UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ENGENHARIA DE SÃO CARLOS

Leonardo Enrique Anastácio dos Reis

Uso de redes neurais artificiais para detecção de fraudes em sistemas biométricos de reconhecimento facial

São Carlos

2020

Ficha catalográfica elaborada pela Biblioteca Prof. Dr. Sérgio Rodrigues Fontes da EESC/USP com os dados inseridos pelo(a) autor(a).

Reis, Leonardo Enrique Anastácio dos
R518u Uso de redes neurais artificiais para detecção de fraudes em sistemas biométricos de reconhecimento facial / Leonardo Enrique Anastácio dos Reis; orientador Marcelo Andrade da Costa Vieira. São Carlos, 2020.

Monografia (Graduação em Engenharia Elétrica com ênfase em Eletrônica) -- Escola de Engenharia de São Carlos da Universidade de São Paulo, 2020.

1. reconhecimento facial. 2. fraude. 3. Data Augmentation. 4. Fine-Tuning. 5. aprendizado de máquina. I. Título.

Eduardo Graziosi Silva - CRB - 8/8907

FOLHA DE APROVAÇÃO

Nome: Leonardo Enrique Anastacio dos Reis

Título: "Uso de redes neurais artificiais para detecção de fraudes em sistemas biométricos de reconhecimento facial"

Trabalho de Conclusão de Curso defendido e aprovado em 03 / 12 / 2020,

com NOTA 9,0 (NOVE, ZERO), pela Comissão Julgadora:

Prof. Associado Marcelo Andrade da Costa Vieira - Orientador SEL/EESC/USP

Prof. Associado Adilson Gonzaga - Sênior/SEL/EESC/USP

Mestre Arthur Chaves Costa - Doutorando SEL/EESC/USP

Coordenador da CoC-Engenharia Elétrica - EESC/USP:
Prof. Associado Rogério Andrade Flauzino

A minha família.

AGRADECIMENTOS

Este trabalho de conclusão de curso é consequência de anos de estudos desenvolvidos durante a graduação em Engenharia Elétrica, todo conhecimento adquirido por meio de projetos pessoais e profissionais na área de dados e também de todo o apoio familiar.

Agradeço, primeiramente, a Deus, por todas as bençãos durante esses anos de graduação, todas as oportunidades abertas e todos os cuidados durante essa caminhada.

Agradeço aos meus pais, por terem me mostrado desde o início que os estudos deveriam estar em primeiro lugar na minha vida, por todo apoio ao longo desses anos e por todo amor demonstrado.

Agradeço a minha namorada, por sempre estar ao meu lado me incentivando, por toda paciência durante esses anos e por sempre me dar força em todos os momentos, amo você.

Agradeço a minha sogra, por ser uma mãe pra mim em São Carlos, pelas palavras de incentivo e pelos conselhos em diversos momentos.

Agradeço aos meus professores da graduação, em especial ao professor Marcelo, por ter me incentivado neste projeto de conclusão de curso desde o inicio, durante a matéria de Visão Computacional, e por transmitir os conhecimentos necessários para a elaboração do trabalho. Agradeço também a professora Janete Crema, por ter sido minha primeira orientadora na Universidade e ter transferido a importância do engajamento com pesquisas acadêmicas.

Agradeço aos meus amigos de graduação, em especial ao Pedro Caramalac, Mateus Bonati, Felipe Mostardeiro, Weslley Perissin, Vinicius Andreguetti, Lucas Frascarelli, Bruno Freitas, Larissa Lima e Gabriel Ribeiro, por toda parceria durante estes anos, nunca se esqueçam: *"vai dar bom sempre"*.

Agradeço a Serasa Experian, por ter proporcionado a oportunidade do estágio profissional durante a graduação e a todos meus ex-colegas de trabalho, em especial ao José, Vlademir, Luiz, e Felipe, por terem me auxiliado no inicio da carreira e terem transmitido um conhecimento sólido que levarei comigo sempre.

Agradeço ao SiDi, pela oportunidade de ingressar em um novo projeto profissional, por todo apoio durante a finalização deste trabalho e por todo incentivo de continuar expandindo

minha experiência na área de dados. Agradeço também aos meus novos colegas de trabalho, por compartilhar um conhecimento em dados e novas tecnologias, sem as quais, não conseguiria finalizar este trabalho com êxito.

“Em um dado dia, uma dada circunstância, você acha que tem um limite.
Você então tenta ir para esse limite e você toca esse limite, e você pensa: ’Ok, este é o limite.’
Logo que você toca esse limite, algo acontece e de repente você pode ir um pouco mais longe.
Com o poder da sua mente, sua determinação, seu instinto, e a experiência também,
você pode voar muito alto.”

Ayrton Senna

RESUMO

REIS, L. E. A. **Uso de redes neurais artificiais para detecção de fraudes em sistemas biométricos de reconhecimento facial**. 2020. 89p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020.

A inserção do uso de reconhecimento facial em sistemas biométricos forneceu uma maior velocidade e a praticidade durante os procedimentos de autenticação, no entanto, com incremento desses mecanismos em diversas aplicações fez crescer também a necessidade de procedimentos que evitem falsificações e tentativas de fraude, também chamados de métodos *anti-spoofing* facial. A utilização de métodos de detecção de vida (movimento dos olhos e da face), comparação de face e fundo da imagem, identificação da qualidade da imagem por meio de descritores de cor, textura, e processamentos para a retirada de possíveis ruídos, são utilizados com o intuito de avaliar a veracidade da face. Agrega-se, também, a estes métodos, o uso de tecnologias de aprendizado de máquina com o objetivo de reconhecer padrões existentes tanto para faces reais, quanto para as adulteradas. Desta maneira, o objetivo deste trabalho é avaliar o comportamento das Redes Neurais Artificiais quando expostas a dois diferentes cenários: a identificação da identidade da pessoa por trás do sistema de autenticação e a classificação de fraude biométrica utilizando o mesmo sistema de avaliação. Para isto, utilizou-se, como método de avaliação, a técnica de *Fine-Tuning*, retreinando um modelo convolucional de classificação facial, expondo-o a uma base de faces reais de 5105 imagens e de 7509 imagens fraudulentas (falsas) também, a técnica de *Data Augmentation* por meio de um pré-processamento, com o filtro Gaussiano passa-alta e o descritor de textura *Local Binary Pattern* (LBP). Como resultado, observou-se que a utilização de apenas uma rede neural para o sistema biométrico de reconhecimento facial convencional, quando exposto a entrada de imagens falsas, possui um decaimento em sua performance de classificação, enquanto, a utilização do mesmo sistema para a classificação de veracidade das faces, apresenta um melhor desempenho, ainda mais se atrelado a técnicas de pré-processamento. Desta maneira, conclui-se que a utilização de diferentes redes neurais artificias, para diferentes objetivos, contribui tanto para melhorar a eficiência dos métodos de reconhecimento facial quanto para a detecção de fraudes.

Palavras-chave: reconhecimento facial, fraude, *Data Augmentation*, *Fine-Tuning*, aprendizado de máquina.

ABSTRACT

REIS, L. E. A. **Use of artificial neural networks for spoofing detection in biometric face recognition systems**. 2020. 89p. Monografia (Trabalho de Conclusão de Curso) - Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2020.

The insertion of the use of facial recognition in biometric systems provided greater speed and practicality during authentication procedures, however, with the increase of these mechanisms in several applications, the need for procedures that prevent forgeries and fraud attempts, also called of facial anti-spoofing methods. The use of methods of detecting life (movement of the eyes and face), comparing the face and background of the image, identifying the quality of the image through color descriptors, texture, and processing to remove possible noise, are used in order to assess the veracity of the face. The use of machine learning technologies is also added to these methods in order to recognize existing patterns for both real and adulterated faces. Thus, the objective of this work is to evaluate the behavior of Artificial Neural Networks when exposed to two different scenarios: the identification of the person's identity behind the authentication system and the classification of biometric fraud using the same evaluation system. For this, the Fine-Tuning technique was used as an evaluation method, retraining a convolutional model of facial classification, exposing it to a base of real faces of 5105 images and 7509 fraudulent (false) images as well. Data Augmentation technique through pre-processing, with the high-pass Gaussian filter and the Local Binary Pattern (LBP) texture descriptor. As a result, it was observed that the use of only one neural network for the conventional facial recognition biometric system, when exposed to the entry of false images, has a decline in its classification performance, while the use of the same system for the classification of veracity of the faces, presents a better performance, even more linked to pre-processing techniques. Thus, it is concluded that the use of different artificial neural networks, for different purposes, contributes both to improve the efficiency of facial recognition methods and to the detection of fraud.

Keywords: face recognition, fraud, *Data Augmentation*, *Fine-Tuning*, machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

API	*Application Programming Interface* - Interface de programação de aplicações
AUC	*Area Under Curve* - Área sob a curva
CE	*Categorical Cross-Entropy* - Crossentropia Categórica
CPU	*Central Processing Unit* - Unidade central de processamento
CNN	*Convolutional Neural Network* - Rede neural convolucional
COLAB	Google Colaboratory
FN	*False Negative* - Falso negativo
FP	*False Positive* - Falso positivo
GPU	*Graphics Processing Unit* - Unidade de processamento gráfico
LBP	*Local Binary Pattern* - Padrão Binário Local
MLP	*Multilayer Percepton* - Percepton multi camada
NUAA	Nanjing University of Aeronautics and Astronautics
PB	Passa-baixa
RAM	*Random Access Memory* - Memória de acesso aleatório
ReLu	*Rectified Linear Unit* - Unidade Linear Retificada
RBM	*Restricted Boltzmann Machine* - Máquina de Boltzmann Restrita
RNA	Rede Neural Artificial
SGD	*Stochastic Gradient Descent* - Gradiente Descendente Estocástico
TP	*True Positive* - Verdadeiro positivo
TPU	*Tensor Processing Unit* - Unidade de Processamento de Tensor
VGG	*Visual Geometry Group*

SUMÁRIO

1 INTRODUÇÃO

1.1 Biometria

Com o desenvolvimento da tecnologia nos últimos anos, tanto no campo de uso pessoal, quanto no uso profissional, os sistemas que necessitam confirmar a identidade dos usuários, por motivos de segurança a dados, passaram a utilizar algumas maneiras para a identificação humana com o intuito de evitar tentativas de fraude. Pode-se dividir estes métodos de autenticação em três maneiras, considerando os conhecimentos pessoais, neste caso, fazendo-se uso de senhas, perguntas pessoais ou PINs; levando em conta itens que encontram-se em posse humana, como *tokens* ou *pen-drivers* com informações de autenticação criptografadas; também, características pessoais, as quais são denominadas características biométricas (WEAVER, 2006).

Segundo o Dicionário Priberam da Língua Portuguesa, biometria pode ser definida como:

- medição dos seres vivos e de propriedades mensuráveis;
- estudo das propriedades únicas mensuráveis de cada individuo, em especial para verificação automática de identidade;
- cálculo da duração provável da vida.

Assim no escopo deste trabalho, a biometria é baseada em características pessoais, as quais podem ser divididas em dois campos, o comportamental e biológico (WEAVER, 2006). A primeira abordagem refere-se a padrões de assinatura, reconhecimento de voz, maneira de andar, entre outros; já o segundo campo trata-se de escaneamento de iris e retina, impressões digitais, reconhecimento facial, entre outros. Esses atributos únicos podem ser utilizados como forma de prevenir tentativas de acesso a ambientes pessoais por pessoas não autorizadas, utilizando-se de equipamentos que verifiquem a autenticidade da conexão (BABICH, 2012).

Sendo assim, muitos dispositivos eletrônicos, desde computadores pessoais, até transações bancárias e confirmações cadastrais por parte de governos, tem utilizado características biométricas como forma de autenticação com o intuito de minimizar fraudes e acessos indevidos. Em especial, tem-se empregado o reconhecimento facial nos sistemas de autenticação, por ser uma característica única de cada indivíduo, método que não é invasivo e não colaborativo, como o

reconhecimento de íris, ou que é facilmente reproduzida por fraudadores, como o reconhecimento por impressão digital (SCHUCKERS, 2002).

1.2 Técnicas de Fraude

As tentativas de fraudes existentes em sistemas biométricos podem ser dividas em dois cenários: (1) quando o ataque ao sistema é realizado forçando o indivíduo que possui a permissão de conexão a realizar e validar o acesso; (2) quando as características biométricas são falsificadas, utilizando-se de moldes reais ou digitais. Para o caso de reconhecimento facial, o processo de autenticação torna-se mais vulnerável pela alta disponibilidade de imagens faciais na internet, em especial em redes sociais, o que acaba favorecendo o processo de fraude, diferentemente do reconhecimento por íris ou impressões digitais, para os quais não há essa facilidade de captura de informações biométricas (SCHUCKERS, 2002).

Com o intuito coibir estas tentativas de fraude, tem-se que, para o primeiro cenário, a existência desde câmeras de segurança, até botões de pânico ou alarmes, são soluções empregadas para tentar evitar esse tipo de invasão. Já para o segundo cenário, torna-se necessário a implementação de algoritmos que identifiquem padrões de fraude com o intuito de minimizar os ataques aos sistemas biométricos (SCHUCKERS, 2002).

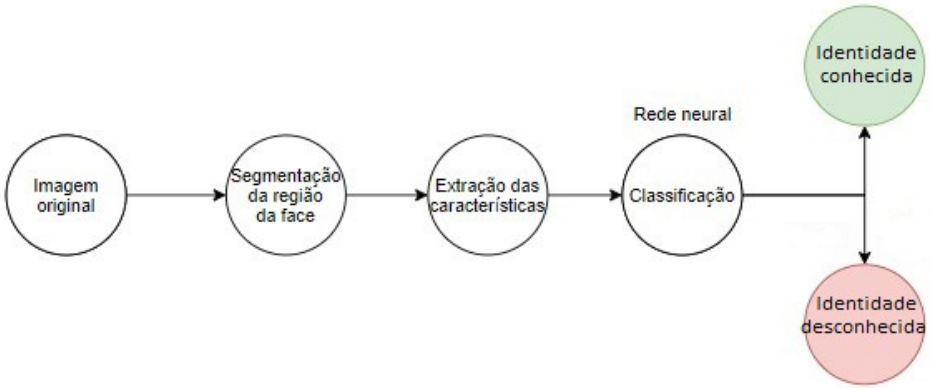
Alguns testes realizados em procedimentos de autenticação biométrica, por meio de pesquisas científicas, constataram que os equipamentos ainda são vulneráveis a ataques de falsificação. Essas pesquisas abordaram a confecção de dedos falsos com impressões digitais desejadas, a utilização de métodos para capturar impressões digitais existentes nas superfícies empregadas para autenticação, e também fotos e vídeos para legitimar o reconhecimento facial, e, para todas as técnicas, houve momentos em que os sistemas verificaram o acesso como verdadeiro (SCHUCKERS, 2002). Desta maneira, tem-se que as principais técnicas de fraude em sistemas de reconhecimemento facial resumem-se na utilização de fotografias e vídeos em alta resolução, máscaras 2D para o recorte da área dos olhos para movimentação ocular, e máscaras 3D, obtidas por meio de impressoras de última geração.

1.3 Justificativa

Com o avanço tecnológico dos últimos anos, a utilização de sistemas biométricos para autenticação de acesso passou a ser implementada com maior frequência e a fazer parte da rotina da população. Um dos sistemas biométricos convencionais implementado pode ser observado

na Figura 1, para o qual é efetuado a extração da imagem face e suas características, e, posteriormente, realizada a classificação da imagem em relação a sua identidade juntamente com a avaliação de veracidade da face por trás da câmera (GALBALLY et al., 2014).

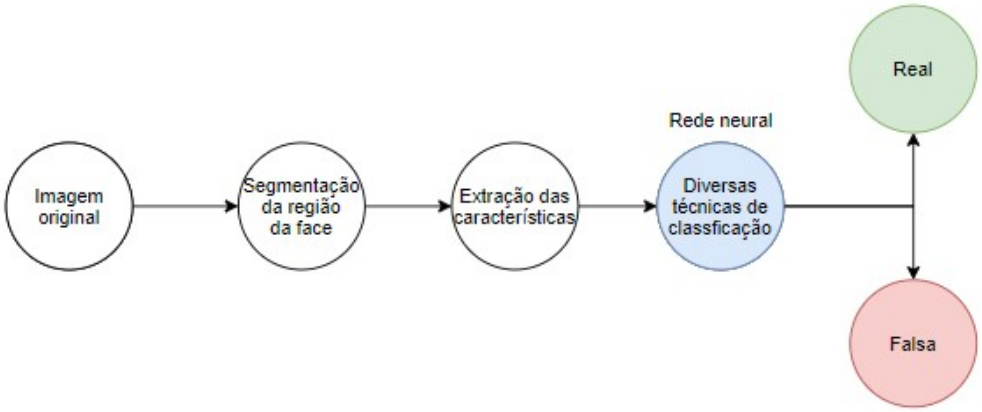
Figura 1 – Sistema biométrico convencional - Adaptado (Fonte: GALBALLY et al., 2014)



.

Observa-se que alguns trabalhos recentes, como SOUZA (2019) e DESAI (2019), buscaram abordar, com diferentes métodos de aprendizado de máquina, técnicas de aprimoramento a estes sistemas por meio de redes neurais convolucionais profundas (CNNs), máquina de Boltzmann restrita (RBM), redes neurais temporais e redes neurais recorrentes, conforme mostra a Figura 2. Esses observaram que a proposta de novas arquiteturas de redes profundas, para imagens em duas dimensões, em relação aos modelos existentes, são boas alternativas para a detecção de fraude em sistemas de reconhecimento facial, tanto no quesito de extração de características de fisionomia, quanto para o processamento do reconhecimento facial em dispositivos com arquiteturas computacionais menos robustas.

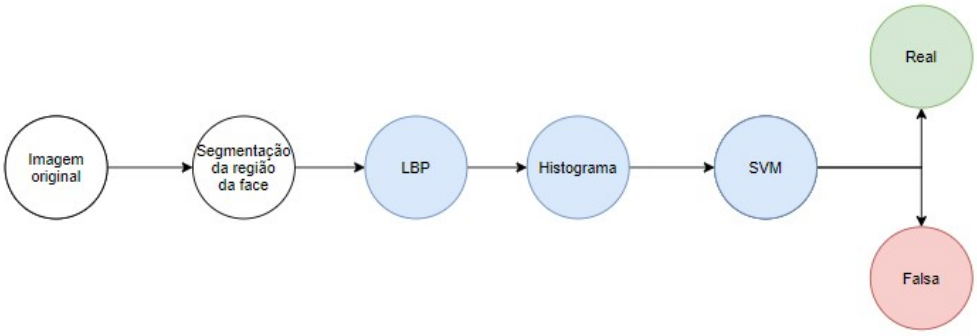
Figura 2 – Sistema biométrico proposto na literatura recente utilizando redes neurais convolucionais (Fonte: Autor)



.

Além disso, estudos relacionados a detecção de vida nas imagens da face, como observado em TIRUNAGARI et al. (2015), por meio dos movimentos oculares e labiais, que utilizam conceitos da dinâmica dos fluidos, por meio do fluxo sanguíneo, são empregados na prevenção de fraudes. Esta técnica faz uso de métodos de decomposição de vídeos em uma sequência de *frames*, para o qual as imagens são submetidas a um descritor de textura, neste caso, o *Local-Binary Pattern* (LBP), e, posteriormente, extraídos os seus histogramas, para que estes sejam submetidos a um modelo SVM (*Support Vector Machine*) para a classificação - Figura 3.

Figura 3 – Sistema biométrico proposto na literatura recente por meio de pré-processamento de imagens (Fonte: Autor)



.

Entretanto, todas estas abordagem buscaram somente soluções voltadas para a modificação na arquitetura do sistema de reconhecimento biométrico já existente. Desta maneira, observa-se que, no cenário atual, com o aumento do acesso a tecnologias por parte da população e a maior exposição de imagens faciais em veículos de imprensa, redes sociais, além de novas

técnicas de impressão 3D, torna-se necessário a implementação de novas arquiteturas de reconhecimento facial, utilizando desde técnicas de pré-processamento de imagens, até diferentes técnicas de aprendizagem profunda, visando uma melhor detecção de fraude e performance de processamento.

1.4 Objetivo

Este estudo tem como objetivo: a) a verificação do desempenho da atual abordagem do sistema biométrico convencional quando exposto a tentativas de *spoofing* facial, por meio da inserção de imagens falsas em um modelo de classificação facial; b) a verificação do comportamento do modelo de classificação facial quando exposto somente a imagens reais e falsas; c) a avaliação de desempenho da implementação de pré-processamento, por meio do filtro Gaussiano e do descritor de textura LBP, com a técnica de *Data Augmentation*, em um modelo de classificação facial para classificação de imagens reais e falsas; d) a comparação dos resultados do sistema biométrico convencional quando exposto a tentativas fraude com outra rede responsável somente pela identificação da veracidade da face.

1.5 Organização da monografia

Em adição a este capítulo introdutório, constará também neste trabalho de conclusão de curso as seguintes partes:

- Capítulo 2 - Estudo Teórico: Nesse capítulo será apresentada toda a teoria aplicada neste trabalho, desde as técnicas de processamento de imagens e reconhecimento de padrões, até o funcionamento das redes neurais e sua construção.

- Capítulo 3 - Materiais: Nesse capítulo serão especificados todas as técnicas utilizadas no desenvolvimento deste trabalho e as ferramentas utilizadas na sua elaboração.

- Capítulo 4 - Métodos: Nesse capítulo serão apresentadas as características da rede neural utilizada, além das métricas de avaliação e a especificação dos testes desenvolvidos.

- Capítulo 5 - Resultados: Nesse capítulo serão mostrados os resultados obtidos dos testes propostos.

- Capítulo 6 - Conclusão: Nesse capítulo será apresentada uma conclusão dos estudos realizados, das implicações de possíveis implementações em sistemas reais e de futuros trabalhos a serem propostos.

2 ESTUDO TEÓRICO

2.1 Processamento de imagem

2.1.1 Domínio Espacial

Como primeira abordagem de processamento de imagens, tem-se as operações baseadas no domínio espacial - manipulação direta dos pixels da imagem - as quais, por meio dos filtros passa-alta, atuam no realce de imagem, procurando melhorar a qualidade das imagens, adequando-as para uma futura aplicação de reconhecimento de padrões (ACHARYA, 2005). As técnicas de realce são dividas em duas categorias: a primeira são as transformações de intensidade, as quais agem sobre o nível cinza da imagem; e, a segunda, a filtragem espacial, a qual considera a vizinhança de cada pixel antes de realizar qualquer processamento na imagem. Em especial tem-se as técnicas que são utilizadas para o aguçamento das imagens, que são utilizadas com o intuito de evidenciar as transições das imagens, gerando, desta maneira, um aumento da nitidez. (GONZALEZ WOODS, 2009).

2.1.1.1 Filtro Gaussiano

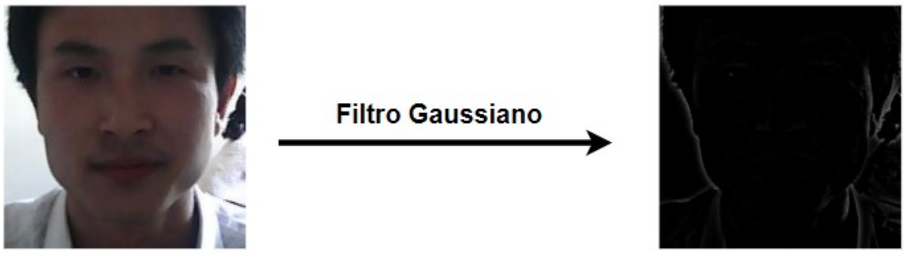
O filtro Gaussiano é um filtro utilizado em processos de suavização de imagens o qual proporciona a atenuação do ruído. No entanto, pode-se utilizar o filtro Gaussiano como filtro passa-alta, com o intuito aplicar técnicas de aguçamento e realçar as regiões de alta frequência das imagens.

Para este aplicação, tem-se na Equação 2.1 o filtro Gaussiano passa-baixa dado por G_b e o filtro Gaussiano passa-alta dado por G_a, o qual é gerado por meio do complemento do filtro gaussiano passa-baixa.

$$G_a = 1 - G_b \tag{2.1}$$

Como exemplo da utilização do filtro Gaussiano passa-alta em imagens faciais, pode-se observar na Figura 4 a imagem original a esquerda e a imagem após a aplicação do filtro Gaussiano passa-alta a direita.

Figura 4 – Exemplo da utilização do filtro Gaussiano em uma imagem - Esquerda: Original - Direita: Filtro Gaussiano. (Fonte: Autor)



Desta maneira, observa-se que a aplicação do filtro Gaussiano para a detecção de fraude em imagens faciais é válida, pois, para as regiões em que tem-se o reflexo da luz natural na face real, a utilização do filtro Gaussiano passa-alta gera destaque, fenômeno não observado nas imagens falsas - o qual será observado posteriormente no Capítulo 4.

2.2 Representação e Descrição

Para a etapa de representação e descrição, ambos os procedimentos se iniciam após a obtenção de resultados da etapa de segmentação, o qual fornece insumos referentes as partes da imagem, como bordas, por exemplo. O processo de representação é dividido em duas abordagens, a primeira quando o foco é dado a região interna da forma - representação por região - e a segunda quando o foco volta-se para a região extrema da forma - representação por fronteira. Já para o processo de descrição, tem-se a seleção de características, na qual os atributos da imagem são extraídos gerando, desta maneira, informações quantitativas da imagem (GONZALEZ et al., 2009).

2.2.1 Descritor de textura

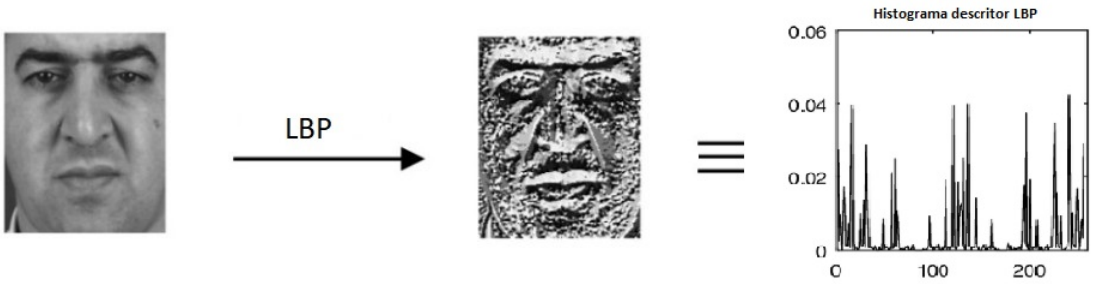
A utilização de descritores de textura em uma imagem proporciona a identificação de uma região e, posteriormente, a classificação da mesma, por meio de algumas propriedades como rugosidade, suavidade e regularidade (OLIVEIRA et al, 2014). Existem três principais métodos de abordagem de uso destes descritores: estatística, a qual gera uma caracterização de textura como suave, rugosa, granulada, entre outras; estrutural, em que arranjos primitivos da imagem são analisados, gerando, por exemplo, a descrição por meio de linhas paralelas espaçadas; e, por fim, a espectral, na qual utiliza-se do espectro de Fourier para detectar comportamentos semelhantes dos picos de energia presentes no espectro (GONZALEZ et al., 2009).

2.2.1.1 *Local-Binary Pattern*

Local-Binary Pattern (LBP) pode ser definido como um descritor de textura não paramétrico, o qual resume as informações de estruturas locais de uma imagem, comparando-as a cada pixel existente em sua vizinhança, tornando-se um descritor muito poderoso que despertou interesse de pesquisas nos campos de visão computacional e processamento de imagem (HUANG, 2001).

O LBP tem sido aplicado em diversas áreas, desde análise de imagens faciais, recuperação de imagens, análise biomédica, até mesmo modelagem de ambiente e sensoriamento remoto. As pesquisas mais recentes apresentam resultados que sustentam que a abordagem LBP fornece ótimos resultados na representação e análise facial tanto em imagens fixas, quanto em sequências de vídeo (HUANG, 2001). Desta maneira a utilização do LBP para a análise de imagens faciais para detecção de fraude é pertinente, pois a intensidade de cinza nos pixels das imagens reais difere-se em comparação com as imagens falsas, ficando evidente com a aplicação do descritor de textura LBP - exemplos dessa diferença serão abordados no Capítulo 4. Como da aplicação do descritor LBP em uma imagem tem-se a Figura 5,

Figura 5 – Utilização do descritor de textura LBP em uma imagem - Esquerda: Original - Centro: Mapa de textura gerado pelo descritor LBP - Direita: Histograma do nível de cinza do mapa de textura gerado pelo descritor LBP. Adaptado (Fonte: HUANG et al., 2011)



2.3 Reconhecimento

O reconhecimento de imagens é o procedimento utilizado após a segmentação de regiões em que as características de cada região da imagem são interpretadas, e um rótulo é atribuído as áreas da imagem por meio da utilização dos resultados obtidos nos descritores (QUEIROZ, 2006). Além disso, quando vários objetos possuem características semelhantes, define-se que estes são atribuídos a uma mesma classe. Já em relação as metodologias abordadas na etapa de reconhecimento, estas podem ser divididas em duas categorias: a teórica, na qual são utilizados

descritores quantitativos - forma, área, comprimento e textura - para as análises; e a estrutural, na qual descritores qualitativos são abordados para o entendimento dos padrões existentes nos objetos (GONZALEZ et al., 2009).

2.3.1 Vetor de Características

Como abordado anteriormente, uma classe de padrões é definida por imagens que possuam características semelhantes, ou seja, quando compartilham propriedades extraídas por meio dos descritores. Na prática, observa-se três tipos de arranjos utilizados para descrever as classes as quais os objetos pertencem, o vetor, as *strings* e as arvores - as quais representam descrições estruturais (GONZALEZ et al., 2009). A união de todas essas características das imagens, obtidas por meio de medidas estatísticas, geram os componentes que formam o vetor de características (HARALICK, 2010). Como forma de representar o vetor de características, tem-se a Equação 2.2, na qual cada componente do vetor representa uma característica atribuída por um descritor, formando, assim, uma matriz $nx1$ sendo n o número de descritores presentes na classe (GONZALEZ et al., 2009).

$$x = \begin{bmatrix} d_1 \\ d_2 \\ . \\ . \\ . \\ d_n \end{bmatrix} \tag{2.2}$$

2.3.2 Classificadores

Durante a etapa de classificação de imagens, tem-se o reconhecimento baseado na utilização de funções de decisão. As técnicas que baseiam-se na comparação dos vetores de características das imagens, em relação a vetores protótipos de cada classe pré definidos, são denominadas casamento de classes. A realização desses comparativos pode ser efetuada de diversas formas, sendo, a mais simples, a classificação que faz uso da distância mínima entre o vetor das imagens e o vetor padrão das classes, escolhendo, desta maneira, a menor distância encontrada para a tomada de decisão. Além disso, técnicas baseadas na correlação, ou em outras operações probabilísticas, entre os vetores, também podem ser utilizadas com o intuito de encontrar padrões para o reconhecimento (GONZALEZ et al., 2009). Como exemplo da utilização de classificadores, observa-se a aplicação abordada por Queiroz (2006):

"Um sistema para a classificação de imagens coletadas da Web em duas classes semânticas, gráficos e fotografias, foi apresentado. O sistema utilizou um método de classificação baseado em árvores de decisão (ID3, um algoritmo de indução de árvores de decisão a partir de exemplos, popular na área de IA). Foi identificado um conjunto de características adequadas à separação entre as duas classes semânticas escolhidas. Características marcantes de fotografias identificadas foram: (i) existências de objetos reais com uma tendência a texturas e ausência de regiões com cores constantes; (ii) pequenas diferenças na proporção (altura x largura); (iii) poucas ocorrências de regiões com alta saturação de cores; e (iv) presença de um grande número de cores utilizadas. As características identificadas como marcantes de gráficos foram: (i) presença de objetos artificiais com bordas bem definidas bem como a presença de regiões cobertas com cores saturadas; e (ii) grandes diferenças na proporção e tendência a serem menores em tamanho do que fotografias. Assim, foram definidas métricas sobre o número de cores, a cor predominante, o vizinho mais distante, a saturação, o histograma de cores, o histograma do vizinho mais distante, a proporção das dimensões e a menor dimensão."

2.4 Rede Neural Artificial

Rede Neural Artificial (RNA) é definida como uma estrutura desenvolvida para assemelhar-se ao cérebro humano; o conhecimento adquirido pela RNA por meio de um processo de aprendizagem e a utilização de sistema análogo aos neurônios para armazenamento de conhecimento evidenciam a semelhança (HAYKIN et al., 2001).

As unidades denominadas neurônios artificiais são utilizadas para fornecer as RNAS uma interligação maciça, proporcionando o cálculo de funções matemáticas em relação a entrada da rede. Além disso, estas unidades são descritas por diversas camadas interligadas por conexões, as quais são associadas a diferentes pesos, sendo estes utilizados para ponderar as entradas recebidas da rede (SOARES FILHO et al., 2018).

A aplicação desses métodos de cálculos, em larga escala, utilizando redes neurais artificiais até 2006 não era possível, por conta de todo impedimento tecnológico existente na época. Entretanto, após o surgimento de novas tecnologias e capacidade de processamento, os modelos baseados em redes neurais de aprendizado em profundidade (*Deep Learning*) permitiram que este tipo de método de classificação fosse empregado em diferentes frentes, como visão compu-

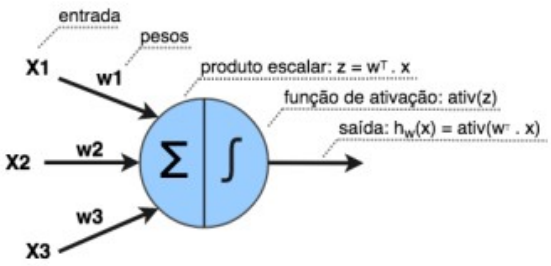
tacional, reconhecimento de fala, aplicações de multimídia no *TensorFlow* e processamento de linguagem natural (BUSSON et al., 2018).

2.4.1 Perceptron

A estrutura primitiva de uma Rede Neural Artificial foi desenvolvida por Frank Rosenblatt em 1957 e é denominada Perceptron. Na Figura 6 tem-se a representação do sistema da RNA Perceptron, em que cada entrada possui um peso associado a ela, na qual o produto escalar é aplicado, a soma de cada operação resultante é realizada, e, após isso é efetivado o cálculo da função de ativação presente no algoritmo Perceptron, gerando, assim, sua saída - o cálculo realizado é representado pela Equação 2.3, em que x representa a entrada, w representa o peso atribuído a entrada e b uma constante qualquer (BUSSON et al., 2018).

$$f(x) = \begin{cases} 1, & \text{se } w.x + b > 0 \\ 0, & \text{para outros casos} \end{cases} \tag{2.3}$$

Figura 6 – Estrutura de um Perceptron. (Fonte: BUSSON, 2015)



Pode-se utilizar o modelo de Perceptron em multicamadas (MLP) para realizar um processamento paralelo para uso do aprendizado supervisionado por meio de Redes Neurais Artificiais (JOST, 2015). Nesta aplicação, cada neurônio da rede neural é responsável por aprender e ativar uma função para cada classe específica, obtendo, como resultado, uma saída dada por uma função $argmax$ a qual seleciona o neurônio que possuiu uma maior ativação entre as classes avaliadas. Além disso, os neurônios podem ser estruturados em diversas camadas, nas quais as entradas dos neurônios das camadas mais profundas são ligados nas saídas dos neurônios das camadas anteriores, e, desta maneira, a rede neural aplica transformações lineares e não-lineares de forma hierárquica afim de gerar representações para dos dados de entrada e realizar classificações (BUSSON et al., 2018). Após todo o processamento de aprendizado em profundidade - utilizando dados rotulados, por exemplo - espera-se que a rede neural artificial seja

capaz de prever saídas baseadas nos dados de entrada fornecidos; esses modelos são utilizados para classificação e previsão de valores (JOST et al., 2015).

2.4.2 Rede Neural Convolucional

Para os procedimentos de filtragem de imagens no domínio da frequência, tem-se como principio o teorema da convolução. A operação de convolução entre dois vetores A e B, quando avaliada no domínio espacial, pode ser definida como $g(x, y) = f(x, y) * h(x, y)$, sendo compreendida como a soma dos produtos dos valores dos vetores que se sobrepõe a cada passo temporal. Já para o domínio da frequência, a mesma relação também é válida, sendo denotada por $G(u, v) = F(u, v)H(u, v)$, em que as funções G, F e H representam as Transformada de Fourier das funções g, f e h, respectivamente, sendo a função $H(u, v)$ denominada função de transferência do filtro (MARQUES FILHO et al., 1999).

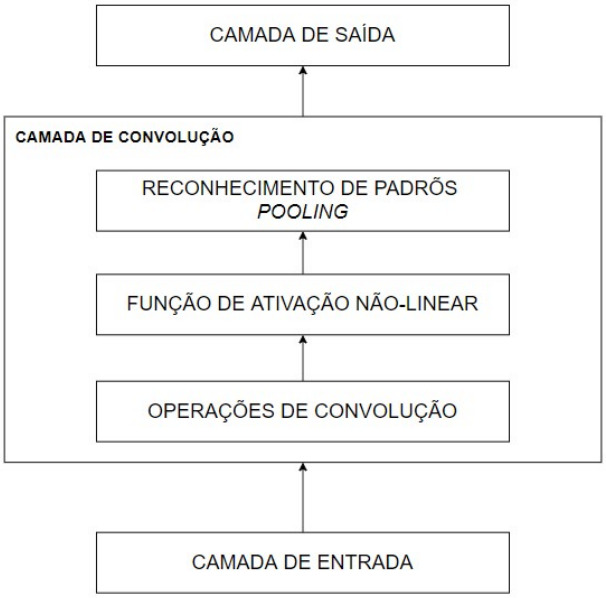
Por meio de todo avanço tecnológico nos últimos anos, tanto no campo de novas soluções para aprendizado em profundidade - *Deep Learning* - quando em relação a capacidade computacional, tornou-se possível a solução de vários problemas existentes. No entanto, outras questões surgiram, como no exemplo descrito por Pavlovsky (2017) em *Introduction To Convolutional Neural Networks*:

> *"Digamos que queremos detectar o rosto humano a partir da imagem. Uma rede neural simples atribuiria cada pixel a um neurônio na camada de entrada. Mas o que isso significa? Isso significa que não mantemos informações espaciais de pixels. Dividimos a imagem em neurônios individuais e depois alimentamos a rede com eles. Mas em caso de reconhecimento de rosto, você tem partes como olhos. Os olhos são objetos complexos compostos de várias partes. Você tem pupila, íris, esclera e até pálpebras. Todo olho os tem. Você seria capaz de detectar o olho apenas por um pixel ou apenas por uma parte? Provavelmente não. Somente o todo em ordem específica faz sentido. Se você treinar a rede neural na imagem do olho, só funcionará se o olho estiver na mesma posição exata da imagem todas as vezes. Quando você move, escala ou gira o olho, a rede inevitavelmente falhará em prever a saída correta. Precisamos de uma maneira de procurar padrões específicos em vez de pixels individuais. E é isso que as redes neurais convolucionais fazem."* - Tradução livre.

Desta maneira, torna-se necessário a implementação de Redes Neurais Convolucionais (CNNs), afim de resolver questões referente ao processamento de imagens de forma dinâmica, afim da rede neural ser capaz de prever entradas dinâmicas. As Redes Neurais Convolucionais são redes que utilizam do cálculo da operação de convolução para o processamento de dados organizados em grade (BUSSON et al., 2018). A etapa de convolução recebe a imagem fornecida como entrada, realiza as operações no núcleo denominado *kernel* e fornece como resultado um mapa de características (PACHECO, 2018). As CNNs são indicadas para o processamento de imagens, e reconhecimento de padrões, pelo fato desta, por meio das camadas de convolução, encontrar *features* que caracterizem diferentes comportamentos nas imagens de entrada (BUSSON et al., 2018).

De forma mais específica, as camadas que compõe uma Rede Neural Convolucional podem ser dividas em três etapas - como ilustrado na Figura 7. A primeira etapa consiste na execução de diversas operações de convolução ocorrendo em paralelo. Na segunda etapa, para cada ativação linear produzida, executa-se uma função de ativação não-linear, como forma de um estágio detector. Já no terceiro e último estágio, utiliza-se um processo denominado *pooling*, no qual algumas características obtidas nas fases anteriores são agrupadas, com o intuito do reconhecimento de padrões (GOODFELLOW et al., 2015).

Figura 7 – Componentes das Camadas da Rede Neural Artificial. (Fonte: GOODFELLOW et al., 2015)



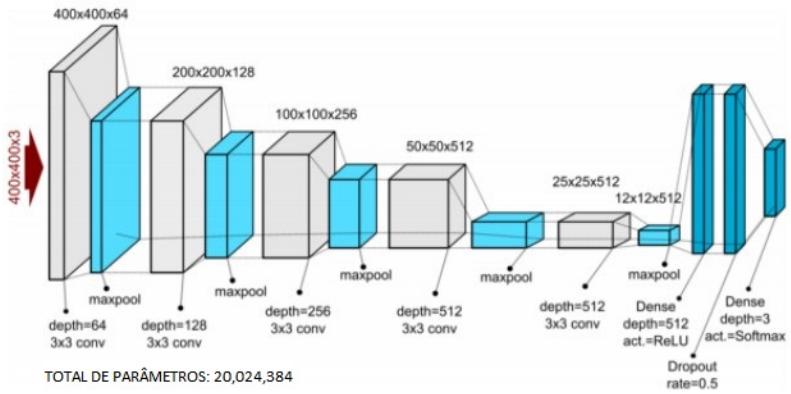
2.4.2.1 VGGFace

Como modelo de aprendizado profundo pré-treinado, tem-se o VGGFace, um modelo desenvolvido pelos membros do Visual Geometry Group (VGG) da University of Oxford, em 2015, para reconhecimento facial. O objetivo da VGGFace é o reconhecimento facial - desde uma única foto até um conjunto de faces presentes em um vídeo. O desenvolvimento desta rede neural só foi possível por conta da disponibilidade de conjunto de dados para treinamento - neste caso utilizou-se 2,6 milhões de imagens de, aproximadamente, 2,6 mil pessoas - e, também, da evolução da complexidade de modelos de rede neurais convolucionais (CNN), podendo atingir resultados comparáveis a modelos de reconhecimento de imagens desenvolvidos pelo Google e pelo Facebook (PARKHI et al., 2015).

Pode-se observar nas Figuras 8 e 9 as camadas presentes na VGGFace utilizada.

Figura 8 – Camadas VGGFace. (Fonte: JAWOREK-KORJAKOWSKA et al., 2019)

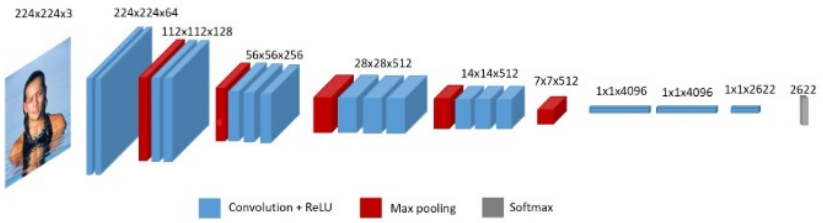
Figura 9 – Camadas VGGFace. (Fonte: JAWOREK-KORJAKOWSKA et al., 2019)



Como camadas do modelo VGGFace, tem as camadas convolucionais - neste caso cinco blocos -, os quais são sempre seguidos da camada de *Pooling*, a qual, possui o objetivo de sub-amostrar a imagem de entrada para diminuir a carga computacional, o uso de memória e o número de parâmetros. Além disso, após as operações de convolução, tem-se as camadas com as funções de ativação, as quais promovem a não-linearidades ao modelo, proporcionando que a rede aprenda qualquer tipo de característica. E, por fim a camada *Fully Connected* a qual

transforma os parâmetros aprendidos por todos os neurônios da camada anterior, em N saídas que representam a quantidade de classes desejada como saída do modelo (GÉRON, 2019). Todas essas características citadas podem ser observadas na Figura 10.

Figura 10 – Camadas VGGFace. (Fonte: ANALYTICS VIDHYA)



2.5 Técnicas *anti-spoofing*

Com a intenção de detectar possíveis fraudes em sistemas de autenticação facial, há alguns anos, diversas pesquisas tem sido desenvolvidas para melhorar o desempenho das técnicas utilizadas até o momento. Uma abordagem observada por PAN et al. (2007) é a identificação da utilização de imagens estáticas para fraudar sistemas por meio da movimentação dos olhos, retirando do processo de verificação vários frames em poucos segundos e comparando-os para poder realizar a classificação - Figura 11.

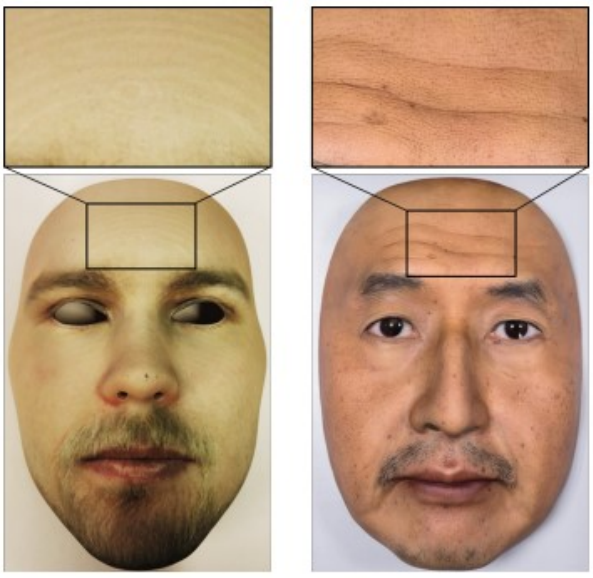
Figura 11 – Técnica *anti-spoofing* de identificação de movimentação dos olhos. (Fonte: PAN, 2007)

Utilizando-se dos mesmos princípios de extração de partes do rosto, SINGH et al. (2017) realizou a verificação da autenticidade de imagens por meio da movimentação da boca, afim de confirmar a existência de vida por trás dos aparelhos - Figura 12.

Figura 12 – Técnica *anti-spoofing* de identificação de movimentação da boca. (Fonte: SINGH et al., 2017)

Além disso, outro método proposto foi abordado por LI et al. (2016) o qual utilizou de técnicas de processamento de imagem para detectar o fluxo sanguíneo presente na face humana e assim coibir tentativas de fraude - Figura 13.

Figura 13 – Técnica *anti-spoofing* de identificação do fluxo sanguíneo. (Fonte: LI et al., 2016)



Por outro lado, há também estudos que buscam aprimorar as técnicas de aprendizado profundo, que, no sistema convencional de autenticação, é utilizada para classificar a identidade da pessoa, afim de extrair características das imagens e classificá-las em reais ou falsas. Como observado anteriormente, SOUZA (2019) e DESAI (2019), buscaram aplicar diferentes abordagens de aprendizado de máquina como redes neurais convolucionais (CNNs), máquina

Boltzmann restrita (RBM), redes neurais temporais e redes neurais recorrentes afim de comparar o desempenho de cada arquitetura quando expostos a faces falsas.

3 MATERIAIS

3.1 Google Colaboratory

Para a avaliação das aplicações realizadas neste trabalho, utilizou-se a plataforma do Google Colaboratory como ambiente de processamento. Segundo o Google, o Colaboratory (Colab), é um ambiente desenvolvido pelo Google Research no qual é possível que qualquer pessoa implemente um código em Python dentro do navegador. O Colab é uma solução desenvolvida especialmente para aplicações de aprendizado de máquina e análise de dados com o uso de hardware acelerado para Tensorflow, entre outras soluções, com GPU e TPU, na qual utiliza-se um Jupyter Notebook.

Além disso, segundo o Google, o serviço do Google Colaboratory é fornecido de forma gratuita com os recursos operacionais já pré configurados, incluindo GPUs. As GPUs disponível no ambiente de processamento variam entre Nvidia K80s, T4s, P4s e P100s. Para este trabalho utilizou-se também uma memória RAM de 12.72GB disponíveis e um disco de 70GB.

3.2 Tensorflow

Como ambiente pré configurado, tem-se a aplicação TensorFlow em Python, a qual é uma biblioteca de código aberto empregada para projetos voltados para aprendizado de máquina. A solução do TensorFlow foi originalmente desenvolvida pela equipe Google Brain, a qual empregou esta biblioteca em pesquisas voltadas para aprendizado de máquina e redes neurais profundas; atualmente a utilização do TensorFlow permite que os estudos relacionados a Inteligência Artificial possam ser empregados de forma fácil e rápida, tanto em Python, quanto em C++.

O TensorFlow é um sistema de aprendizado de máquina que opera em grande escala e em ambientes heterogêneos. O TensorFlow utiliza-se de grafos de fluxo de dados para representar a computação, estado compartilhado e as operações que alteram esse estado. Este processo é criado para mapear os nós de um gráfico de fluxo de dados em muitas máquinas ou em um *cluster*, incluindo CPUs *multicore*, GPUs e também no *Tensor Processing Units* (TPUs). Esta arquitetura fornece a flexibilidade para o desenvolvedor de aplicativos, pois, enquanto nos servidores de parâmetros projeta-se o gerenciamento de estado integrado ao sistema, o TensorFlow permite que

os desenvolvedores experimentem novas otimizações e algoritmos de treinamento (GOOGLE BRAIN, 2016).

3.3 Keras

Como solução integrada ao TensorFlow, tem-se o Keras, uma API (Interface de programação de aplicações) desenvolvida por François Chollet, com código aberto, a qual permite a utilização do TensorFlow por meio de uma interface acessível e produtiva para a resolução de problemas de aprendizado de máquina, minimizando as ações necessárias por meio do usuário e emitindo erros de forma direta e transparente.

Com a criação do Keras, tornou-se possível o maior aproveitamento da escalabilidade e dos recursos existentes na plataforma do TensorFlow, permitindo o usuário executar o Keras em TPU ou também em grandes clusters de GPUs. Além disso o Keras possui várias funções voltadas para a construção de partes necessárias em projetos de redes neurais, como camadas, funções de perda, funções de ativação, otimizadores, podendo empregá-las também em redes neurais convolucionais e recorrentes.

3.4 Dataset facial

Como base de imagens faciais voltadas para aplicações anti-spoofing, tem-se a NUAA (*Nanjing University of Aeronautics and Astronautics*), uma das base mais utilizadas atualmente neste campo de estudo. A base NUUA consiste em um relação de 12620 imagens de 15 pessoas diferentes, divididas em duas categorias, imagens reais - denominada *ClientFace* - e imagens falsas - denominada *ImposterFace*. As imagens foram obtidas, pelo grupo de pesquisadores da Universidade de Nanjing, por meio da utilização de *webcams*, para as fotos reais, e, para fotos falsas, com fotos das fotos. Além disso, as imagens foram obtidas com pessoas de diferentes idades e gêneros e também em diferentes posições.

Na Figura 14 pode-se observar uma imagem real e na Figura 15 uma imagem falsa. É possível concluir visualmente as diferenças existentes entre as imagens considerando aspectos como brilho, textura e cor.

Figura 14 – Imagem real. (Fonte: NUAA)

Figura 15 – Imagem falsa. (Fonte: NUAA)

4 MÉTODOS

Neste capítulo serão apresentados os métodos utilizados no desenvolvimento desta monografia, abordando desde as características e funções utilizadas na rede neural convolucional, as métricas de avaliação dos resultados obtidos, as técnicas utilizadas, até o método proposto, em comparação com a literatura, e as particularidades dos testes realizados.

4.1 Características do modelo

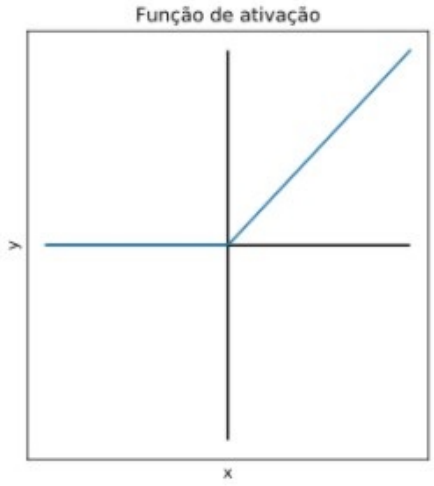
4.1.1 Funções de ativação

Como funções de ativações, foram utilizadas nas camadas intermediárias e não convolucionais do modelo VGGFace as funções *Rectified Linear Unit* (ReLU) e *softmax*.

A função de ativação *Rectified Linear Unit*, mais conhecida como ReLU, é uma forma de função de ativação usada comumente em modelos de aprendizado profundo. Em essência, a função retorna zero se receber uma entrada negativa e se receber um valor positivo, a função retornará o mesmo valor positivo. Os benefícios de usar a função ReLU são que sua simplicidade a torna uma função relativamente barata de calcular. Como não há matemática complicada, o modelo pode ser treinado e executado em um tempo relativamente curto. Da mesma forma, ele converge mais rápido, o que significa que a inclinação não se estabiliza conforme o valor de X fica maior (GOODFELLOW, 2016).

$$f(x) = max(0, x) \tag{4.1}$$

Figura 16 – Rectified Linear Unit (ReLU). (Fonte: EXPERT ACADEMY)



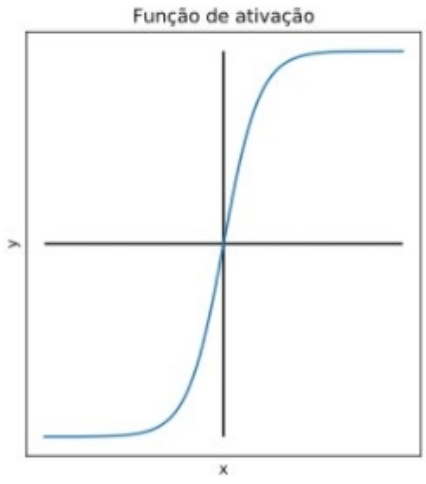
.

A função de ativação *softmax*, ou função exponencial normalizada, é uma função logística desenvolvida para aplicações de dimensões múltiplas. Esta função é utilizada tanto em problemas que envolvem regressão logística multinomial quanto como a última função de ativação de uma rede neural para normalizar a saída de uma rede para uma distribuição de probabilidade sobre as classes de saída previstas (GOODFELLOW, 2016).

O funcionamento da função *softmax* - Equação 4.2 - é dado pelo cálculo de uma pontuação para o vetor z para cada classe K, para o qual é realizado o cálculo do exponencial, e, posteriormente, a normalização, obtendo como resultado a probabilidade do vetor pertencer a cada classe. Sendo assim, antes da aplicação da função *softmax*, alguns componentes do vetor podem possuir valores negativos ou maiores que um e a soma deles podem não somar um. Após a aplicação da função, cada componente presente na entrada possuirá um valor no intervalo $(0, 1)$, e os componentes somarão um, de forma que possam ser interpretados como probabilidades, e também correspondendo, os maiores componentes de entrada, as maiores probabilidades (GOODFELLOW, 2016).

$$\Theta(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}, \text{para} j = 1, ..., K. \tag{4.2}$$

Figura 17 – Softmax. (Fonte: EXPERT ACADEMY)



.

4.1.2 Função de Otimização

Como função de custo para otimização do modelo, utilizou-se a função do Gradiente Descendente Estocástico, ou SGD (*stochastic gradient descent*, em inglês), para a qual, o algoritmo é utilizado para calcular o gradiente da função de perda da rede em relação a cada peso individual na rede. Para cada passagem para frente pela rede, retorna-se uma certa função de perda parametrizada, e utiliza-se cada um dos gradientes criados para cada um dos pesos, multiplicando-os por uma certa taxa de aprendizado, para mover os pesos em qualquer direção que seu gradiente esteja apontando(BOTTOU, 2010).

O SGD diferencia-se dos outros gradientes existentes por escolher aleatoriamente uma instância no conjunto de treinamento em cada etapa e calcular os gradientes para uma amostra dos dados, procurando mínimos locais. Desta maneira, o algoritmo do SGD acaba possuindo uma velocidade de treinamento maior, por realizar cálculos somente em uma instância por etapa, e permite, também, o treinamento de um grande conjunto de dados, pois somente uma instância precisa ficar na memória por iteração (GÉRON, 2019).

4.1.3 Função de perda

Como função de perda utilizou-se o método denominado *Categorical Cross-Entropy*. A *Cross-Entropy* (CE) é um método heurístico utilizado para resolver problemas de otimização combinatória, calculando-se a diferença entre duas distribuições de probabilidade em relação ao mesmo conjunto de eventos (MANNOR et al, 2005). A função de perda de CE é quase a única escolha para tarefas de classificação na prática. Seu uso predominante é apoiado teoricamente

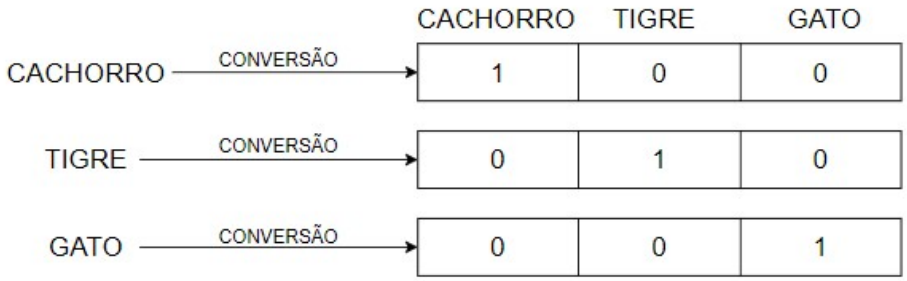
por sua associação com a minimização da divergência de probabilidade entre a distribuição empírica de um conjunto de dados e a confiança do classificador para o conjunto de dados (NAR et al, 2019).

Desta maneira a função de perda de *Cross-Entropy*, mede o desempenho de um modelo de classificação cuja saída é um valor de probabilidade entre 0 e 1, a qual é dada pela Equação 4.3.

$$CE = -\sum_{i=1}^{L} y_i . \log \vec{y}_i, \text{ sendo L o cumprimento da vetor de saída} \tag{4.3}$$

Para modelos de classificação com N classes diferentes, mas não mutuamente exclusivas, utiliza-se o método denominado *Categorical Cross-Entropy*, no qual cada classe é convertida para um vetor $1XN$ possuindo valores unitários para a posição que representa sua classe e valores nulos para as outras classe. Na Figura 18 tem-se um exemplo desta aplicação.

Figura 18 – Exemplo *Categorical Cross-Entropy*. (Fonte: Autor)



.

4.1.4 Métricas de avaliação

Como métricas de avaliação do desempenho das redes neurais treinadas, utilizou-se desde a acurácia obtida a cada época de treinamento, como a perda em cada iteração. Além disso, métricas como precisão, revocação, *F1-score*, a área sob a curva ROC (AUC) e matriz de confusão foram observadas por se tratar de um modelo de classificação com diversas classes.

A acurácia, pode ser definida como a quantidade de amostras que foram classificadas corretamente, tanto positivas quanto negativas. No entanto, é necessário atentar-se em relação a essa métrica pois como não há um padrão na quantidade de imagens pertencentes a cada classe existente no modelo treinado e há um número considerável de classes, sempre será observado

mais segmentos da classe negativa. Dessa maneira mesmo fixando um valor para as predições da classe negativas, a acurácia apresentaria um valor alto (MARQUES, 2017).

A precisão é definida como a acurácia das previsões positivas, ou seja, a quantidade de segmentos classificados positivos que realmente pertencem a classe selecionada (Equação 4.4). Já revocação, ou *Recall*, é a sensibilidade do modelo, a qual expressa a taxa de verdadeiros positivos, isto é, a quantidade de classes positivas que foram corretamente identificadas (Equação 4.5). Para ambas equações, tem-se que TP é o número de verdadeiros positivos, FP é o número de falsos positivos e FN é o número de falsos negativos (GÉRON, 2019).

$$\text{precisão} = \frac{TP}{TP + FP} \tag{4.4}$$

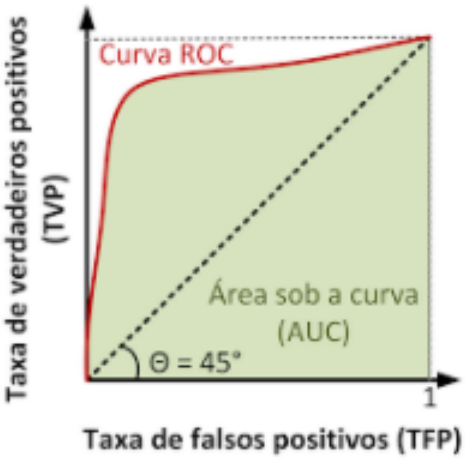
$$\text{recall} = \frac{TP}{TP + FN} \tag{4.5}$$

Tem-se também o *F1-score*, que, pode ser definida como uma métrica que combina os valores de precisão e revocação utilizando o cálculo da média harmônica entre as duas outras métricas - Equação 4.6 (GÉRON, 2019).

$$\text{F1} = \frac{2}{\frac{1}{\text{precisão}} + \frac{1}{\text{revocação}}} \tag{4.6}$$

A curva ROC (traduzida do inglês como características operacionais do receptor) representa a taxa de verdadeiros positivos (revocação) em relação a taxa de falsos positivos. Os valores na curva ROC variam em cada eixo de zero até um, sendo considerado um bom classificador aquele que possui um comportamento distante da linha pontilhada - Figura 19 - a qual representa um classificador aleatório. Já o valor da Área sob a curva (AUC) é utilizado para a comparação entre classificadores, sendo este considerado satisfatório para valores próximos a um (GÉRON, 2019).

Figura 19 – Curva ROC com representação do cálculo AUC. (Fonte: REBELLO, 2020)



Além disso utilizou-se o cálculo da matriz de confusão, para a qual obtêm-se a quantidade de vezes em que uma classe A - valores nas linhas da matriz - foi classificada como uma classe B - valores nas colunas da matriz. Desta maneira, cada linha da matriz representa a classe verdadeira que se deseja classificar e, cada coluna da matriz representa a classe em que o dado foi

classificado, ou seja, para se obter bons resultados, deve-se possuir maiores valores na diagonal principal da matriz (GÉRON, 2019).

4.2 Técnicas utilizadas

4.2.1 *Fine-tuning*

A técnica de *Fine-tuning*, em geral, significa fazer pequenos ajustes em um processo para obter a saída ou desempenho desejado. O *Fine-tuning* do aprendizado profundo envolve o uso de pesos de um algoritmo de aprendizado profundo anterior para programar outro processo de aprendizado profundo semelhante. Pesos são usados para conectar cada neurônio em uma camada a cada neurônio na próxima camada da rede neural. O processo de *Fine-tuning* diminui significativamente o tempo necessário para programar e processar um novo algoritmo de aprendizado profundo, pois ele já contém informações vitais de um algoritmo de aprendizado profundo pré-existente.

Desta maneira, como o conjunto de 2.6MM de imagens utilizadas para treinar o modelo VGGFace é muito maior em relação ao conjunto de imagens utilizadas neste trabalho, entre 6 mil reais e 6 mil falsas, a utilização dos pesos pré-treinados pela VGGFace tornou-se necessária, sendo possível com a implementação da técnica de *Fine-tuning*.

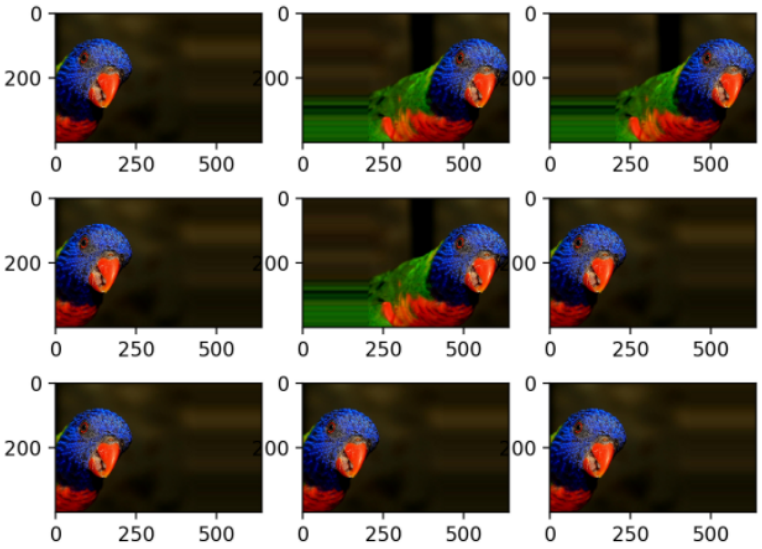
4.2.2 *Data Augmentation*

A técnica de *Data Augmentation*, pode ser definida como uma técnica de regularização a qual consiste na geração de novos objetos para treinamento do modelo proposto, por meio dos objetos já existentes, e, desta maneira, aumentando, consideravelmente, o tamanho da base de dados utilizada para treinamento e reduzindo o sobre-ajuste do modelo (GÉRON, 2019).

Como exemplo, tem-se as Figuras 20 e 21 nas quais, a primeira representa a imagem original utilizada para classificação no modelo, e, a segunda, as imagens geradas para a implementação da técnica de *Data Augmentation*, utilizando técnicas de rotação de imagens e alteração de cores.

Figura 20 – Exemplo *Data Augmentation* - Imagem original. (Fonte: BROWNLEE, 2019)

Figura 21 – Exemplo *Data Augmentation* - Imagens geradas. (Fonte: BROWNLEE, 2019)



Desta maneira, a avaliação da aplicação da técnica de *Data Augmentation*, por meio da adição de imagens com filtro Gaussiano passa-alta e do descritor de textura LBP, é apropriada para a detecção de fraude em reconhecimento facial. A extração de diferentes características das imagens agrega valor ao modelo classificatório, realizando diferentes comparações entre as imagens reais e falsas, considerando, neste caso, zonas de alta frequência e distribuição da intensidade de cinza em relação aos pixels vizinhos.

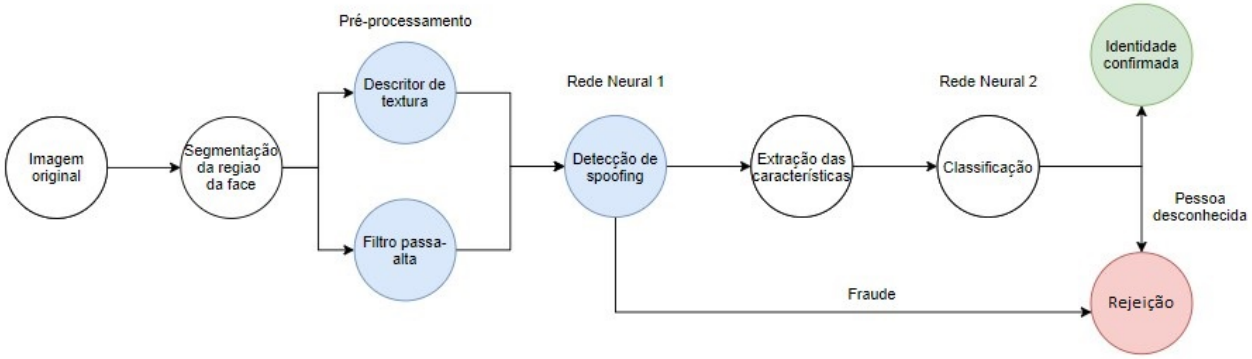
4.3 Método proposto

Observa-se que o método convencional utilizado em sistemas biométricos - como visto na Figura 1 - realiza a verificação da identidade da face. Dessa maneira, torna-se necessária a avaliação do desempenho da atual abordagem utilizada e a implementação de alternativas ao método convencional de reconhencimento facial, avaliando o comportamento da rede neural para a classificação de imagens reais e falsas, verificando assim a autenticidade das imagens.

Além disso, os trabalhos recentes propostos por SOUZA (2019) e DESAI (2019) investigaram diferentes procedimentos de detecção de fraude por meio da avaliação de diversas abordagens de arquiteturas de redes neurais artificiais, porém, não foi avaliado o desempenho do aplicação quando exposta a diferentes técnicas de pré-processamento das imagens faciais, por meio, por exemplo, da utilização conjunta de descritores de textura e filtros passa-alta.

Sendo assim, a proposta deste trabalho para a autenticação biométrica por meio da face, seria necessário, primeiramente, a realização de um pré-processamento das imagens faciais, por meio de descritor de textura e filtro passa-alta (LBP e filtro Gaussiano, respectivamente) e utilizando técnicas de *Fine-tuning* e *Data Augmentation*. Posteriormente, torna-se preciso a implementação de um modelo classificatório para avaliar a autenticidade da imagem utilizada para a autenticação, o qual será avaliado neste trabalho, e outro para identificação da identidade da face, como já utilizado na arquitetura convencional, diferenciando-se do sistema biométrico atual, como pode-se observar na Figura 22.

Figura 22 – Proposta de cenário ideal para autenticação biométrica facial (Fonte: Autor)



.

4.4 Testes realizados

Pode-se dividir o conjunto de testes em duas etapas: a primeira com o intuito de verificar a abordagem convencional do reconhecimento facial, utilizando imagens rotuladas com a identidade das pessoas e expondo o modelo a imagens falsas; a segunda visando analisar o desempenho do modelo quando exposto somente a classificação de imagens reais e falsas.

4.4.1 *Fine-tuning* para quinze classes de imagens reais

Com o intuito de avaliar o modelo pré-treinado da VGGFace realizou-se a técnica de *Fine-tuning* no modelo, retreinando, desta maneira, as quatro últimas camadas com as seguintes características:

- *Batch Size*: 32
- Número de épocas: 100
- Parâmetros treináveis: 13,109,250
- Parâmetros não treináveis: 14,714,688
- Quantidade de classes: 15
- Característica(s) das imagens: somente imagens reais
- Métricas utilizadas: Acurácia, Perda, Precisão, Revocação, Área sob a curva ROC e Matriz de Confusão

Para este teste foi utilizado por classe a quantidade de imagens que pode-se observar na Figura 23, todas como o exemplo presente na Figura 24.

Figura 23 – Quantidade de imagens utilizadas no teste de *Fine-tuning* para quinze classes de imagens reais

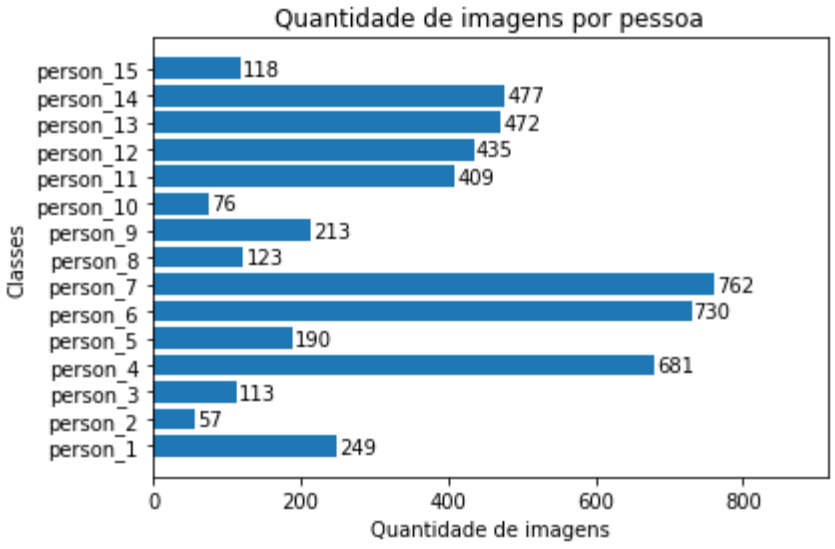


Figura 24 – Exemplo de imagem real utilizada no teste de *Fine-tuning* para quinze classes de imagens reais


.

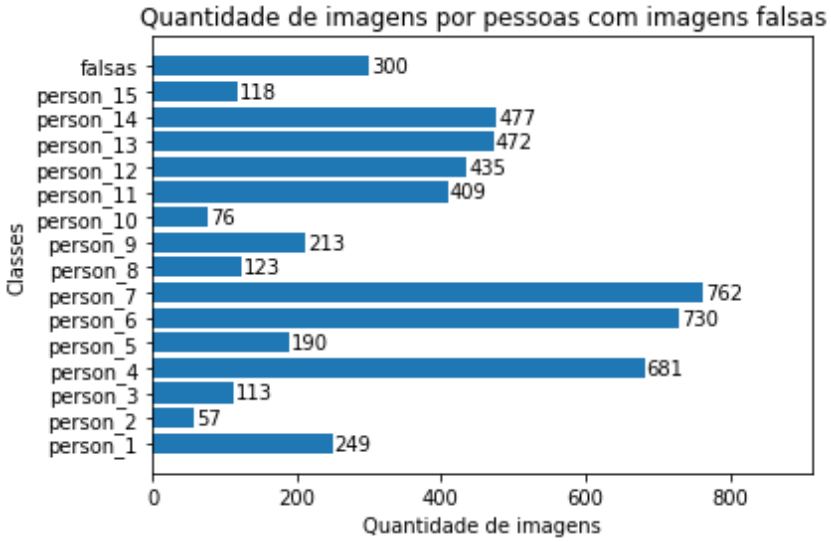
4.4.2 *Fine-tuning* para quinze classes de imagens reais e uma classe falsa

Para avaliar como o modelo pré-treinado da VGGFace comporta-se em relação a imagens falsas, realizou-se a técnica de *Fine-tuning* no modelo, retreinando, desta maneira, as quatro últimas camadas com as seguintes características:

- *Batch Size*: 32
- Número de épocas: 100
- Parâmetros treináveis: 13,109,250
- Parâmetros não treináveis: 14,714,688
- Quantidade de classes: 16
- Característica(s) das imagens: 15 classes reais e 1 classe falsa
- Métricas utilizadas: Acurácia, Perda, Precisão, Revocação, Área sob a curva ROC e Matriz de Confusão

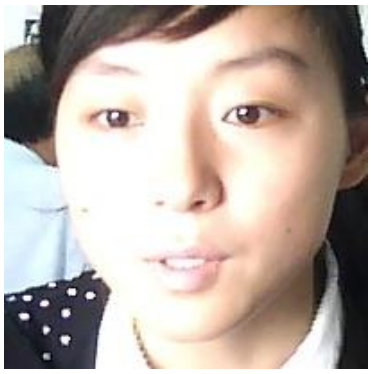
Para este teste foi utilizado por classe a quantidade de imagens que pode-se observar na Figura 25, selecionando, de forma aleatória, em torno de vinte imagens falsas relacionadas as pessoas presentes em cada classe real - 300 imagens falsas. Na Figura 26 pode-se observar um exemplo de uma imagem real e na Figura 27 um exemplo de uma imagem falsa.

Figura 25 – Quantidade de imagens utilizadas no teste de *Fine-tuning* para quinze classes de imagens reais e uma classe falsa



.

Figura 26 – Exemplo de imagem real utilizada no teste de *Fine-tuning* para quinze classes de imagens reais e uma classe falsa



.

Figura 27 – Exemplo de imagem falsa utilizada no teste de *Fine-tuning* para quinze classes de imagens reais e uma classe falsa


.

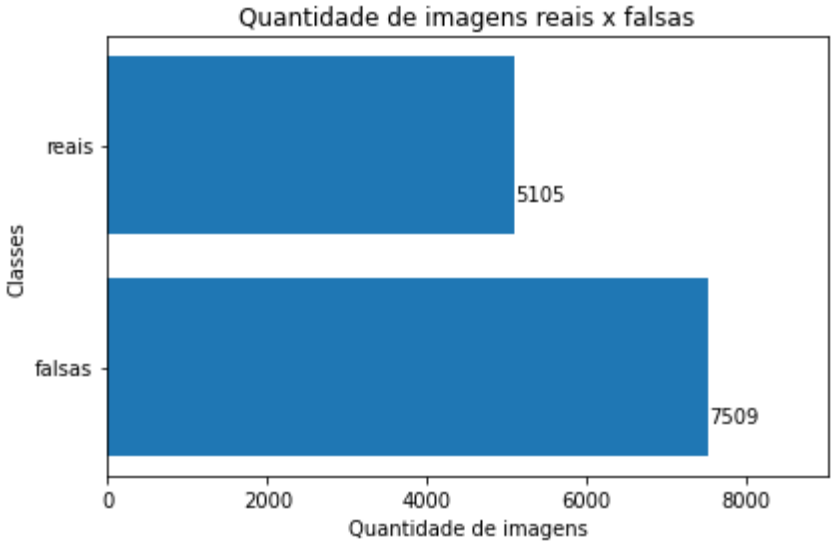
4.4.3 *Fine-tuning* para uma classe real e uma classe falsa

Com o intuito de avaliar a forma como o modelo pré-treinado da VGGFace comporta-se na classificação somente de imagens reais e imagens falsas, realizou-se a técnica de *Fine-tuning* no modelo, retreinando, desta maneira, as quatro últimas camadas com as seguintes características:

- *Batch Size*: 128
- Número de épocas: 100
- Parâmetros treináveis: 13,109,250
- Parâmetros não treináveis: 14,714,688
- Quantidade de classes: 2
- Característica(s) das imagens: 1 classe real e 1 classe falsa
- Métricas utilizadas: Acurácia, Perda, Precisão, Revocação, Área sob a curva ROC e Matriz de Confusão

Para este teste foi utilizado por classe a quantidade de imagens que pode-se observar na Figura 28. Na Figura 29 pode-se observar um exemplo de uma imagem real e na Figura 30 um exemplo de uma imagem falsa.

Figura 28 – Quantidade de imagens utilizadas no teste de *Fine-tuning* para uma classe real e uma classe falsa



.

Figura 29 – Exemplo de imagem real utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa

.

Figura 30 – Exemplo de imagem falsa utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa


.

4.4.4 *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

Para avaliar a forma como o modelo pré-treinado da VGGFace comporta-se na classificação somente de imagens reais e imagens falsas, realizou-se a técnica de *Fine-tuning* no modelo, retreinando, desta maneira, as quatro últimas camadas com as seguintes características:

- *Batch Size*: 128
- Número de épocas: 100
- Parâmetros treináveis: 13,109,250
- Parâmetros não treináveis: 14,714,688
- Quantidade de classes: 2
- Característica(s) das imagens: 1 classe real e 1 classe falsa com descritor de textura LBP
- Métricas utilizadas: Acurácia, Perda, Precisão, Revocação, Área sob a curva ROC e Matriz de Confusão

Para este teste foi utilizado também a técnica de *Data Augmentation* em ambas as classes - real e falsa - utilizando, além das imagens originais processadas no teste anterior, os resultados do processo de descrição com o descritor de textura LBP. Desta maneira, a quantidade de imagens por classe utilizada neste treinamento pode ser observada na Figura 31. Na Figura 32 pode-se observar um exemplo de uma imagem real e na Figura 33 um exemplo de uma imagem falsa.

Figura 31 – Quantidade de imagens utilizadas no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

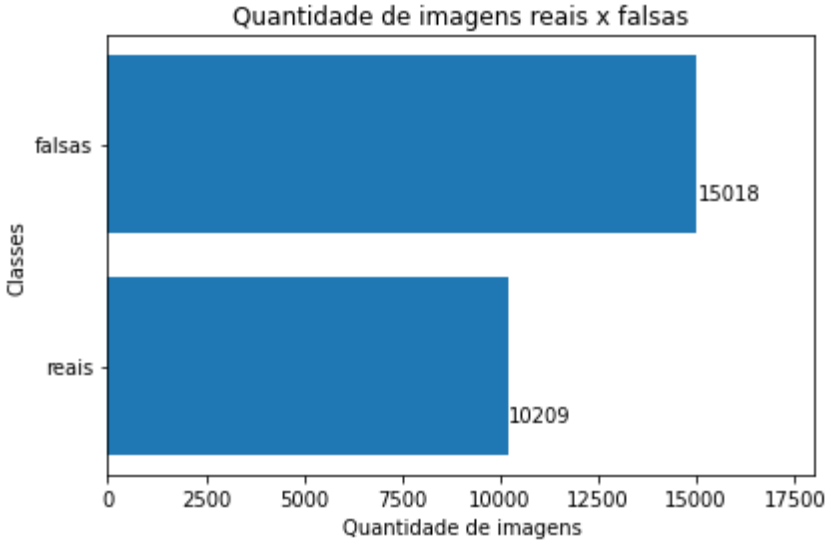
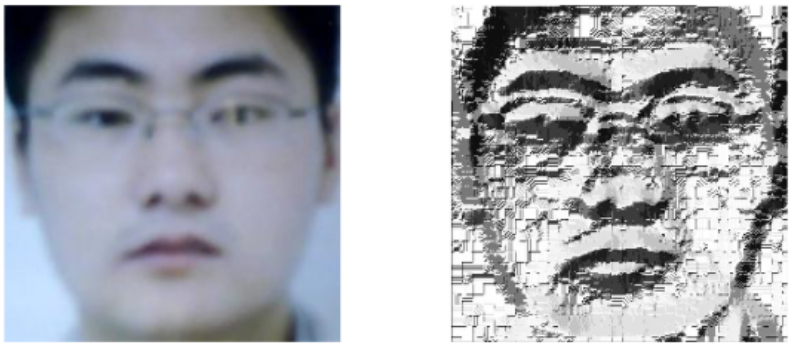


Figura 32 – Exemplo de imagens reais utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP. Esquerda: Original - Direita: Mapa de textura gerado pelo LBP

Figura 33 – Exemplo de imagens falsas utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP. Esquerda: Original - Direita: Mapa de textura gerado pelo LBP



.

4.4.5 *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

Com o intuito de avaliar a forma como o modelo pré-treinado da VGGFace comporta-se na classificação somente de imagens reais e imagens falsas, realizou-se a técnica de *Fine-tuning* no modelo, retreinando, desta maneira, as quatro últimas camadas com as seguintes características:

- *Batch Size*: 128
- Número de épocas: 100
- Parâmetros treináveis: 13,109,250
- Parâmetros não treináveis: 14,714,688
- Quantidade de classes: 2
- Característica(s) das imagens: 1 classe real e 1 classe falsa com filtro Gaussiano aplicado
- Métricas utilizadas: Acurácia, Perda, Precisão, Revocação, Área sob a curva ROC e Matriz de Confusão

Neste teste foi utilizado também a técnica de *Data Augmentation* em ambas as classes - real e falsa - utilizando, além das imagens originais processadas anteriormente, os resultados do processo de filtragem com o filtro Gaussiano. Desta maneira, a quantidade de imagens por classe

utilizada neste treinamento pode ser observada na Figura 34. Na Figura 35 pode-se observar um exemplo de uma imagem real e na Figura 36 um exemplo de uma imagem falsa.

Figura 34 – Quantidade de imagens utilizadas no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano.

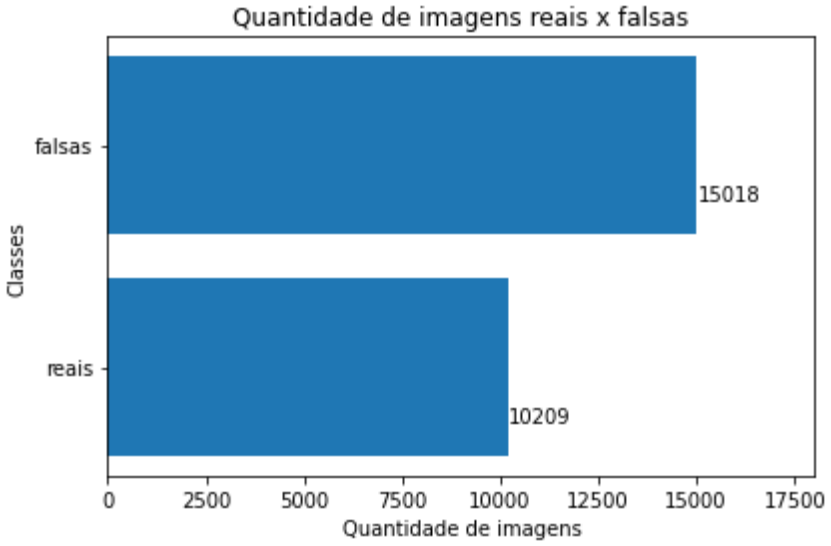


Figura 35 – Exemplo de imagens reais utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano. Esquerda: Original - Direita: Filtro Gaussiano

Figura 36 – Exemplo de imagens falsas utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano. Esquerda: Original - Direita: Filtro Gaussiano

.

4.4.6 *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* com ambos os filtros

Afim de avaliar a forma como o modelo pré-treinado da VGGFace comporta-se na classificação somente de imagens reais e imagens falsas, realizou-se a técnica de *Fine-tuning* no modelo, retreinando, desta maneira, as quatro últimas camadas com as seguintes características:

- *Batch Size*: 128
- Número de épocas: 100
- Parâmetros treináveis: 13,109,250
- Parâmetros não treináveis: 14,714,688
- Quantidade de classes: 2
- Característica(s) das imagens: 1 classe real e 1 classe falsa com filtros LBP e Gaussiano aplicados
- Métricas utilizadas: Acurácia, Perda, Precisão, Revocação, Área sob a curva ROC e Matriz de Confusão

Para este último teste foi utilizado também a técnica de *Data Augmentation* em ambas as classes - real e falsa - utilizando, além das imagens originais processadas no teste anterior, os resultados do processo de filtragem com o filtro Gaussiano e o descritor de textura LBP. Desta

maneira, a quantidade de imagens por classe utilizada neste treinamento pode ser observada na Figura 37. Na Figura 38 pode-se observar um exemplo de uma imagem real e na Figura 39 um exemplo de uma imagem falsa.

Figura 37 – Quantidade de imagens utilizadas no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation*

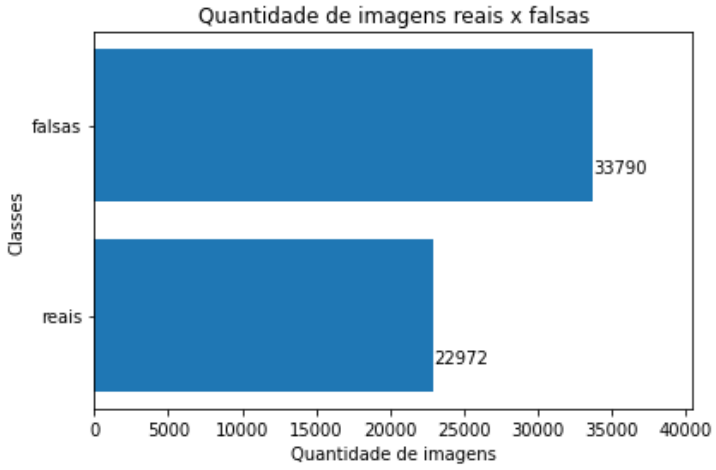


Figura 38 – Exemplo de imagens reais utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation*. Esquerda: Original - Centro: Filtro Gaussiano - Direita: Mapa de textura gerado pelo LBP

Figura 39 – Exemplo de imagens falsas utilizada no teste de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation*. Esquerda: Original - Centro: Filtro Gaussiano - Direita: Mapa de textura gerado pelo LBP

5 RESULTADOS E DISCUSSÕES

5.1 Resultados

5.1.1 *Fine-tuning* para quinze classes de imagens reais

Como resultado do teste realizado considerando a técnica de *Fine-tuning* para quinze classes de imagens reais, encontram-se os seguintes resultados:

- Acurácia do modelo: Figura 40
- Perda do modelo: Figura 41
- Área sob a Curva ROC em função do número de épocas: Figura 42
- Matriz de confusão do modelo: Figura 43
- Relatório de classificação do modelo: Tabela 1

Figura 40 – Acurácia para o teste utilizando *Fine-tuning* para quinze classes de imagens reais

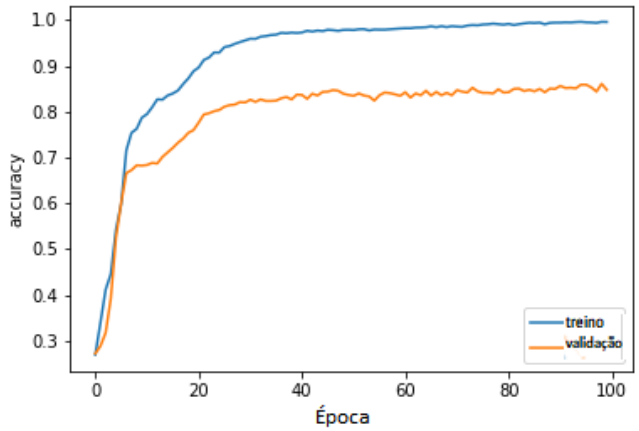


Figura 41 – Perda para o teste utilizando *Fine-tuning* para quinze classes de imagens reais

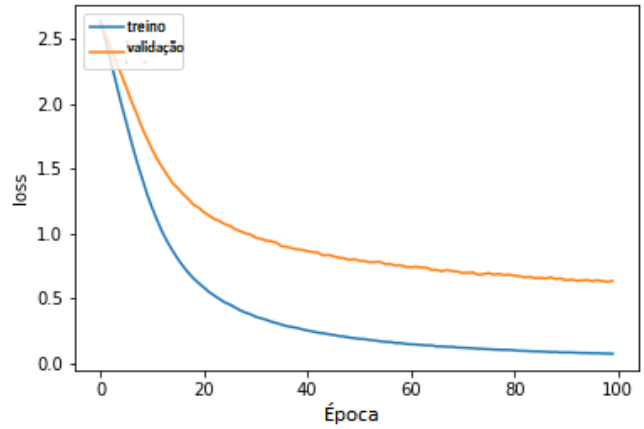


Figura 42 – Área sob a Curva ROC em função do número de épocas para o teste utilizando *Fine-tuning* para quinze classes de imagens reais

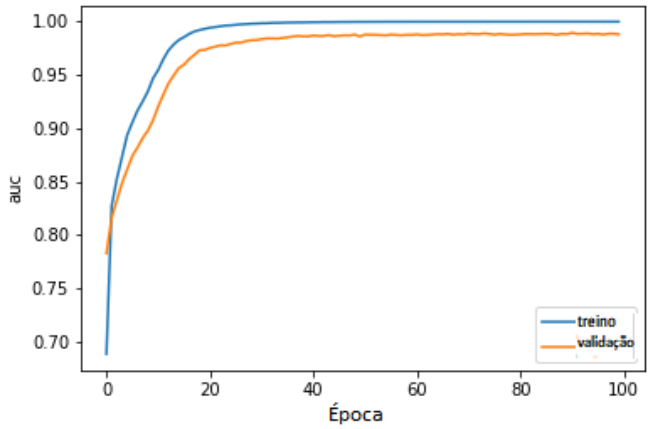
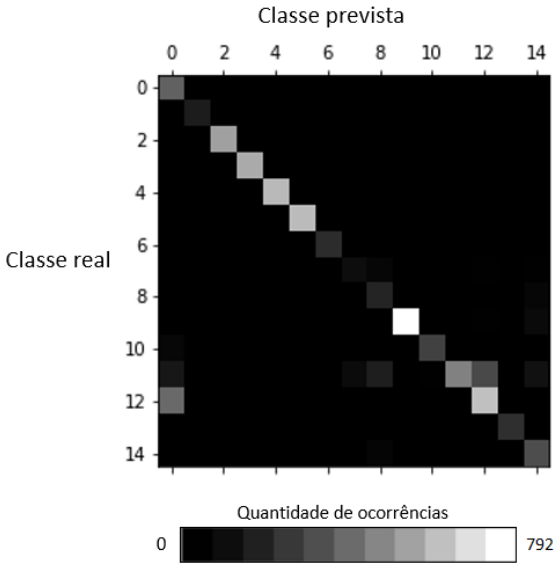


Figura 43 – Matriz de confusão para o teste utilizando *Fine-tuning* para quinze classes de imagens reais



.

Tabela 1 – Relatório de classificação para o teste utilizando *Fine-tuning* para quinze classes de imagens reais

	precisão	recall	f1-score
person_1	0,42	1,00	0,59
person_2	0,62	0,76	0,68
person_3	0,45	0,91	0,61
person_4	1,00	0,98	0,99
person_5	1,00	0,93	0,96
person_6	1,00	0,46	0,63
person_7	0,72	0,65	0,68
person_8	1,00	1,00	1,00
person_9	0,74	0,87	0,80
person_10	1,00	1,00	1,00
person_11	1,00	1,00	1,00
person_12	1,00	1,00	1,00
person_13	1,00	1,00	1,00
person_14	1,00	1,00	1,00
person_15	1,00	1,00	1,00
acurácia			0,85

5.1.2 *Fine-tuning* para quinze classes de imagens reais e uma classe falsa

Como desfecho do teste realizado considerando a técnica de *Fine-tuning* para quinze classes de imagens reais e uma classe falsa, encontram-se os seguintes resultados:

- Acurácia do modelo: Figura 44
- Perda do modelo: Figura 45
- Área sob a Curva ROC em função do número de épocas: Figura 46
- Matriz de confusão do modelo: Figura 47
- Relatório de classificação do modelo: Tabela 2

Figura 44 – Acurácia para o teste utilizando *Fine-tuning* para quinze classes de imagens reais e uma classe falsa

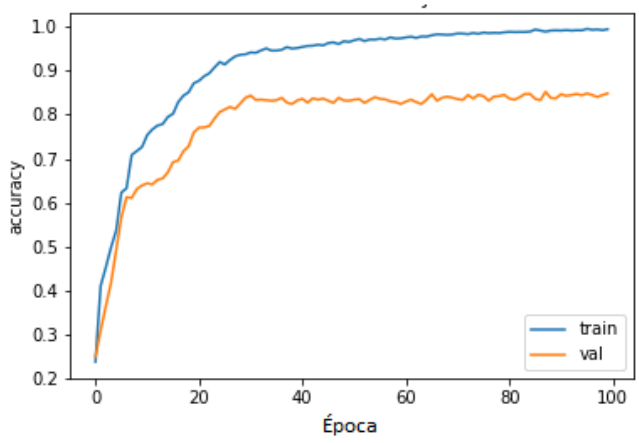


Figura 45 – Perda para o teste utilizando *Fine-tuning* para quinze classes de imagens reais e uma classe falsa

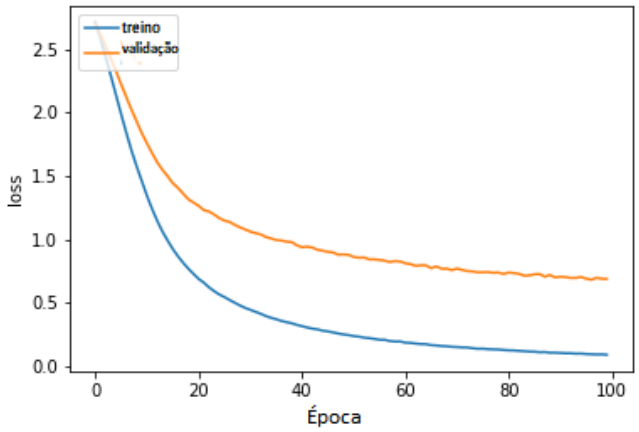


Figura 46 – Área sob a Curva ROC em função do número de épocas para o teste utilizando *Fine-tuning* para quinze classes de imagens reais e uma classe falsa

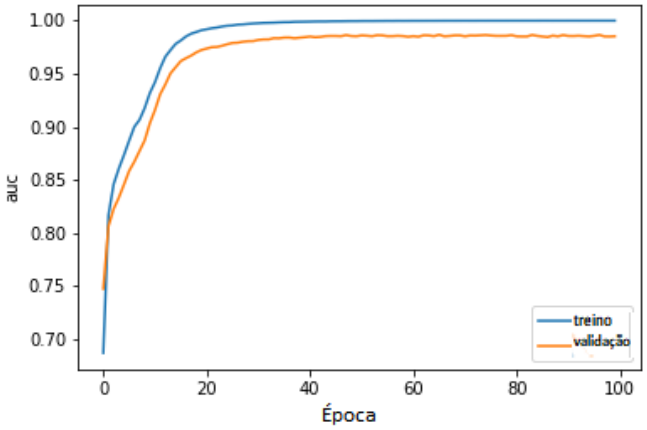
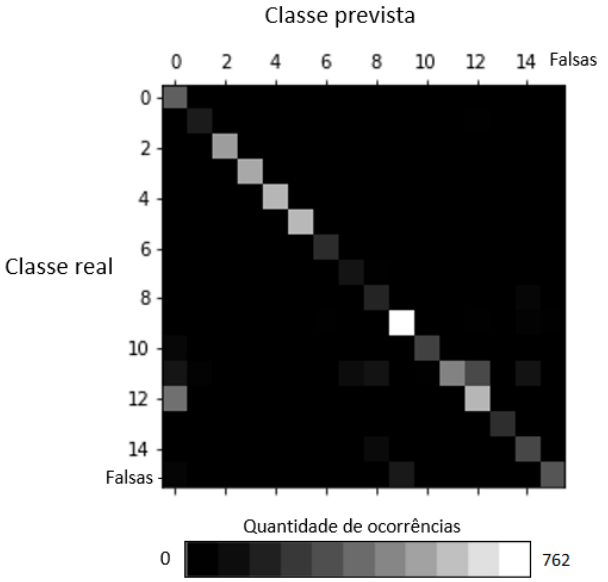


Figura 47 – Matriz de confusão para o teste utilizando *Fine-tuning* para quinze classes de imagens reais e uma classe falsa



.

Tabela 2 – Relatório de classificação para o teste utilizando *Fine-tuning* para quinze classes de imagens reais e uma classe falsa

	precisão	recall	f1-score
person_1	0,39	1,00	0,56
person_2	0,62	0,94	0,74
person_3	0,54	0,85	0,66
person_4	0,91	0,97	0,94
person_5	0,98	0,89	0,94
person_6	1,00	0,46	0,63
person_7	0,70	0,62	0,66
person_8	1,00	1,00	1,00
person_9	0,71	0,87	0,78
person_10	0,88	0,95	0,91
person_11	1,00	1,00	1,00
person_12	1,00	1,00	1,00
person_13	1,00	1,00	1,00
person_14	1,00	1,00	1,00
person_15	0,97	1,00	0,99
falsas	0,99	0,73	0,84
acurácia			0,84

5.1.3 *Fine-tuning* para uma classe real e uma classe falsa

Como desfecho do teste realizado considerando a técnica de *Fine-tuning* para uma classe real e uma classe falsa, encontram-se os seguintes resultados:

- Acurácia do modelo: Figura 48
- Perda do modelo: Figura 49
- Área sob a Curva ROC em função do número de épocas: Figura 50
- Matriz de confusão do modelo: Figura 51
- Relatório de classificação do modelo: Tabela 3

Figura 48 – Acurácia para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa

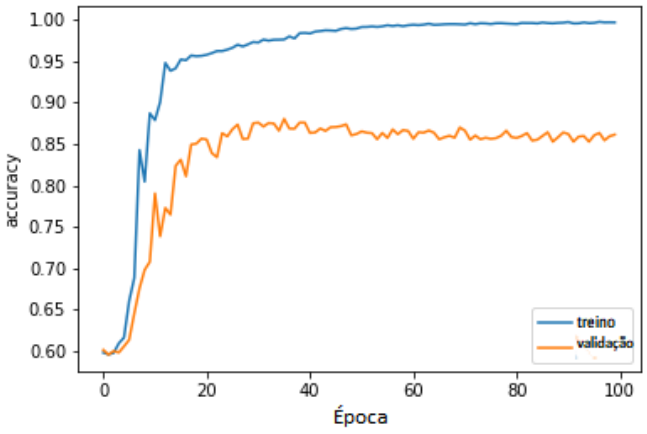


Figura 49 – Perda para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa

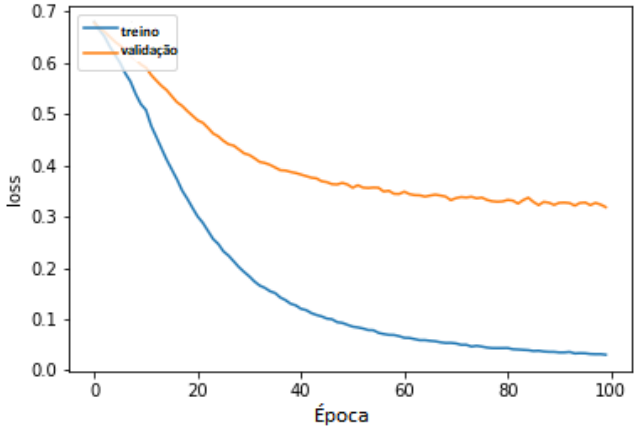


Figura 50 – Área sob a Curva ROC em função do número de épocas para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa

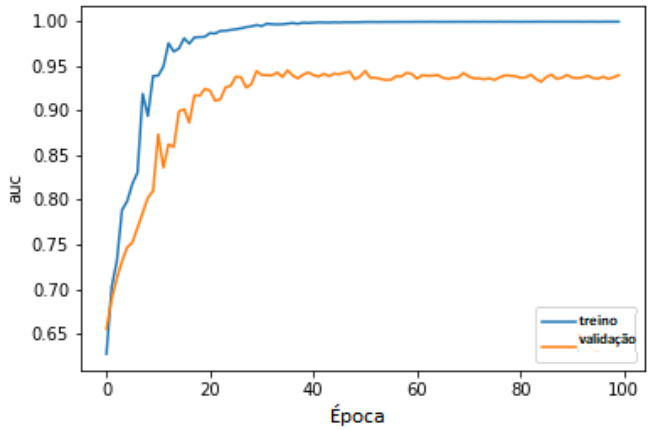


Figura 51 – Matriz de confusão para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa

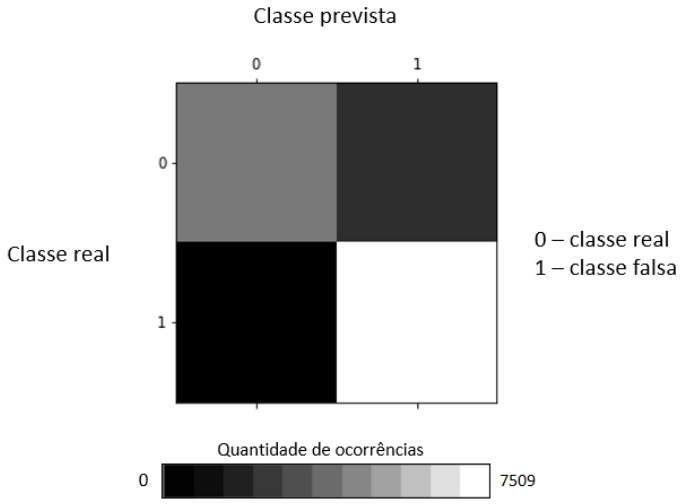


Tabela 3 – Relatório de classificação para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa

	precisão	recall	f1-score
reais	0,94	0,70	0,80
falsas	0,83	0,97	0,89
acurácia			0,86

5.1.4 *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

Como desfecho do teste realizado considerando a técnica de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP, encontram-se os seguintes resultados:

- Acurácia do modelo: Figura 52
- Perda do modelo: Figura 53
- Área sob a Curva ROC em função do número de épocas: Figura 54
- Matriz de confusão do modelo: Figura 55
- Relatório de classificação do modelo: Tabela 4

Figura 52 – Acurácia para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

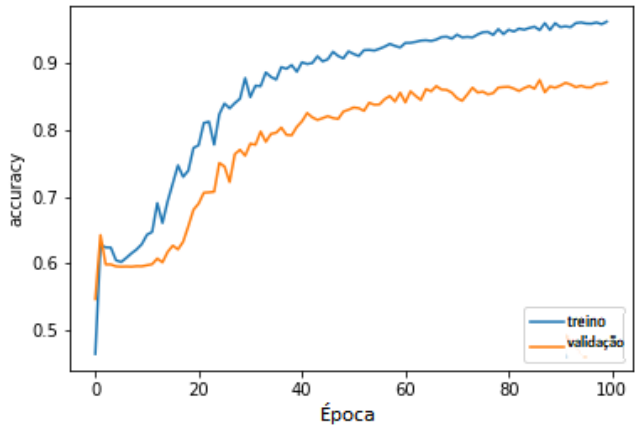


Figura 53 – Perda para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

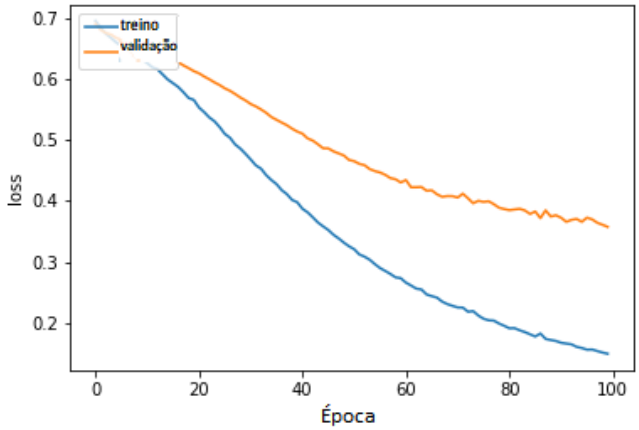


Figura 54 – Área sob a Curva ROC em função do número de épocas para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

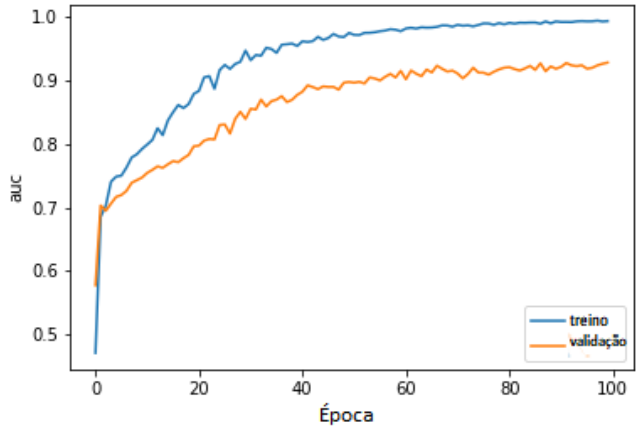
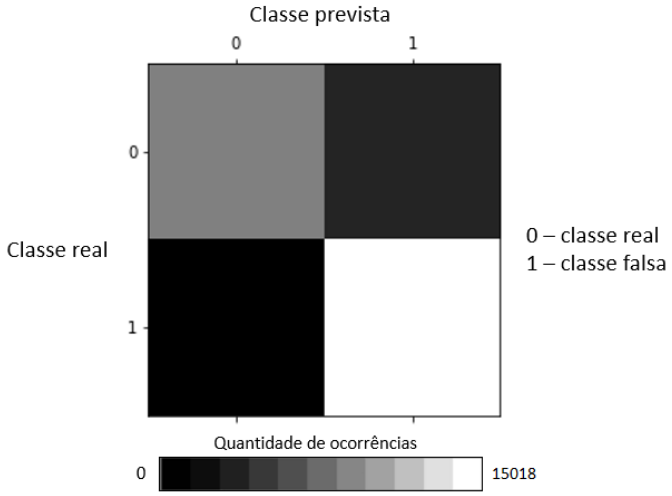


Figura 55 – Matriz de confusão para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP



.

Tabela 4 – Relatório de classificação para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando descritor de textura LBP

	precisão	recall	f1-score
real	0,92	0,74	0,82
falsa	0,84	0,96	0,90
acurácia			0,87

5.1.5 *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

Como desfecho do teste realizado considerando a técnica de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano, encontram-se os seguintes resultados:

- Acurácia do modelo: Figura 56
- Perda do modelo: Figura 57
- Área sob a Curva ROC em função do número de épocas: Figura 58
- Matriz de confusão do modelo: Figura 59
- Relatório de classificação do modelo: Tabela 5

Figura 56 – Acurácia para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

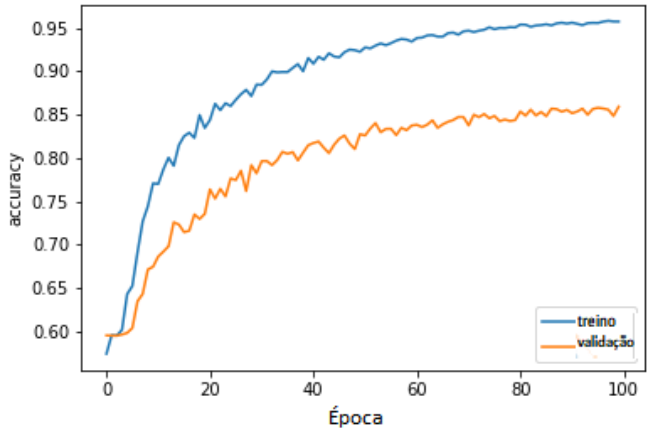


Figura 57 – Perda para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

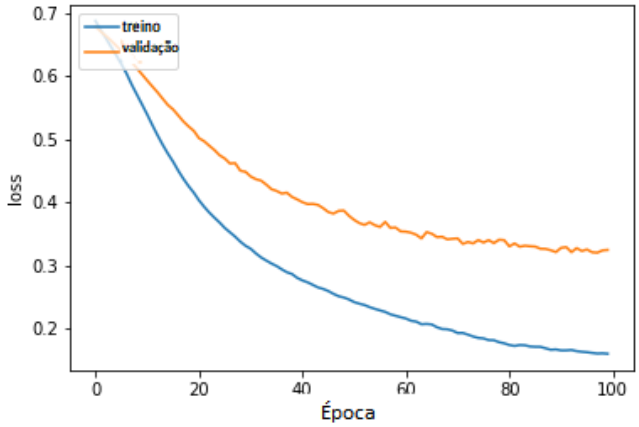


Figura 58 – Área sob a Curva ROC em função do número de épocas para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

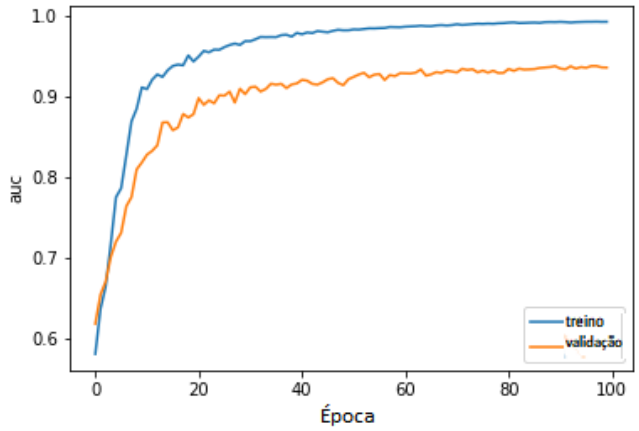


Figura 59 – Matriz de confusão para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

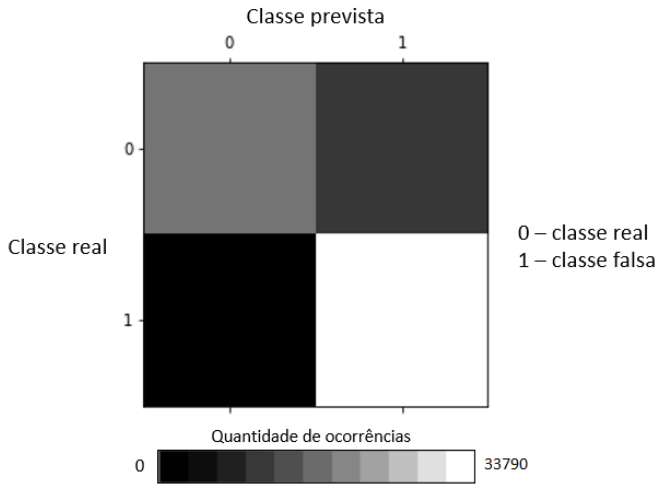


Tabela 5 – Relatório de classificação para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando filtro Gaussiano

	precisão	recall	f1-score
real	0,97	0,67	0,79
falsa	0,81	0,99	0,89
acurácia			0,86

5.1.6 *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros

Como desfecho do teste realizado considerando a técnica de *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros, encontram-se os seguintes resultados:

- Acurácia do modelo: Figura 60
- Perda do modelo: Figura 61
- Área sob a Curva ROC em função do número de épocas: Figura 62
- Matriz de confusão do modelo: Figura 63
- Relatório de classificação do modelo: Tabela 6

Figura 60 – Acurácia para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros

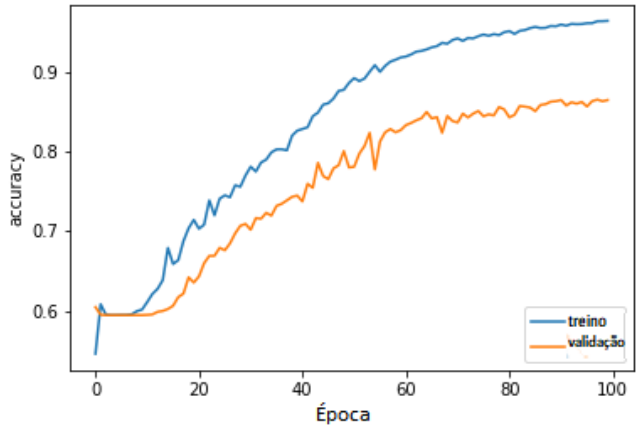


Figura 61 – Perda para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros

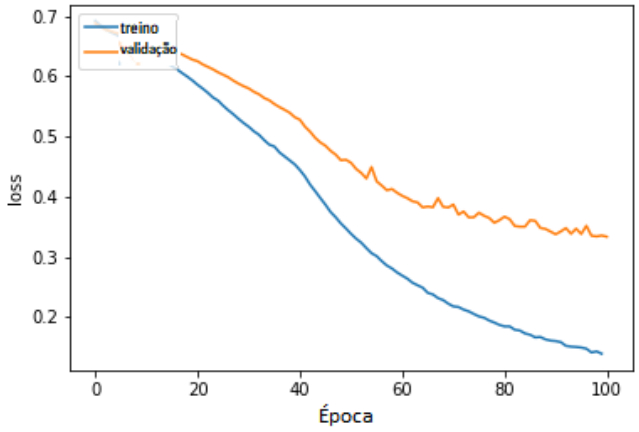


Figura 62 – Área sob a Curva ROC em função do número de épocas para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros

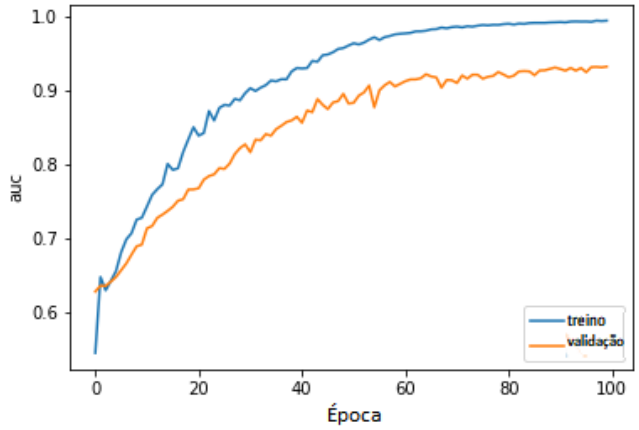


Figura 63 – Matriz de confusão para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros

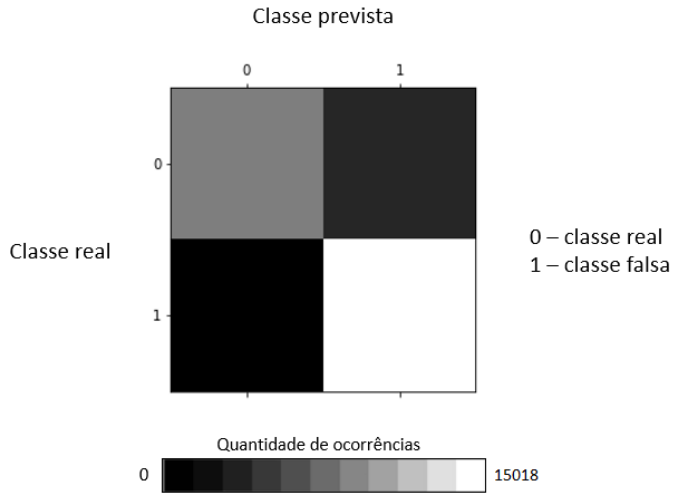


Tabela 6 – Relatório de classificação para o teste utilizando *Fine-tuning* para uma classe real e uma classe falsa com *Data Augmentation* utilizando ambos os filtros

	precisão	recall	f1-score
real	0,92	0,73	0,81
falsa	0,84	0,95	0,89
acurácia			0,86

5.2 Discussões

Considerando os primeiros testes realizados para quinze classes reais e, posteriormente, com a adição de uma classe de imagens falsas, pode-se observar, por meio do resultado consolidado presente na Tabela 7, que houve uma queda no desempenho do modelo em geral para todas as métricas avaliadas. Além disso, por meio das matrizes de confusão apresentadas anteriormente, evidencia-se que, para algumas classes específicas, o modelo apresentou uma queda significante após o incremento de imagens falsas.

Tabela 7 – Resultados consolidados do reconhecimento facial considerando a identidade da pessoa - comparação da eficiência do reconhecimento de imagens reais antes e depois da adição de imagens falsas

Real	15 classes reais	15 classes reais e 1 falsa
Acurácia	**0,850**	0,840
Média precisão	**0,863**	0,838
Média recall	**0,904**	0,896
Média f1-score	**0,863**	0,844

Já para os testes realizados apenas com classes reais e falsas, aplicando a técnica de *Data Augmentation*, observa-se, por meio das Tabelas 8 e Tabela 9, que, tanto para imagens reais, quanto para imagens falsas, a adição do descritor de textura LBP favoreceu a classificação das imagens. Para este caso, houve um incremento de todas as métricas para a classificação de imagens reais, e, para imagens falsas, observou-se apenas uma queda de 0.01 na taxa de *Recall* em relação ao teste realizado sem filtro.

Quando observa-se os outros dois testes realizados com filtros - Gaussiano e LBP+Gaussiano, tem-se que o modelo não apresentou alteração na acurácia, em relação ao teste efetuado sem a adição de filtros, e uma piora na taxa de precisão - no caso do Gaussiano -, e uma piora na taxa de *Recall* - no caso de ambos os filtros.

Tabela 8 – Resultados consolidados do reconhecimento facial considerando apenas classes reais - comparação da eficiência da técnica de *Data Augmentation*

Real	Sem Filtro	LBP	Gaussiano	LBP + Gaussiano
Acurácia	0,86	**0,87**	0,86	0,86
Precisão	0,94	0,92	**0,97**	0,92
Recall	0,70	**0,74**	0,67	0,73
f1-score	0,80	**0,82**	0,79	0,81

Tabela 9 – Resultados consolidados do reconhecimento facial considerando apenas classes falsas - comparação da eficiência da técnica de *Data Augmentation*

falsa	Sem Filtro	LBP	Gaussiano	LBP + Gaussiano
Acurácia	0,86	**0,87**	0,86	0,86
Precisão	0,83	**0,84**	0,81	0,84
Recall	0,97	0,96	**0,99**	0,95
f1-score	0,89	**0,90**	0,89	0,89

Dessa maneira, comparando todos os experimentos realizados, pode-se observar que a

existência de imagens falsas na predição do modelo, feito que caracterizaria uma tentativa de fraude, prejudica o desempenho do modelo quando este busca classificar a identidade de uma pessoa. No entanto, quando utiliza-se uma classificação somente entre imagens reais e falsas, o modelo avaliado (VGGFace) demonstrou bons resultados ao abstrair características das imagens e classificá-las de maneira correta - vide as matrizes de confusão apresentadas nos resultados -, possuindo um comportamento ainda melhor quando utilizada a técnica de *Data Augmentation* com o descritor de textura LBP.

Sendo assim, observa-se que a abordagem da utilização de duas redes neurais convolucionais distintas - uma para classificação de identidade e outra para detecção de fraude - torna-se válida pelo fato da rede neural de classificação de identidade possuir uma piora no desempenho quando exposta a imagens falsas e uma melhora ao ser treinada para diferenciar faces reais de faces falsas.

6 CONCLUSÃO

Esse trabalho teve como objetivo realizar um estudo sobre como o sistema biométrico de reconhecimento facial comporta-se quando exposto a imagens falsas e propor novas abordagens para detecção de fraude, desde combinar ao sistema uma fase de pré-processamento das imagens da face por meio de descritores de textura e filtros passa-alta, até a implementação de duas redes neurais para detecção de fraude e identificação de identidade.

Dessa maneira, os resultados obtidos nos experimentos mostraram que o sistema convencional utilizado para classificação de identidade possui uma queda de desempenho quando exposto a imagens falsas. Além disso, inferiu-se também, que, quando este sistema biométrico é atrelado a um procedimento de pré-processamento das faces e fragmentado em duas redes neurais, uma para detecção de fraude e outra para classificação da identidade, o desempenho obtido é superior ao convencional.

Vale ressaltar que todos os testes realizados foram obtidos exclusivamente para a base de dados testada (NUAA) e não consideraram em nenhum momento a robustez do *hardware* necessário para a predição das classes e o tempo de predição - como observado em outros trabalhos citados nesta monografia. Desta maneira, para a implementação do sistema *anti-spoofing* facial em dispositivos móveis ou terminais bancários, por exemplo, torna-se necessário o estudo para balancear todos estes fatores, que são custo, velocidade de processamento e taxa de assertividade do modelo. Além disso deve-se considerar também o aprimoramento do modelo para diferentes fisionomias humanas, considerando raça, etnia e sexo, além da possibilidade de existirem alterações na aparência humana por meio de procedimentos estéticos.

REFERÊNCIAS BIBLIOGRÁFICAS

- A. C. Weaver, "Biometric authentication,"in Computer, vol. 39, no. 2, pp. 96-97, Feb. 2006, doi: 10.1109/MC.2006.47.

- ACHARYA, T., RAY, A. K. Image Processing- Principles and Applications. John Wiley and Sons, Inc. 2005.

- BABICH, Aleksandra. Biometric Authentication. Types of biometric identifiers. 2012. 53 f. TCC (Graduação) - Curso de Information Technology, Haaga-Helia University Of Applied Sciences, Helsinki, 2012.

- BOTTOU, Léon. Large-Scale Machine Learning with Stochastic Gradient Descent. Proceedings Of Compstat'2010, v. 1, n. 1, p. 177-186, set. 2010. Physica-Verlag HD.

- BRAIN, Google et al. TensorFlow: A System for Large-Scale Machine Learning. In: USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION, 12., 2016, Savannah, Ga, Usa. TensorFlow: A system for large-scale machine learning. Savannah, Ga, Usa: Usenix Association, 2016. p. 1-21.

- BROWNLEE, Jason. How to Configure Image Data Augmentation in Keras. Disponível em: https://machinelearningmastery.com/how-to-configure-image-data-augmentation-when-training-deep-learning-neural-networks/. Acesso em: 21 ago. 2020.

- BUSSON, Antonio José G. et al. Desenvolvendo Modelos de Deep Learning para Aplicações Multimídia no Tensorflow. Rio de Janeiro: Puc-rio, 2018.

- G. Pan, L. Sun, Z. Wu and S. Lao, "Eyeblink-based Anti-Spoofing in Face Recognition from a Generic Webcamera,"2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, 2007, pp. 1-8, doi: 10.1109/ICCV.2007.4409068.

- GONZALEZ, Rafael C. et al. PROCESSAMENTO DIGITAL DE IMAGENS. São Paulo: Pearson, 2009.

- GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep Learning. Londres: Mit Press, 2016.

- GOOGLE (org.). Colaboratory. Disponível em: https://research.google.com/colaboratory/faq.html. Acesso em: 12 set. 2020.

- GOSE, E.; JOST, S.; JOHNSONBAUGH, R. Pattern Recognition and Image Analysis, Prentice Hall, 1996.

- GURNEY, Kevin. An introduction to neural networks. Londres: Ucl Press, 1997.

- JAWOREK-KORJAKOWSKA, Joanna; KLECZEK, Pawel; GORGON, Marek. Melanoma Thickness Prediction Based on Convolutional Neural Network With VGG-19 Model Transfer Learning. 2019 Ieee/cvf Conference On Computer Vision And Pattern Recognition Workshops (Cvprw), Long Beach, Ca, Usa, v. 1, n. 1, p. 2748-2756, jun. 2019. IEEE. http://dx.doi.org/10.1109/cvprw.2019.00333.

- JOST, I.. Aplicação de Deep Learning em dados refinados para Mineração de Opiniões. Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2015.

- GÉRON, Aurélian. Mãos à obra: Aprendizado de Máquina com Scikit-Learn TensorFlow. Rio de Janeiro: O'Reilly, 2019.

- HADID, Abdenour. The Local Binary Pattern Approach and its Applications to Face Analysis. 2008 First Workshops On Image Processing Theory, Tools And Applications, Sousse, v. 1, n. 1, p. 1-9, nov.

- HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, v. 3, n. 6, p. 610–621, Nov 1973. ISSN 0018-9472.2008. IEEE. http://dx.doi.org/10.1109/ipta.2008.4743795.

- HAYKIN, S. S.. Redes Neurais: princípios e prática. 2ed. Ed. Bookman Companhia Ed, 2001.

- HUANG, Di; SHAN, Caifeng; ARDABILIAN, Mohsen; WANG, Yunhong; CHEN, Liming. Local Binary Patterns and Its Application to Facial Image Analysis: a survey. Ieee Transactions On Systems, Man, And Cybernetics, Part C (Applications And Reviews), [S.L.], v. 41, n. 6, p. 765-781, nov. 2011. Institute of Electrical and Electronics Engineers (IEEE). http://dx.doi.org/10.1109/tsmcc.2011.2118750.

- J. Galbally, S. Marcel and J. Fierrez, "Biometric Antispoofing Methods: A Survey in Face Recognition,"in IEEE Access, vol. 2, pp. 1530-1552, 2014, doi: 10.1109/ACCESS.2014.2381273.

- I. Chingovska, A. Anjos and S. Marcel, "On the effectiveness of local binary patterns in face anti-spoofing,"2012 BIOSIG - Proceedings of the International Conference of Biometrics Special Interest Group (BIOSIG), Darmstadt, 2012, pp. 1-7.

- KERAS API REFERENCE. Probabilistic losses. Disponível em: https://keras.io/api/losses/probabilistic-losses/sparsecategoricalcrossentropy-class. Acesso em: 10 ago. 2020.

- L. Li, Z. Xia, L. Li, X. Jiang, X. Feng and F. Roli, "Face anti-spoofing via hybrid convolutional neural network,"2017 International Conference on the Frontiers and Advances in Data Science (FADS), Xi'an, 2017, pp. 120-124, doi: 10.1109/FADS.2017.8253209.

- Manminder Singh, A.S. Arora, A robust anti-spoofing technique for face liveness detection with morphological operations, Optik, Volume 139, 2017, Pages 347-354, ISSN 0030-4026, https://doi.org/10.1016/j.ijleo.2017.04.004. item MANNOR, Shie; PELEG, Dori; RUBINSTEIN, Reuven. The cross entropy method for classification. Proceedings Of The 22Nd International Conference On Machine Learning - Icml '05, Montreal, v. 1, n. 1, p. 561-568, ago. 2005. ACM Press.

- MARQUES, Victor Gutemberg Oliveira. Avaliação do desempenho das redes neurais convolucionais na detecção de ovos de esquistossomose. 2017. 49 f. TCC (Graduação) - Curso de Engenharia da Computação, Universidade Federal de Pernambuco, Recife, 2017.

- MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. Processamento Digital de Imagens, Rio de Janeiro: Brasport, 1999.

- NAR, K.; OCAL, O.; SASTRY, S.; RAMCHANDRAN, K.. Cross-Entropy Loss and Low-Rank Features Have Responsibility for Adversarial Examples. Arxiv. Nova York, p. 1-10. jan. 2019.

- NIELSEN, Michael. Neural Networks and Deep Learning. São Francisco: Determination Press, 2015.

- OLIVEIRA, Weiner Esmerio Batista de; PRADO, Alisson Fernandes do; FERNANDES, Sandro Roberto; FACEROLI, Silvana Terezinha. CLASSIFICAÇÃO DE PADRÕES UTI-

LIZANDO DESCRITORES DE TEXTURA. Juiz de Fora: Instituto Federal de Educação e Tecnologia do Sudeste de Minas Gerais – Campus Juiz de Fora, 2014.

- PACHECO, César Augusto Rodrigues; PEREIRA, Natasha Sophie. Deep Learning Conceitos e Utilização nas Diversas Áreas do Conhecimento. Anápolis - Go: UnievangÉlica, 2018.

- PARKHI, Omkar M.; VEDALDI, Andrea; ZISSERMAN, Andrew. Deep Face Recognition. Procedings Of The British Machine Vision Conference 2015, Oxford, v. 1, n. 1, p. 41-53, set. 2015. British Machine Vision Association. http://dx.doi.org/10.5244/c.29.41.

- PAVLOVSKY, Vojtech. Introduction To Convolutional Neural Networks. 2017. Disponível em: https://www.vojtech.net/posts/intro-convolutional-neural-networks/. Acesso em: 10 ago. 2020

- QUEIROZ, José Eustáquio Rangel de; GOMES, Herman Martins. Introdução ao Processamento Digital de Imagens. Campina Grande: Ufcg, 2006.

- REBELLO, Gabriel; HU, Yining; THILAKARATHNA, Kanchana; BATISTA, Gustavo; SENEVIRATNE, Aruna; DUARTE, Otto Carlos Muniz Bandeira. Melhorando a Acurácia da Detecção de Lavagem de Dinheiro na Rede Bitcoin. In: ANAIS PRINCIPAIS DO SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS (SBRC), 38. , 2020, Rio de Janeiro. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020 . p. 728-741. ISSN 2177-9384. DOI: https://doi.org/10.5753/sbrc.2020.12321.

- SCHUCKERS, Stephanie A.C. Spoofing and Anti-Spoofing Measures. Information Security Technical Report, [S.L.], v. 7, n. 4, p. 56-62, dez. 2002. Elsevier BV. http://dx.doi.org/10.1016/s1363-4127(02)00407-7.

- SERENGIL, Sefik Ilkin. Deep Face Recognition with Keras. 2018. Disponível em: https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/. Acesso em: 06 ago. 2020.

- SOARES FILHO, MaurÍcio Marques. REDES NEURAIS ARTIFICIAIS: DO NEURÔNIO ARTIFICIAL À CONVOLUÇÃO. 2018. 84 f. TCC (Graduação) - Curso de Tecnologia em Sistemas de Computação, Universidade Federal Fluminense, Niterói, 2018.

- SOUZA, Gustavo Botelho de. Detecção de ataques a sistemas de reconhecimento facial utilizando abordagens eficientes de aprendizado de máquina em profundidade. 2019. 260 f.

Tese (Doutorado) - Curso de Ciência da Computação, Universidade Federal de São Carlos, São Carlos, 2019.

- TIRUNAGARI, S. et al. Detection of face spoofing using visual dynamics. IEEE Transactions on Information Forensics and Security, v. 10, n. 4, p. 762–777, 2015.
- VERSLOOT, Christian. How to use sparse categorical crossentropy in Keras. Disponível em: https://www.machinecurve.com/index.php/2019/10/06/how-to-use-sparse-categorical-crossentropy-in-keras/. Acesso em: 10 ago. 2020.