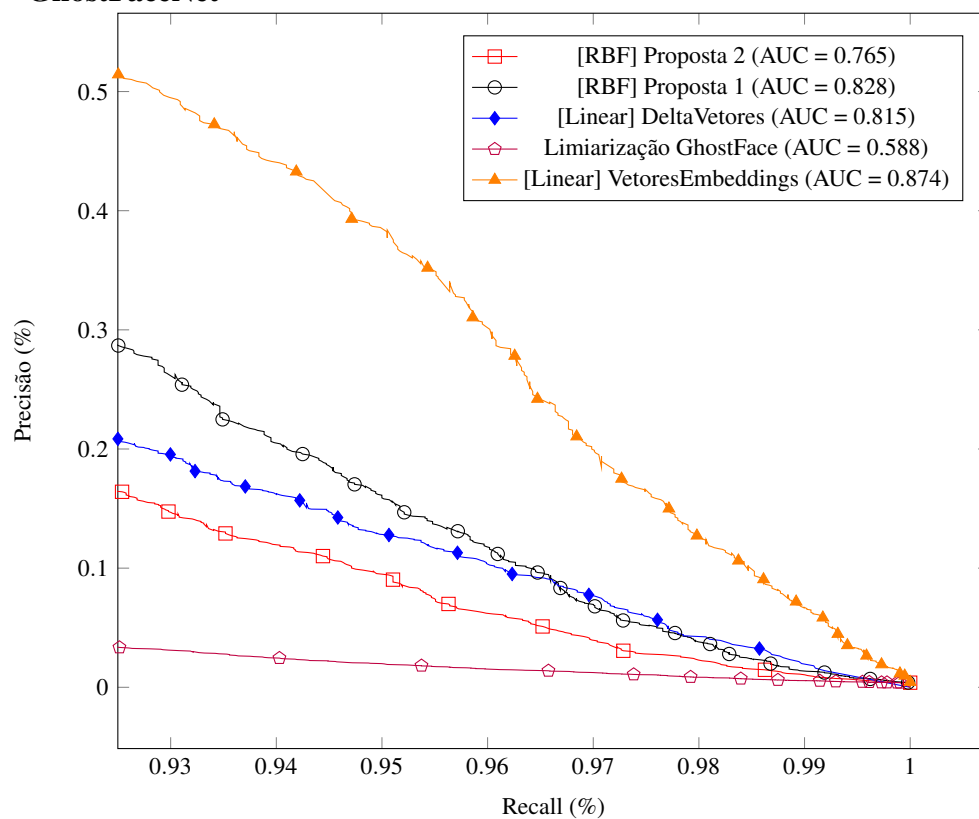
A princípio, calcularam-se os gráficos das curvas ROC de cada método, a fim de utilizar o valor da área abaixo de cada curva como uma quantificação numérica de seus respectivos

Figura 26 – Curvas PR médias de estratégias de manipulação, utilizando vetores da **ArcFace**



Fonte: Elaborado pelo autor

Figura 27 – Curvas PR médias de estratégias de manipulação, utilizando vetores da **GhostFaceNet**



Fonte: Elaborado pelo autor

métodos. No entanto, a diferença de valor das áreas das curvas ROC foi bastante pequena de um método para outro. Isto provavelmente aconteceu porque a curva ROC pode não ser a forma mais apropriada de representar o desempenho global de um classificador em contexto de classes extremamente desbalanceadas, como é o caso abordado pelo presente estudo. Diante disso, fez-se o cálculo das curvas de Precisão-Recall, que tendem a refletir de forma mais transparente o comportamento de classificadores que atuam nesse contexto. Esse tipo de curva coloca a métrica de precisão no eixo Y, punindo a pontuação dos classificadores que não conseguem manter um nível de precisão alto enquanto elevam a taxa de verdadeiros positivos (eixo X); essa deficiência, quando ocorre em um classificador, sinaliza sua maior fragilidade para falsos positivos, sobretudo quando o conjunto de dados é extremamente desbalanceado, e a classe minoritária é a positiva.

Considerando que cada método gerou classificadores de *kernel* linear e RBF, tomou-se, em cada método, o *kernel* que gerou melhor desempenho global para ser exibido nesses gráficos. Note-se que, nas Figuras 25, 26 e 27, os métodos de geração individual de classificadores obtiveram desempenho superior, com um empate técnico entre a Proposta 1 e o método VetoresEmbeddings, no que toca aos *embeddings* da FaceNet, e com uma ligeira superioridade da Proposta 1 quando ambos foram testados com *embeddings* da ArcFace. Utilizando, porém, os da GhostFaceNet, o método VetoresEmbeddings se sobressaiu significativamente, em comparação à Proposta 1.

Comparando a Proposta 2 e o método DeltaVetores, pode-se observar que o classificador do DeltaVetores se mostrou mais certo que o da Proposta 2, para os gráficos de Precisão-Recall da FaceNet e da GhostFaceNet. Contudo, no gráfico da ArcFace, prevalece a Proposta 2. Isso é um indício de que métodos que produzem vetores de distâncias, ao invés de utilizar o *embedding* inteiro, podem ter uma afinidade maior com a ArcFace, já que, tanto a Proposta 2, como a Proposta 1, estão entre os métodos que obtiveram as maiores AUC na Figura 26. Isso possivelmente se deve ao fato de que a ArcFace produz uma grande separação angular entre *embeddings* de identidades distintas, assim como um estreitamento angular entre *embeddings* da mesma identidade, de modo que cada elemento do vetor de distâncias permaneça em uma faixa de valores com pouca variação, favorecendo a separabilidade entre classes.

Além disso, como o método DeltaVetores utiliza diferenças absolutas para montar o vetor de diferenças, as amostras do classificador podem não representar plenamente todas as nuances da separação angular do espaço de características da ArcFace. Semelhante fenômeno não parece ocorrer no caso da FaceNet, que produz um espaço Euclidiano, em que o vetor de diferenças

ponto a ponto representa mais precisamente a distância entre os *embeddings*, o que justifica o melhor desempenho do método DeltaVetores com os *embeddings* desta rede. Ademais, os *embeddings* da ArcFace possuem quatro vezes mais elementos que os da FaceNet, o que pode ter influenciado negativamente sobre os métodos que utilizam, durante o treinamento, amostras que têm o mesmo tamanho dos *embeddings* (VetoresEmbeddings e DeltaVetores); evidência disso é que esses métodos, no gráfico da ArcFace, sofreram uma queda de desempenho em relação ao gráfico da FaceNet.

Com relação à Figura 27, observa-se uma queda de desempenho de todos os métodos em relação aos outros gráficos de Precisão-Recall. Semelhantes quedas de desempenho em relação a outros modelos já foram reportadas em um artigo recente, já mencionado neste capítulo, o qual aponta uma acurácia de 75% para a GhostFaceNet e 97,4% para a FaceNet, em um conjunto de imagens montado pelos próprios autores. É possível que particularidades dos conjuntos de dados utilizados, como recorte, tamanho das imagens e outros fatores dificultem a extração de características por essa rede convolucional. Isso pode ter afetado, no gráfico da GhostFaceNet, sobretudo os métodos baseados em vetores de distâncias (Proposta 1 e Proposta 2): representações aparentemente pouco discriminativas no espaço de características podem ter impedido que o vetor de distâncias reflita parte relevante da informação, sendo necessário utilizar modelos mais pesados, e. g., DeltaVetores, para atingir resultados mais satisfatórios.

Ainda analisando a GhostFaceNet, observa-se que, para o *kernel* RBF, os métodos DeltaVetores e VetoresEmbeddings, que utilizam, ambos, amostras com a mesma extensão dos *embeddings* da rede, tiveram um viés em favor da classe negativa, o que se mostra na Tabela 3. Isso ocorreu porque todas as métricas exibidas nessa tabela foram geradas para um limiar de probabilidade de 0,5; nesse limiar, os classificadores ainda consideram que quase tudo é negativo, e o comportamento real do classificador é melhor caracterizado na Figura 27, em que foram testados diversos limiares. Isso é um indício de que a SVM teve dificuldades em encontrar a solução para o problema, possivelmente devido a sobreposições de classes após a transformação dos dados pelo RBF, o que se uniu ao desbalanceamento de classes que é próprio do problema de verificação. Outra hipótese é de que a faixa de pesquisa para os hiperparâmetros C e γ pode não ter sido suficientemente extensa.

4.1.1 Espaço ocupado em memória pelos modelos

Após o treinamento dos modelos, foram salvos, em formato *pickle*, os melhores de cada método para serem testados em identidades não vistas, como será mostrado na subseção seguinte. Ao analisar o tamanho dos modelos, observou-se que os classificadores gerados pelo método DeltaVetores ocuparam significativamente mais memória que os oriundos da Proposta 2.

Quando foram treinados usando *embeddings* da GhostFace, o classificador do DeltaVetores, com *kernel* linear, se mostrou quase três vezes mais pesado que o da Proposta 2, que ocupou 200 kilobytes; no caso da ArcFace, o tamanho foi cerca de vinte vezes o do classificador da Proposta 2, o qual ocupou cerca de 20 kilobytes de memória; no caso da FaceNet, o classificador da estratégia DeltaVetores ocupa cerca de três a quatro vezes mais espaço.

Assim, a Proposta 2 está em vantagem no consumo de memória em relação ao método DeltaVetores em virtude do tamanho das amostras, pois as amostras de treinamento do classificador utilizado na Proposta 2 desta dissertação não dependem do tamanho do *embedding* da rede convolucional utilizada para extrair características das imagens. O tamanho de cada amostra é sempre $K - 1$, onde K é o número da classe menos populada do conjunto de dados, e os elementos do vetor de amostra são valores de distância, euclidiana ou cosseno; ao passo que a estratégia DeltaVetores utiliza amostras com a mesma extensão dos *embeddings* da rede convolucional, tornando-a mais pesada e fazendo-a demandar mais processamento durante o treino, a ponto de ter sido necessário realizar amostragem via clusterização para que o treinamento fosse concluído. Dessa forma, o tamanho reduzido das amostras da Proposta 2 torna igualmente menores os vetores suporte que foram armazenados no classificador, o que contribuiu para um baixo consumo de memória, tornando essa solução bastante vantajosa para sistemas que tenham limitação de recursos de memória.

Quando se compara o tamanho dos modelos da Proposta 1 com os do método VetoresEmbeddings, descobre-se que, para o *kernel* linear, a Proposta 1 produz classificadores que ocupam cerca de 1,5 kilobytes para a FaceNet e para a ArcFace, e 2 kilobytes para a GhostFace; em contrapartida, os da estratégia VetoresEmbeddings possuem tamanhos de 15 kilobytes para a FaceNet, e cerca de 70 kilobytes para ArcFace e GhostFace. Com o *kernel* RBF também há ganhos significativos.

Tabela 5 – Resultados da Estratégia Proposta no Conjunto de Dados CFP

Extrator	Estratégia	FP		FN		Precisão		Recall	
		Média	Desvio	Média	Desvio	Média	Desvio	Média	Desvio
FaceNet	Proposta 2	0,019	0,03	3,74	9,66	93,95	9,90	96,25	9,66
	DeltaVetores	0,009	0,020	3,43	7,16	96,84	6,58	96,53	7,16
ArcFace	Proposta 2	0,02	0,04	15,55	20,12	91,78	12,09	84,44	20,20
	DeltaVetores	1,58	1,98	12,59	15,71	39,64	31,21	87,40	15,71
GhostFaceNet	Proposta 2	1,48	1,32	46,07	27,96	11,04	8,42	53,92	27,96
	DeltaVetores	8,57	0,79	21,95	1,57	24,58	1,21	78,04	1,57

Fonte: Elaborado pelo autor

4.1.2 Desempenho de métodos globais em base de dados com identidades não vistas

Para testar a generalização e estabilidade dos métodos globais (Proposta 2 e DeltaVetores), utilizou-se o conjunto de dados CFP (*Celebrities in Frontal-Profile*), escolhendo nesse conjunto 442 identidades que não aparecem no conjunto de dados original, ou seja, o conjunto OFPF (*Open Famous People Faces*), que contém 258 identidades. Realizou-se a verificação facial de todas as 442 identidades do conjunto não visto pelo treinamento, sem efetuar nenhum tipo de retreinamento, ou ajuste fino. Utilizou-se a mesma metodologia de testes empregada na avaliação do classificador que gerou os resultados das seções acima.

Ao utilizar os *embeddings* da FaceNet, ambos os classificadores treinados com 258 identidades tiveram um alto desempenho no conjunto de dados de prova, apresentando os resultados descritos na Tabela 5. Já ao utilizar *embeddings* da ArcFace, ambos os classificadores apresentaram uma perda de recall, em comparação aos resultados obtidos com os *embeddings* da FaceNet. Entretanto, o classificador gerado pela Proposta 2 manteve uma precisão elevada, enquanto a precisão do DeltaVetores caiu drasticamente.

No caso da ArcFace, a Proposta 2 atingiu uma AUC de 0,95 na curva de precisão e recall, enquanto o método **DeltaVetores** só atingiu o valor de 0,80. A proposta global deste trabalho apresentou taxas semelhantes de precisão utilizando a FaceNet e a ArcFace como redes extratoras, o que indica certa robustez do método.

Com relação à GhostFaceNet, o valor da AUC obtido pelo DeltaVetores foi de aproximadamente 0,45, enquanto o da Proposta 2 foi de aproximadamente 0,35. Nota-se que houve uma marcada queda de desempenho em ambas as estratégias, o que se pode notar, inclusive, pela perda de precisão e recall exibidas na Tabela 5. No caso da Proposta 2, a perda foi maior, possivelmente devido a uma baixa discriminatividade que os *embeddings* da GhostFaceNet ofereceram para o conjunto de dados de treinamento, fazendo com que os vetores de distância

percam sua capacidade de representar com exatidão os padrões das classes.

5 CONCLUSÃO

Este trabalho propôs dois métodos para desempenhar a tarefa de verificação facial. O classificador da proposta global, após o treinamento, mostrou-se suficientemente robusto para ser utilizado em um conjunto de dados com 442 identidades desconhecidas, mantendo uma alta precisão. A técnica proposta manteve a taxa de falsos positivos baixa ao trocar da FaceNet para a ArcFace, apresentando uma perda de recall semelhante à do outro método global, apresentando, entretanto, a vantagem adicional de um baixo consumo de memória, destacando-se como solução para sistemas que operam em ambientes de recursos limitados. Já a proposta de classificadores por identidade mostrou desempenho aceitável em relação aos demais métodos, dada a simplicidade dos modelos treinados e o reduzido tamanho dos dados de entrada. Ambas as propostas demonstraram ter uma maior afinidade com os *embeddings* da ArcFace, indicando que o espaço de características gerado por essa rede pode ser mais eficiente que o da FaceNet, em virtude da separação angular imposta por sua função objetivo; em contrapartida, a extensão desses *embeddings* é um fator estratégico que deve ser ponderado antes de optar pela ArcFace em detrimento da FaceNet.

Trabalhos futuros podem evoluir na direção de treinamentos de classificadores utilizando o método proposto em conjuntos de dados mais amplos, que contenham mais imagens por usuário, a fim de aumentar a extensão dos vetores de distância utilizados para o treinamento, possivelmente melhorando o desempenho.

REFERÊNCIAS

- ALANSARI, M.; HAY, O. A.; JAVED, S.; SHOUFAN, A.; ZWEIRI, Y.; WERGHI, N. Ghostfacenets: Lightweight face recognition model from cheap operations. **IEEE Access**, IEEE, v. 11, p. 35429–35446, 2023.
- ARDIAWAN, M. I.; NEGARARA, G. P. K. A comparative analysis of facenet, vggface, and ghostfacenets face recognition algorithms for potential criminal suspect identification. **Journal of Applied Artificial Intelligence**, v. 5, n. 2, p. 34–49, 2024.
- BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. In: SPRINGER. **Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9**. [S.l.], 2006. p. 404–417.
- BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 19, n. 7, p. 711–720, 1997.
- BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.: s.n.], 1992. p. 144–152.
- BROMLEY, J.; GUYON, I.; LECUN, Y.; SÄCKINGER, E.; SHAH, R. Signature verification using a "siamese" time delay neural network. **Advances in neural information processing systems**, v. 6, 1993.
- BUCHANAN, B. G.; SMITH, R. G. Fundamentals of expert systems. **Annual review of computer science**, v. 3, n. 1, p. 23–58, 1988.
- CAHYONO, F.; WIRAWAN, W.; RACHMADI, R. F. Face recognition system using facenet algorithm for employee presence. In: IEEE. **2020 4th international conference on vocational education and training (ICOVET)**. [S.l.], 2020. p. 57–62.
- CHOPRA, S.; HADSELL, R.; LECUN, Y. Learning a similarity metric discriminatively, with application to face verification. In: IEEE. **2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)**. [S.l.], 2005. v. 1, p. 539–546.
- COHEN, L. Whodunit?—violence and the myth of fingerprints: Comment on harding. **Configurations**, Johns Hopkins University Press, v. 2, n. 2, p. 343–347, 1994.
- COLE, S. What counts for identity? the historical origins of the methodology of latent fingerprint identification. **Science in Context**, Cambridge University Press, v. 12, n. 1, p. 139–172, 1999.
- DENG, J.; GUO, J.; XUE, N.; ZAFEIRIOU, S. Arcface: Additive angular margin loss for deep face recognition. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2019. p. 4690–4699.
- DENG, J.; GUO, J.; YANG, J.; XUE, N.; KOTSIA, I.; STEFANOS, Z. **ArcFace: Additive Angular Margin Loss for Deep Face Recognition**. 2018. Disponível em: <<https://arxiv.org/abs/1801.07698>>.
- DENG, J.; ZAFERIRIOU, S. Arcface for disguised face recognition. In: **proceedings of the IEEE/CVF international conference on computer vision workshops**. [S.l.: s.n.], 2019. p. 0–0.

FEI-FEI, L.; FERGUS, R.; PERONA, P. One-shot learning of object categories. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 28, n. 4, p. 594–611, 2006.

FIRMANSYAH, A.; KUSUMASARI, T. F.; ALAM, E. N. Comparison of face recognition accuracy of arcface, facenet and facenet512 models on deepface framework. In: IEEE. **2023 International conference on computer science, information technology and engineering (ICCoSITE)**. [S.l.], 2023. p. 535–539.

FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, JSTOR, p. 1189–1232, 2001.

GABOR, D. Theory of communication. part 1: The analysis of information. **Journal of the Institution of Electrical Engineers-part III: radio and communication engineering**, IET, v. 93, n. 26, p. 429–441, 1946.

GE, W. Deep metric learning with hierarchical triplet loss. In: **Proceedings of the European conference on computer vision (ECCV)**. [S.l.: s.n.], 2018. p. 269–285.

GHAIDA, D. R.; SIEDHARTA, V. V.; HAKIM, U.; MUTIJARSA, K.; SEPTIANA, A. I.; ROSMANSYAH, Y. Svm-classified facenet and eigenface models under lighting and occlusion variations for face recognition. In: IEEE. **2024 International Seminar on Application for Technology of Information and Communication (iSemantic)**. [S.l.], 2024. p. 415–420.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

GUO, Y.; ZHANG, L.; HU, Y.; HE, X.; GAO, J. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In: SPRINGER. **Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part III 14**. [S.l.], 2016. p. 87–102.

HAN, K.; WANG, Y.; TIAN, Q.; GUO, J.; XU, C.; XU, C. Ghostnet: More features from cheap operations. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 1580–1589.

HARWOOD, B.; BG, V. K.; CARNEIRO, G.; REID, I.; DRUMMOND, T. Smart mining for deep metric learning. In: **Proceedings of the IEEE international conference on computer vision**. [S.l.: s.n.], 2017. p. 2821–2829.

HE, K.; ZHANG, X.; REN, S.; SUN, J. 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.

HERMANS, A.; BEYER, L.; LEIBE, B. In defense of the triplet loss for person re-identification. **arXiv preprint arXiv:1703.07737**, 2017.

HUANG, G. B.; LEE, H.; LEARNED-MILLER, E. Learning hierarchical representations for face verification with convolutional deep belief networks. In: IEEE. **2012 IEEE conference on computer vision and pattern recognition**. [S.l.], 2012. p. 2518–2525.

HUANG, G. B.; MATTAR, M.; BERG, T.; LEARNED-MILLER, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. In: **Workshop on faces in 'Real-Life' Images: detection, alignment, and recognition**. [S.l.: s.n.], 2008.

HUANG, Y.; WANG, Y.; TAI, Y.; LIU, X.; SHEN, P.; LI, S.; LI, J.; HUANG, F. Curricularface: adaptive curriculum learning loss for deep face recognition. In: **proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2020. p. 5901–5910.

IYPE, J. K.; SEBASTIAN, S. Comparative analysis of state-of-the-art face recognition models: Facenet, arcfacenet, and openface using image classification metrics. In: SPRINGER. **International Conference on Computational Sciences and Sustainable Technologies**. [S.l.], 2023. p. 222–234.

KIM, M.; JAIN, A. K.; LIU, X. Adafacenet: Quality adaptive margin for face recognition. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2022. p. 18750–18759.

KLARE, B. F.; KLEIN, B.; TABORSKY, E.; BLANTON, A.; CHENEY, J.; ALLEN, K.; GROTH, P.; MAH, A.; JAIN, A. K. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1931–1939.

KLARE, B. F.; KLEIN, B.; TABORSKY, E.; BLANTON, A.; CHENEY, J.; ALLEN, K.; GROTH, P.; MAH, A.; JAIN, A. K. Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1931–1939.

KOCH, G.; ZEMEL, R.; SALAKHUTDINOV, R. *et al.* Siamese neural networks for one-shot image recognition. In: LILLE. **ICML deep learning workshop**. [S.l.], 2015. v. 2.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, 2012.

LIU, W.; ANGUELOV, D.; ERHAN, D.; SZEGEDY, C.; REED, S.; FU, C.-Y.; BERG, A. C. Ssd: Single shot multibox detector. In: SPRINGER. **Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14**. [S.l.], 2016. p. 21–37.

LIU, W.; WEN, Y.; YU, Z.; LI, M.; RAJ, B.; SONG, L. Sphereface: Deep hypersphere embedding for face recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 212–220.

LOWE, D. G. Distinctive image features from scale-invariant keypoints. **International journal of computer vision**, Springer, v. 60, p. 91–110, 2004.

LUCENA, I. B.; SANTOS, L. de O.; SANTOS, M. A. dos; MARQUES, A. G.; LUCIO, J. M. L.; SOUZA, L. F. de F.; REGO, P. A. L.; FILHO, P. P. R. A convolutional network-based applied approach to frontal facial recognition and solutions for occluded images. In: **Congresso Brasileiro de Automática-CBA**. [S.l.: s.n.], 2022. v. 3, n. 1.

MAZE, B.; ADAMS, J.; DUNCAN, J. A.; KALKA, N.; MILLER, T.; OTTO, C.; JAIN, A. K.; NIGGEL, W. T.; ANDERSON, J.; CHENEY, J. *et al.* Iarpa janus benchmark-c: Face dataset and protocol. In: IEEE. **2018 international conference on biometrics (ICB)**. [S.l.], 2018. p. 158–165.

MCNEELY-WHITE, D.; SATTELBERG, B.; BLANCHARD, N.; BEVERIDGE, R. Canonical face embeddings. **IEEE Transactions on Biometrics, Behavior, and Identity Science**, IEEE, v. 4, n. 2, p. 197–209, 2022.

MENG, Q.; ZHAO, S.; HUANG, Z.; ZHOU, F. Magface: A universal representation for face recognition and quality assessment. In: **Proceedings of the IEEE/CVF conference on computer vision and pattern recognition**. [S.l.: s.n.], 2021. p. 14225–14234.

MILLER, D.; BROSSARD, E.; SEITZ, S.; KEMELMACHER-SHLIZERMAN, I. Megaface: A million faces for recognition at scale. **arXiv preprint arXiv:1505.02108**, 2015.

MINSKY, M.; PAPERT, S. A. **Perceptrons, reissue of the 1988 expanded edition with a new foreword by Léon Bottou: an introduction to computational geometry**. [S.l.]: MIT press, 2017.

MONTERO, D.; NIETO, M.; LESKOVSKY, P.; AGINAKO, N. Boosting masked face recognition with multi-task arcface. In: IEEE. **2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)**. [S.l.], 2022. p. 184–189.

OJALA, T.; PIETIKAINEN, M.; MAENPAA, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. **IEEE Transactions on pattern analysis and machine intelligence**, IEEE, v. 24, n. 7, p. 971–987, 2002.

PAPANTONIOU, F. P.; LATTAS, A.; MOSCHOGLLOU, S.; DENG, J.; KAINZ, B.; ZAFEIRIOU, S. Arc2face: A foundation model for id-consistent human faces. In: SPRINGER. **European Conference on Computer Vision**. [S.l.], 2024. p. 241–261.

QI, S.; ZUO, X.; FENG, W.; NAVEEN, I. Face recognition model based on mtcnn and facenet. In: IEEE. **2022 IEEE 2nd international conference on mobile networks and wireless communications (ICMNWC)**. [S.l.], 2022. p. 1–5.

ROMERO, Y. A. L.; Rocha Neto, A. R. da. Face reid method via deep learning. In: SPRINGER. **International Conference on Intelligent Data Engineering and Automated Learning**. Manchester, UK, 2022. p. 369–378.

ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

ROSTUM, H. M.; VÁSÁRHELYI, J. Comparing the effectiveness and performance of image processing algorithms in face recognition. In: IEEE. **2024 25th International Carpathian Control Conference (ICCC)**. [S.l.], 2024. p. 1–5.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

RYANDO, C.; SIGIT, R.; DEWANTARA, B. S. B. *et al.* Comparison of machine learning algorithms for face classification using facenet embeddings. **The Indonesian Journal of Computer Science**, v. 13, n. 4, 2024.

SANCHEZ-MORENO, A. S.; OLIVARES-MERCADO, J.; HERNANDEZ-SUAREZ, A.; TOSCANO-MEDINA, K.; SANCHEZ-PEREZ, G.; BENITEZ-GARCIA, G. Efficient face recognition system for operating in unconstrained environments. **Journal of Imaging**, MDPI, v. 7, n. 9, p. 161, 2021. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/34460797/>>.

- SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 815–823.
- SENGUPTA, S.; CHEN, J.-C.; CASTILLO, C.; PATEL, V. M.; CHELLAPPA, R.; JACOBS, D. W. Frontal to profile face verification in the wild. In: IEEE. **2016 IEEE winter conference on applications of computer vision (WACV)**. [S.l.], 2016. p. 1–9.
- SHI, H.; YANG, Y.; ZHU, X.; LIAO, S.; LEI, Z.; ZHENG, W.; LI, S. Z. Embedding deep metric for person re-identification: A study against large variations. In: SPRINGER. **Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14**. [S.l.], 2016. p. 732–748.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.
- SUN, Y.; WANG, X.; TANG, X. Deep learning face representation from predicting 10,000 classes. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 1891–1898.
- SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Proceedings of the AAAI conference on artificial intelligence**. [S.l.: s.n.], 2017. v. 31, n. 1.
- SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2015. p. 1–9.
- TAIGMAN, Y.; YANG, M.; RANZATO, M.; WOLF, L. Deepface: Closing the gap to human-level performance in face verification. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2014. p. 1701–1708.
- TURK, M. A.; PENTLAND, A. *et al.* Face recognition using eigenfaces. In: **CVPR**. [S.l.: s.n.], 1991. v. 91, p. 586–591.
- VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER Łukasz; POLOSUKHIN, I. Attention is all you need. **arXiv preprint arXiv:1706.03762**, 2017. Disponível em: <<https://arxiv.org/abs/1706.03762>>.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: IEEE. **Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001**. [S.l.], 2001. v. 1, p. I–I.
- WANG, H.; WANG, Y.; ZHOU, Z.; JI, X.; GONG, D.; ZHOU, J.; LI, Z.; LIU, W. Cosface: Large margin cosine loss for deep face recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 5265–5274.
- WANG, M.; DENG, W. Deep face recognition: A survey. **Neurocomputing**, Elsevier, v. 429, p. 215–244, 2021.
- WEN, Y.; ZHANG, K.; LI, Z.; QIAO, Y. A discriminative feature learning approach for deep face recognition. In: SPRINGER. **European conference on computer vision**. [S.l.], 2016. p. 499–515.

WHITELAM, C.; TABORSKY, E.; BLANTON, A.; MAZE, B.; ADAMS, J.; MILLER, T.; KALKA, N.; JAIN, A. K.; DUNCAN, J. A.; ALLEN, K. *et al.* Iarpa janus benchmark-b face dataset. In: **proceedings of the IEEE conference on computer vision and pattern recognition workshops**. [S.l.: s.n.], 2017. p. 90–98.

WILLIAM, I.; RACHMAWANTO, E. H.; SANTOSO, H. A.; SARI, C. A. *et al.* Face recognition using facenet (survey, performance test, and comparison). In: IEEE. **2019 fourth international conference on informatics and computing (ICIC)**. [S.l.], 2019. p. 1–6.

WOLF, L.; HASSNER, T.; MAOZ, I. Face recognition in unconstrained videos with matched background similarity. In: IEEE. **CVPR 2011**. [S.l.], 2011. p. 529–534.

XING, X.; XU, G.; CAI, B.; QING, C.; XU, X. Face verification based on feature transfer via pca-svm framework. In: IEEE. **2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)**. [S.l.], 2017. p. 1086–1091.

YI, D.; LEI, Z.; LIAO, S.; LI, S. Z. Learning face representation from scratch. **arXiv preprint arXiv:1411.7923**, 2014.

ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: SPRINGER. **Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13**. [S.l.], 2014. p. 818–833.

ZHANG, Q.; WU, Y. N.; ZHU, S.-C. Interpretable convolutional neural networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 8827–8836.