UNIVERSIDADE FEDERAL DE SANTA CATARINA

Pedro Henrique Azevedo

DETECÇÃO DE OBJETOS EM IMAGENS PARA VERIFICAÇÃO DE VIVACIDADE

Florianópolis

2024.1

UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CURSO DE SISTEMAS DE INFORMAÇÃO

DETECÇÃO DE OBJETOS EM IMAGENS PARA VERIFICAÇÃO DE VIVACIDADE

Pedro Henrique Azevedo

Trabalho de Conclusão de Curso submetido ao Programa de graduação da Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Sistemas de Informação

Florianópolis - SC

2024 / 1

PEDRO HENRIQUE AZEVEDO

DETECÇÃO DE OBJETOS EM IMAGENS PARA VERIFICAÇÃO DE VIVACIDADE

Trabalho de conclusão de curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

_____

Prof. Renato Cislaghi, Dr.

Coordenador

_____

Prof. Elder Rizzon Santos, Dr.

Orientador

Banca Examinadora:

_____

Prof. Alexandre Gonçalves Silva, Dr.

_____

Thiago Ângelo Gelaim, Dr.

RESUMO

A biometria facial tem ganhado cada vez mais espaço na validação de identidades. Na mesma medida em que isso ocorre, torna-se necessário diminuir a vulnerabilidade desses sistemas. Um indivíduo pode utilizar uma foto, vídeo ou outro tipo de artefato para tentar se passar por outra pessoa, fraudando sua imagem e burlando o sistema de biometria. Este trabalho propõe um modelo de aprendizado de máquina, baseado na rede YOLO, para identificar objetos ou artefatos em imagens que possam indicar uma fraude biométrica. Além disso, foram realizados testes e validações dos resultados. O modelo proposto alcançou resultados de 24,55% no conjunto de dados MSU-MFSD e de 4,55% no conjunto de dados RECOD-MPAD utilizando métrica HTER. Em uma análise mais detalhada das métricas de precisão e recall para as classes propostas, o modelo apresentou bons resultados na identificação de artefatos como mãos e bordas sobressalentes, que indicam a tentativa de passar algum objeto pela biometria ao invés de uma face real. Apesar dos desafios na identificação de algumas classes e dos resultados discrepantes de HTER, a proposta de detecção de objetos para identificar vivacidade mostrou potencial na identificação de alguns tipos de artefatos que denunciam uma fraude biométrica, podendo ser utilizada em pesquisas futuras.

Palavras-chave: inteligência artificial, redes neurais, detecção de objetos, vivacidade

ABSTRACT

Facial biometrics have been increasingly used for identity validation. As this technology becomes more widespread, it is necessary to reduce the vulnerability of these systems. An individual can use a photo, video, or other types of artifacts to impersonate someone else, thereby defrauding their image and bypassing the biometric system. This work proposes a machine learning model, based on the YOLOv8 network, to identify objects or artifacts in images that may indicate biometric fraud. Additionally, tests and validations of the results were conducted. The proposed model achieved results of 24.55% on the MSU-MFSD dataset and 4.55% on the RECOD-MPAD dataset using the HTER metric. In a more detailed analysis of precision and recall metrics for the proposed classes, the model showed good results in identifying artifacts such as hands and protruding edges, which indicate an attempt to pass an object through the biometric system instead of a real face. Despite the challenges in identifying some classes and the discrepant HTER results, the proposed object detection approach for liveliness identification showed potential in detecting certain types of artifacts that signal biometric fraud, and it can be used in future research.

Keywords: artificial intelligence, neural networks, object detection, liveness

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

DNA - Deoxyribonucleic Acid

LCD - Liquid Crystal Display

HTER - Half Total Error Rate

BPCER - Bonafide Presentation Classification Error Rate

APCER - Attack Presentation Classification Error Rate

CNN - Convolutional Neural Network

ReLU - Rectified Linear Unit

R-CNN - Regions With Convolutional Neural Networks

RECOD-MPAD - Research and Coordination Mobile Presentartion-Attack Dataset

CASIA-FASD - Chinese Academy of Sciences Institute of Automation Face Anti-Spoofing Database

NUAA - Nanjing University of Aeronautics and Astronautics

YOLO - You Only Look Once

VOC - Visual Object Classes

mAP - mean Average Precision

PAD - Presentation Attack Detection

SBIQF - Spatial Binary Image Quality Feature

MSU-MFSD - Michigan State University Mobile Face Spoofing Database

1. INTRODUÇÃO

Ferramentas de reconhecimento facial estão se tornando cada vez mais relevantes e necessárias, a fim de promover segurança e eficiência na validação de identidades. A tecnologia de biometria facial se desenvolveu muito nos últimos anos, com uma melhoria na precisão e confiabilidade dos algoritmos, permitindo o uso da tecnologia em aplicações diversas como: análise forense, controles de acesso, comércio eletrônico, entre outros.

A aplicação generalizada de sistemas de reconhecimento facial também traz novos problemas, principalmente no que diz respeito à vulnerabilidade. Existem diversas motivações para que um indivíduo tente realizar um ataque, além disso, os artefatos faciais necessários para executar a fraude podem ser criados de forma simples e rentável. (RAMACHANDRA; BUSCH, 2017).

Uma infinidade de conteúdos em páginas na web ensinam como criar esses artefatos faciais, e atualmente é muito fácil obter uma imagem da face da vítima em suas redes sociais, ou capturar uma imagem da face dela sem que ela perceba, com um aparelho celular por exemplo.

A verificação de vivacidade impede que vídeos *deep fakes*, fotos ou outras falsificações possam ser utilizadas para fraudar informações, garantindo com boa acurácia a existência de um ser humano real dentro dessas mídias. Dorothy E. Denning (2021), membro da National Cyber Security Hall of Fame afirmou que "...impressões biométricas não precisam ser mantidas em segredo, mas o processo de validação deve verificar a vivacidade das leituras.”

Para realizar essa verificação podem ser observadas características como padrões em microtexturas de amostras de imagens de face, ou a exploração da informação temporal em vídeos para determinar movimentos relativos da cabeça, músculos faciais e etc. (RAMACHANDRA; BUSCH, 2017).

Em alguns casos, porém, podem haver elementos físicos que denunciam um possível ataque, ou até artefatos na imagem que permitam uma abordagem como detecção, que pode ser mais direta e eficaz nesses casos do que tentar identificar elementos mais sutis de vivacidade. Segundo Zou, Z et al. (2019), a tarefa de detecção diz respeito a identificar e localizar instâncias de objetos de interesse dentro de uma imagem ou vídeo, a detecção não apenas identifica, como determina a posição de algum objeto por meio de caixas delimitadoras.

Analisando essas características com algoritmos de inteligência artificial é possível indicar automaticamente se o indivíduo está presente na biometria, ou se existe uma tentativa de fraude. O conceito de inteligência artificial é amplo e não possui uma definição exata, uma vez que a própria inteligência em si não possui um conceito totalmente definido, o termo pode ser conceituado de diversas formas. Segundo Luger (2008), a inteligência artificial pode ser definida como o ramo da ciência da computação preocupada com a automação do comportamento inteligente.

Dentro desse contexto, para que seja possível identificar a vivacidade em imagens a partir de uma solução computacional eficiente, este trabalho propõe uma solução utilizando redes neurais. Redes neurais são uma área da inteligência artificial, onde o estudo é focado em desenvolver técnicas para solução de problemas com base na estrutura dos neurônios humanos. Podemos dizer que suas principais

características são: a forma de processamento distribuído, a habilidade de aprender e a capacidade de lidar com dados imprecisos (RUSSEL; NORVIG, 2004).

Entre as características presentes nas redes neurais, um conceito importante é o de aprendizagem. A capacidade de aprender deve fazer parte de qualquer sistema que pretenda possuir inteligência. Agentes inteligentes devem ser capazes de mudar ao longo de suas interações com o mundo, bem como através da experiência de seus próprios estados e processos internos (LUGER, 2008). Técnicas de aprendizado de máquina vêm sendo amplamente adotadas em diversos contextos. Sendo assim existem vários tipos de arquiteturas para redes neurais, como a rede neural recorrente e a rede neural convolucional, dentre outras.

No presente trabalho é proposto um modelo de rede neural com aprendizagem supervisionada. Segundo Russel e Norvig (2010), aprendizagem supervisionada consiste na tarefa de um algoritmo aprender uma função que mapeia uma entrada para uma determinada saída com base em exemplos de pares de entrada e saída. Utilizando essa abordagem, e por meio de uma base de dados consolidada realizar o treinamento e validação de um modelo de aprendizado de máquina, bem como experimentos e avaliações dos resultados obtidos. propondo assim uma solução para o problema da detecção de vivacidade em imagens, utilizando a abordagem de detecção de objetos.

1.1. Objetivos

1.1.1. Objetivo geral

O presente trabalho de conclusão de curso tem como finalidade desenvolver um modelo de aprendizado de máquina através de redes neurais, a fim de detectar objetos em imagens que sinalizem uma tentativa de ataque por fotografia ou vídeo, indicando uma fraude.

1.1.2. Objetivos específicos

- Realizar pesquisa teórica sobre o estado da arte em *machine learning* com abordagem supervisionada, focado em detecção de objetos em imagens, buscando obter uma visão geral do que está sendo utilizado atualmente;
- Analisar algoritmos e metodologias de redes neurais, capazes de solucionar o problema proposto;
- Propor um modelo de rede neural para resolução do problema apresentado, a partir da análise feita no objetivo anterior;
- Realizar experimentos com a abordagem proposta;
- Avaliar os resultados obtidos.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados os conceitos de Biometria, Testes de Vivacidade, Redes Neurais, Redes Neurais Convolucionais e Detecção de Objetos, importantes no desenvolvimento deste trabalho.

2.1. Biometria

O reconhecimento biométrico é baseado na capacidade de identificar exclusivamente uma pessoa, extraindo um ou mais traços biológicos distintivos. Essas características biológicas podem ser o rosto de uma pessoa, impressões digitais, padrões de retina e íris, geometria da mão, voz, DNA ou assinaturas manuscritas (OMAR et al., 2018). Em comparação com outros métodos de autenticação biométrica como impressões digitais ou imagens de íris, o reconhecimento facial possui o diferencial de ser baseado em dados que são facilmente encontrados em domínio público, tornando-o vulnerável a ataques de impostores. A biometria facial não exige necessariamente a interação direta do usuário, podendo ser feita de forma passiva, ou seja, os dados podem ser capturados de forma direta e não intrusiva como vigilância em câmeras de segurança, por exemplo.

2.1.1. Tipos de Ataque

Dentro do problema da detecção da vivacidade existem três principais tipos de fraudes. São elas:

- Ataques por fotos impressas: Utilização de uma imagem do usuário impressa em uma folha de papel. O tamanho e material do papel, bem como a qualidade da impressora, os perfis de cores e a iluminação ambiente, são variáveis relevantes nesse tipo de ataque. Outra característica importante a se levar em consideração é que o conteúdo real da imagem será sempre estático, embora a foto possa ser movimentada em frente à câmera.
- Ataques por *displays*: Utilização de uma tela como um monitor LCD, ou um tablet para exibir uma imagem estática ou um vídeo. A qualidade do ataque dependerá do tamanho do monitor, da densidade dos pixels, dos níveis de contraste, reprodução de cores e etc.
- Ataques por máscaras 3D: Utilização de uma máscara para representar o rosto do indivíduo em formato 3D. Neste caso o esforço do ataque é maior do que nos métodos anteriores, tornando-o menos comum.

2.2. Testes de Vivacidade

Testes de vivacidade utilizam algoritmos de classificação binária que fazem a distinção entre um indivíduo frente a uma câmera e algum tipo de fraude, que nos casos de detecção de face podem ser imagens, vídeos, ou até modelos em 3D (OMAR et al., 2018). Esses testes têm ganhado mais relevância com o crescimento de *deep fakes* e sistemas de biometria facial.

2.2.1. Métodos Baseados em *Hardware* x Métodos Baseados em *Software*

Segundo Ramachandra e Bush (2017), abordagens baseadas em *hardware* exploram as características do rosto humano usando componentes de *hardware* adicionais identificados que funcionam em associação com o reconhecimento facial. Tais abordagens também exigem interação com o *hardware* de captura, que extrai dados adicionais na imagem utilizando:

- Sensores de Profundidade: Capturam informações tridimensionais da superfície da face.
- Câmeras Multiespectrais: Identificam diferentes faixas do espectro eletromagnético (infravermelho, ultravioleta, luz visível). Diferentes características de reflexão e absorção de luz e diferentes comprimentos de onda podem ser usadas para distinguir a pele humana de outro material.
- Sensores Térmicos: Medem a radiação infravermelha emitida pela superfície do rosto que será diferente em relação a objetos inanimados ou fotos.

Em contraponto, os métodos baseados em *software* não precisam de *hardware* específico nem de interação com o usuário, analisando características das imagens ou vídeos capturados, o que os torna mais flexíveis, com mais facilidade de atualização e menor custo. Os métodos baseados em software podem ser divididos em estáticos e dinâmicos.

Métodos estáticos: Projetados para trabalhar em cima de uma imagem sem a necessidade de informações temporais, mas podem ser utilizados também para analisar vídeos quadro a quadro. Esses métodos são conhecidos por terem bom desempenho e baixo custo computacional. Eles podem ser divididos em três tipos:

- Baseados em Textura: Analisam características de da textura da imagem que diferenciam uma amostra real de uma falsificação. A pele humana possui padrões de texturas naturais únicos, difíceis de replicar em materiais falsos como poros e variações naturais. Os métodos baseados em texturas são eficientes na detecção de artefatos em fotos e *display* por diferenciar características como a presença de pigmentos de impressões, reflexão especular, e sombra.
- Baseados em Frequência: Analisam a distribuição de frequências espaciais da imagens, identificando padrões que diferenciam a superfície de um rosto real de uma tentativa de fraude. Diferentes superfícies e materiais refletem e absorvem luz de maneira distinta, gerando diferentes componentes de frequência.
- Híbridos: Combinam técnicas de análise de textura e frequência para fornecer uma detecção mais completa e precisa de vivacidade.

Métodos dinâmicos: Exploram informações temporais do movimento relativo através dos quadros de vídeo. Portanto, uma abordagem dinâmica exigirá mais tempo, assim como maior esforço computacional quando comparado com uma abordagem estática. Bem como a abordagem anterior, pode ser dividida em três tipos:

- Baseados em Movimento: Verificam se há movimentos naturais esperados como piscar, oscilações e variações de expressão facial.
- Baseados em Textura: Analisam as variações temporais das texturas da pele que podem mudar ligeiramente com movimentos, mudanças na iluminação e variações naturais. Características difíceis de replicar com apresentações falsas de fotos ou vídeos.
- Híbridos: Combinam a análise de movimento e textura dinâmica, conseguindo captar uma gama mais ampla de sinais temporais de vivacidade.

2.2.2. Métricas

No contexto de ataques de apresentação, são utilizadas diversas métricas específicas, que permitem uma análise detalhada do desempenho do sistema em diferentes cenários. Entre as principais métricas utilizadas estão: *Half Total Error Rate* (HTER), *Bonafide Presentation Classification Error Rate* (BPCER) e *Attack Presentation Classification Error Rate* (APCER). Essas métricas são fundamentais para a compreensão dos pontos fortes e limitações dos sistemas biométricos, bem como para orientar melhorias e ajustes necessários.

- APCER: Representa a taxa de erro de classificação de apresentações falsas. Medindo a porcentagem dos casos em que um sistema de detecção de apresentação erroneamente classifica uma apresentação falsa como genuína.

  - $APCER = \frac{Número\ de\ Apresentações\ de\ ataque\ Incorretamente\ Aceitas}{Número\ Total\ de\ Apresentações\ de\ Ataque}$

- BPCER: Representa a taxa de erro de classificação de apresentações genuínas, medindo a porcentagem dos casos em que um sistema de detecção de apresentação erroneamente classifica uma apresentação genuína como falsa.

  - $BPCER = \frac{Número\ de\ Apresentações\ Autênticas\ Incorretamente\ Rejeitadas}{Número\ Total\ de\ Apresentações\ de\ Autênticas}$

- HTER: Proporciona uma visão equilibrada do desempenho de sistemas de autenticação, identificando se o sistema é mais propenso a aceitar ataques ou rejeitar amostras genuínas. O HTER pode ser calculado através da média aritmética entre APCER e BPCER.

  - $HTER = \frac{APCER + BPCER}{2}$

2.3. Redes Neurais

Segundo a neurociência, a atividade mental consiste principalmente em atividade eletroquímica nas redes de células cerebrais chamadas neurônios. Inspirados por essa hipótese, trabalhos no campo da inteligência artificial visam criar redes neurais artificiais (RUSSELL; NORVIG, 2016). O objetivo é simular através de camadas de nós, também chamados de neurônios, as conexões feitas no processo de aprendizagem orgânica como ilustrado na figura 1. A fim de desenvolver sistemas inteligentes capazes de classificar e identificar padrões em diferentes conjuntos de dados.

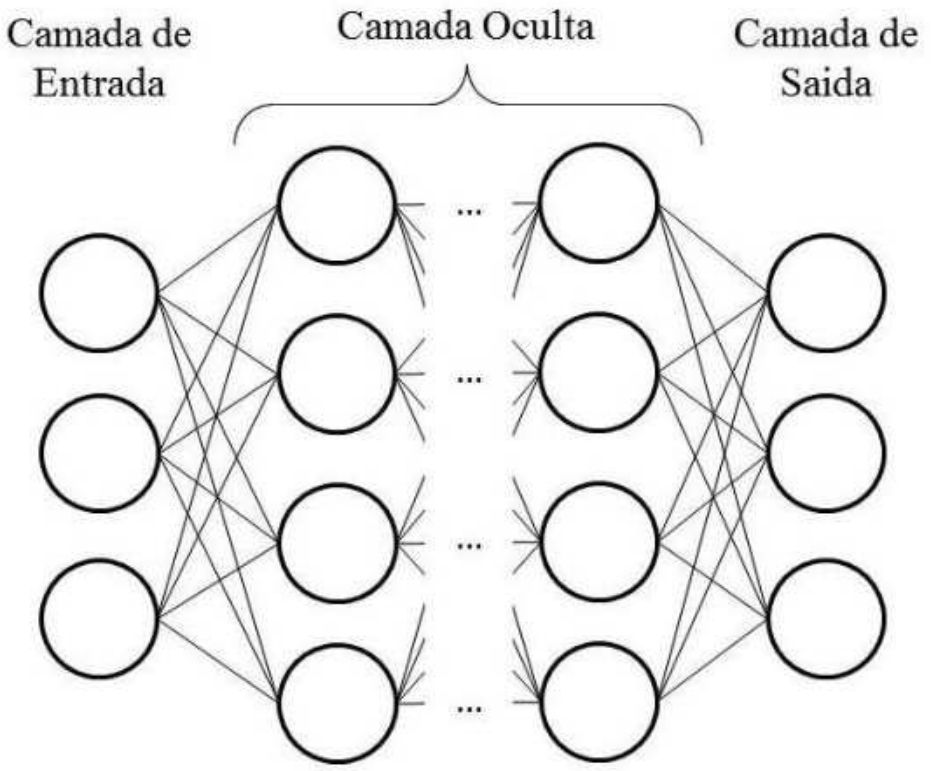


Figura 1 - Ilustração de uma rede neural artificial

Fonte: Gabriel Pereira, 2017

2.3.1. Neurônio Artificial

Um neurônio artificial é uma unidade de processamento, ou seja, uma unidade básica dentro da rede que emula o funcionamento de um neurônio biológico através de uma abstração matemática, que segundo o modelo de Mcculloch e Pitts (1943) é composto pelos elementos da figura 2:

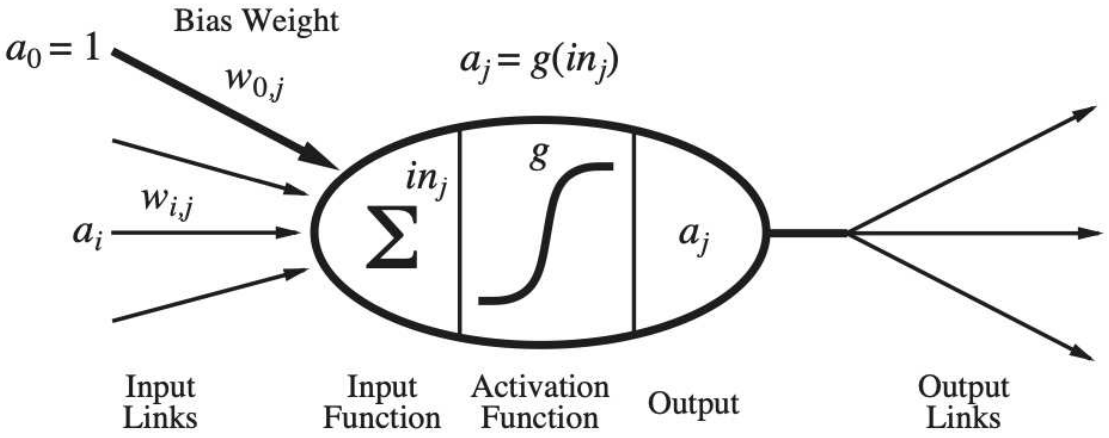


Figura 2 - Ilustração de um neurônio artificial

Fonte: Russell e Norvig, 2016

- Conexões de entrada *(Input Links*): Representa a influência que determinada entrada tem com o neurônio através do seu peso associado, ou seja, cada conexão de entrada é responsável por transmitir o valor ponderado de determinada entrada para o neurônio.
- Peso do Viés (*Bias Weight*): Parâmetro adicional que permite ajustar o ponto de partida da ativação de determinado neurônio, independente se suas entradas.
- Função de entrada (*Input Function*): Calcula a soma ponderada das entradas multiplicada pelos pesos sinápticos correspondentes.
- Função de Ativação (*Activation Function*): Função matemática que transforma a soma ponderada das entradas em uma saída, definindo o comportamento do neurônio ao determinar se ele será ativado ou não com base em suas entradas.
- Saída (*Output*): Valor produzido pelo neurônio após processar as estradas e aplicar a função de ativação.

- Conexões de saída (*Output Links*): Permitem que a saída do neurônio seja transmitida para os outros neurônios na rede. Cada conexão de saída possui um peso sináptico associado, que representa a importância da conexão entre o neurônio atual e o neurônio destino, influenciando a contribuição da saída do neurônio atual na ativação dos neurônios a ele conectados.

2.4. Redes Neurais Convolucionais

As redes neurais convolucionais (CNN) são uma arquitetura especializada em processar dados em formato de grade, como imagens (LECUN et al., 2015). Elas utilizam operações convolucionais para extrair características relevantes dos dados, como bordas e texturas (KRIZHEVSKY et al., 2012). Essas características são aprendidas durante o treinamento do modelo, permitindo que as CNNs identifiquem padrões complexos nas imagens (SIMONYAN; ZISSERMAN, 2014).

Segundo Satapathy et al. (2020) uma rede neural convolucional é uma rede neural de aprendizado profundo usada para o processamento de imagens e aplicativos de visão computacional. Ela utiliza uma arquitetura *multilayer perceptron* e funciona com base no princípio da correlação local. Cada camada consiste em uma pilha de neurônios que reconhecem padrões cada vez mais complexos. Uma rede neural convolucional é composta de camadas convolucionais e de agrupamento que estão bem conectadas com uma rede neural totalmente conectada de várias camadas.

2.4.1. Camada de Entrada

Coloca cada um dos valores de entrada (*pixels*) em um dos neurônios.

2.4.2. Camadas Convolucionais

Camada onde ocorre a operação de convolução, que consiste na multiplicação entre uma matriz de entrada que representa os *pixels* da imagem e um *kernel*, que age como um filtro destinado a extrair as características mais relevantes da imagem, tais como bordas, texturas e outros padrões. Esta operação tem como resultado uma matriz conhecida como *feature map*.

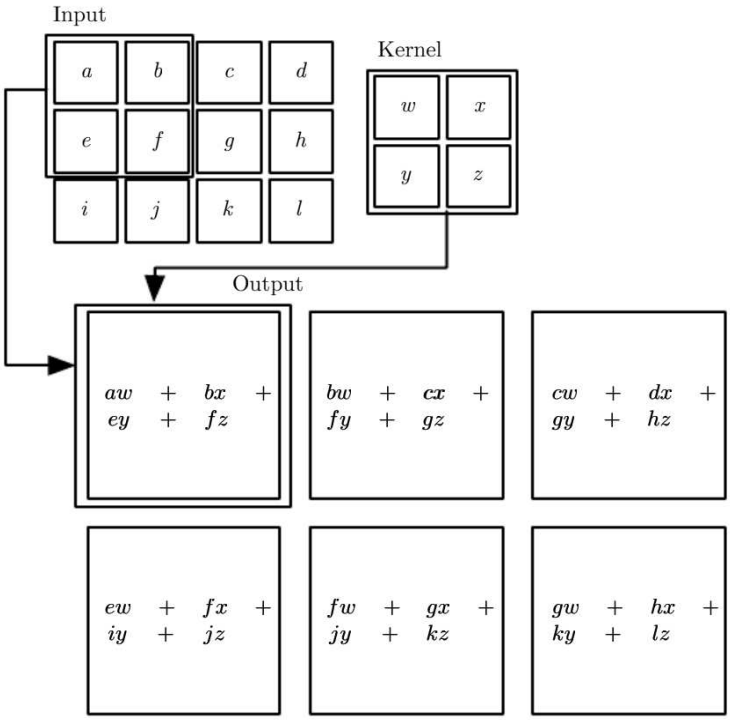


Figura 3 - Exemplo da operação de convolução.

Fonte: Goodfellow, I et al.2016

A figura 3 ilustra o processo de convolução em CNNs, onde um filtro (*kernel*) é aplicado a uma matriz de entrada *(input*) para produzir uma matriz de saída(*output*). Neste exemplo a matriz de entrada (4x3) é processada por um *kernel* (2x2), que é posicionado inicialmente na parte superior esquerda da matriz. Cada elemento do *kernel* é multiplicado pelo elemento correspondente na entrada e os resultados são somados para formar um valor da matriz de saída. Este processo, denominado convolução, envolve mover o *kernel* pela matriz de entrada, realizando multiplicações e somas para cada posição, produzindo valores que formam a matriz de saída. Isso é repetido até que todo o *kernel* tenha passado por toda a matriz de entrada.

2.4.3. Camadas de Ativação

Aplica a função de ativação nos valores presentes no feature map. No caso das redes neurais convolucionais a função ReLU (Rectified Linear Unit) é a mais utilizada devido ao seu menor custo computacional e resultados próximos aos de outras funções. Elas ativam o neurônio caso o dado de entrada seja positivo e retorna zero para valores negativos.

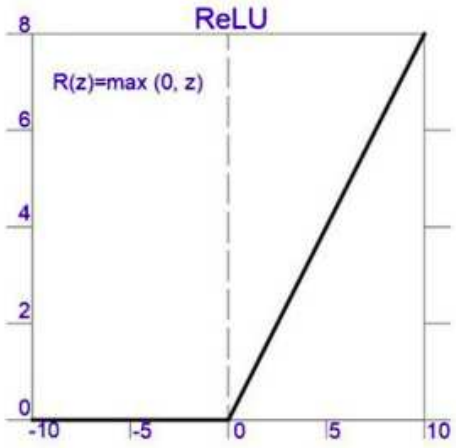


Figura 4 - função de ativação (ReLu).

Fonte: ISSA, Anas (2023)

A figura 4 exemplifica a função ReLU que é definida pela equação R(z)=max(0,z), onde z é o valor de entrada. No gráfico mostra a saída como zero para entradas não positivas e uma saída linear para entradas positivas, com os valores de saída iguais aos valores de entrada (z).

2.4.4. Camadas de Pooling

Utilizada para reduzir a altura e a largura da entrada extraída pelas camadas de convolução. Reduzindo a resolução com operação de *pooling*, obtém-se valores mais expressivos de uma região da matriz, além de de um aumento de performance com menos informações a serem computadas. Entre os diferentes tipos de operações de pooling podem ser citados:

- *Max Pooling*: Nessa operação a matriz de entrada é dividida em pequenas regiões, e dentro de cada uma delas o valor máximo é selecionado para a

composição da matriz de saída. O *max pooling* é útil para representar características como bordas ou texturas, pois enfatiza os valores máximos, que geralmente correspondem a elementos mais sobressalentes da imagem. Por exemplo, considerando uma região de tamanho 2x2 da matriz de entrada:

1	3
2	4

Nesta região, o valor máximo é 4. Portanto o resultado do *max pooling* para essa região assume o valor 4. Esse procedimento é refeito em todas as regiões da matriz de entrada, resultando na matriz de saída reduzida.

- *Average Pooling*: O *average pooling* realiza um processo semelhante, porém em vez de selecionar o valor máximo, é calculada a média dos valores em cada região da matriz de entrada. Considerando a mesma matriz do exemplo anterior:

1	3
2	4

A média dos valores é calculada como:

$\frac{1+3+2+4}{4} = 2,5$

Nesse caso o resultado do average pooling para a região do exemplo é de 2,5. Esta abordagem gera uma representação mais generalizada da área da entrada pois considera todos os valores da região.

2.4.5. Camadas totalmente Conectadas

Camada que conecta todos os neurônios da camada anterior, ou seja, se a camada anterior tem ˜n˜ saídas esta camada terá ˜n˜ neurônios. Assim, as informações aprendidas em outras camadas são utilizadas na identificação de padrões.

2.4.6. Camada de Saída

Produz previsões finais com base no que foi extraído das camadas convolucionais anteriores. Utiliza funções de ativação dependendo da tarefa a ser resolvida. Para classificação multiclasse é comum utilizar-se *softmax*, já para classificação binária é comum utilizar-se *sigmoid*.

- *Softmax*: Converte as saídas lineares da camada anterior nas probabilidades de uma amostra ser de alguma das classes, garantindo que a soma para cada uma delas seja igual a 1.

$$softmax(z_1) = \frac{e^{z_i}}{\Sigma_{j=1} e^{z_j}}$$

Onde z_1 representa a saída linear para a classe i e K é o número de classes

- *Sigmoid*: Nas tarefas de classificação binária, o *sigmoid* mapeia a saída linear para um valor entre 0 e 1, interpretado como a probabilidade de classe positiva. Como detectar a presença ou a ausência de uma característica específica.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

2.5. Detecção de objetos

Detecção diz respeito a uma tarefa de visão computacional de modelos de aprendizado profundo que tem como objetivo principal identificar e localizar instâncias de objetos dentro de imagens ou vídeos. Abaixo são listados alguns métodos diferentes de detecção.

2.5.1. Método Baseado em Dois Estágios

Este método funciona em duas etapas. A primeira consiste em delimitar regiões que podem conter alguma classe de interesse. Na segunda etapa são utilizadas classificações nessas regiões e um refinamento das propostas para determinar a localização dos objetos com mais precisão. As principais abordagens que utilizam este método são R-CNN, Fast R-CNN e Faster R-CNN. Este método possui uma grande desvantagem devido a necessidade de utilizar a rede em cada região por vez, causando alto consumo de memória e tempo de processamento. Este problema foi mitigado a partir da Fast R-CNN que passa a imagem inteira na rede gerando uma *feature map*, onde posteriormente serão projetadas as propostas de região.

2.5.2. Método de Detecção de Único Estágio

Diferentemente do método anterior, o método em um único estágio se destaca pela simplicidade e capacidade de realizar a detecção em um único passo, ou seja, uma única arquitetura de rede neural.

A rede neural convolucional extrai características da imagem, com padrões relevantes para a detecção. Posteriormente, a saída passa por camadas convolucionais adicionais, seguidas de camadas de predição que preveem diretamente as caixas delimitadoras e suas respectivas classes.

Essas características trazem vantagens como eficiência no tempo, simplicidade e flexibilidade. Por outro lado trazem mais dificuldades de detecção de objetos pequenos e de obter mais falsos positivos.

3. TRABALHOS RELACIONADOS

Esta seção apresenta as referências selecionadas a partir de pesquisas no *Google Scholar* e *ResearchGate*, referentes a modelos de redes neurais utilizados para a detecção de vivacidade em imagens, com o propósito de fundamentar o presente trabalho. Foram selecionados um total de cinco artigos e uma tese, os quais utilizam diferentes abordagens para lidar com o problema.

3.1. Detecção de Ataques de Apresentação por Faces em Dispositivos Móveis

A tese de Almeida (2018). Estuda o problema da detecção automática de ataques de apresentação em sistemas de biometria facial, utilizados em dispositivos móveis modernos. O foco do modelo apresentado é a segurança na autenticação de usuário para desbloqueio de dispositivos móveis através de reconhecimento facial, mais precisamente ataques a nível de sensor. Esses ataques podem ser feitos simplesmente mostrando ao sistema uma imagem da vítima, isso requer pouco conhecimento técnico, e além disso, imagens de face estão largamente disponíveis na internet. O modelo apresentado é focado nos ataques de foto impressa e tela, pois os ataques com máscara além de serem menos comuns, por serem feitos em superfícies 3D ao invés de superfícies planas (folhas, monitores LCD, tablets), exigem diferentes métodos.

As abordagens propostas para detecção de ataques de apresentação em dispositivos móveis são todas baseadas no treinamento de redes neurais convolucionais para distinguir entre um acesso real ou um ataque. Todas têm as mesmas camadas convolucionais e de *pooling* no seu núcleo, mas se diferenciam tanto no que veem como entrada na etapa de treinamento, quanto no que estão otimizando no final. A interação entre a entrada e o objetivo da otimização é o que de fato define o problema, direcionando o procedimento de aprendizagem.

3.1.1. Método I: Rede Neural Convolucional Treinada com Todas as Regiões do Rosto

O primeiro método é baseado em realizar o treinamento da rede neural convolucional profunda utilizando imagens alinhadas de toda a face, formato de entrada tradicional na maioria dos algoritmos publicados na literatura. O alinhamento é feito de forma a garantir uma região que contenha apenas a face. A resolução original é preservada evitando redimensionamento desnecessário nesse estágio. A figura 5 mostra o procedimento de treinamento para este método:

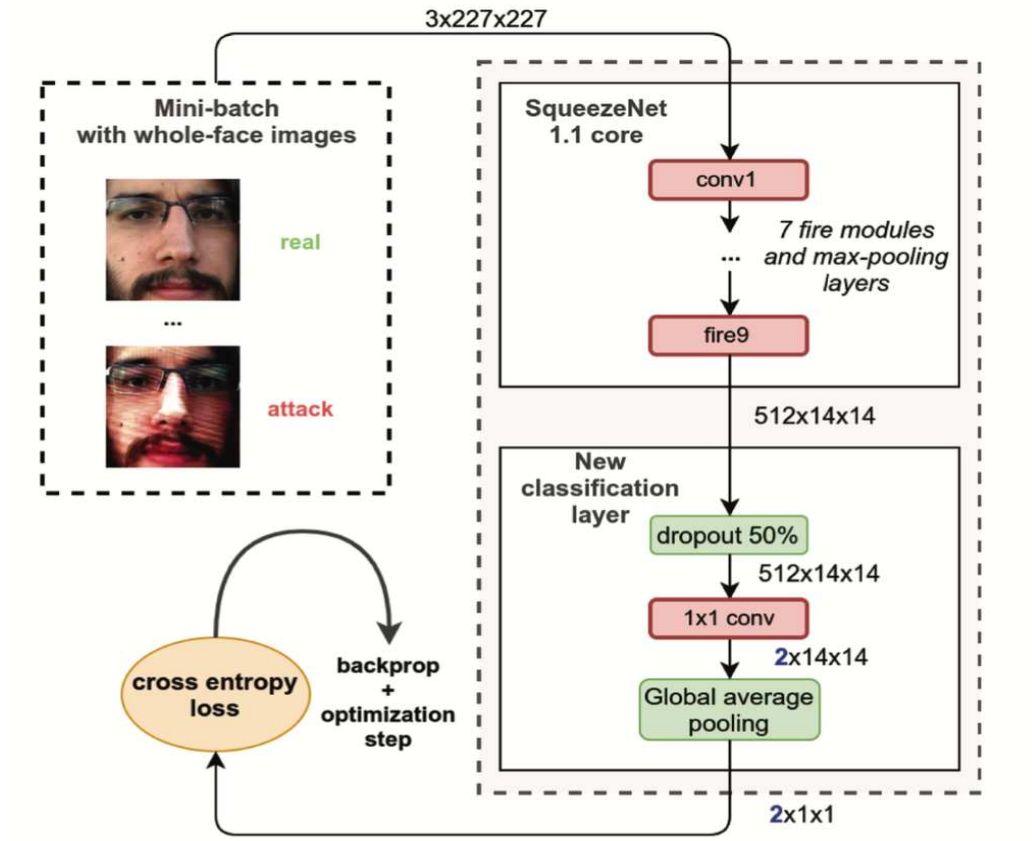


Figura 5 - Procedimento de treinamento para o método

Fonte: Waldir Rodrigues de Almeida, 2018

Para cada iteração são definidos *batches* com 64 imagens aleatórias do conjunto de treinamento. A chance de uma imagem ser selecionada é inversamente proporcional ao número de amostras com o mesmo rótulo no conjunto de treinamento. Portanto, cada *batch* vai possuir aproximadamente 32 amostras de rótulos reais e 32 amostras de ataques de apresentação. A arquitetura da rede escolhida é a *Squeezenet* pois a rede resultante é pequena e rápida o suficiente para ser incorporada diretamente em dispositivos móveis, e ela possui uma estrutura totalmente convolucional, facilitando a interpretação dos resultados. Depois de calculada a perda e as ativações intermediárias, o gradiente da perda em relação a cada parâmetro é calculado via retropropagação. Por fim, para a etapa de otimização foi utilizado um otimizador baseado em gradiente descendente, exigindo um ajuste mínimo dos hiperparâmetros.

3.1.2. Método II: Rede neural convolucional treinada com fragmentos da imagem

O segundo método utiliza a mesma arquitetura anterior, mas difere do primeiro modelando o problema com a tarefa de distinguir fragmentos da imagem em exemplos de ataque, de fragmentos de imagens genuínas. Para fazer isso foram extraídos pedaços de tamanhos variados das imagens com a resolução total e depois redimensionando os mesmos para caber no formato de entrada da rede. Essa abordagem aumenta o número de exemplos disponíveis para treinamento, força a rede a distinguir em diferentes resoluções potencialmente aumentando sua robustez para desfoque e iluminação adversa, e por fim, ao não utilizar imagens com a face total do

usuário a rede é incentivada a não depender de características específicas, o que pode reduzir *overfitting*. A figura 6 ilustra a construção de um *batch* para este método.

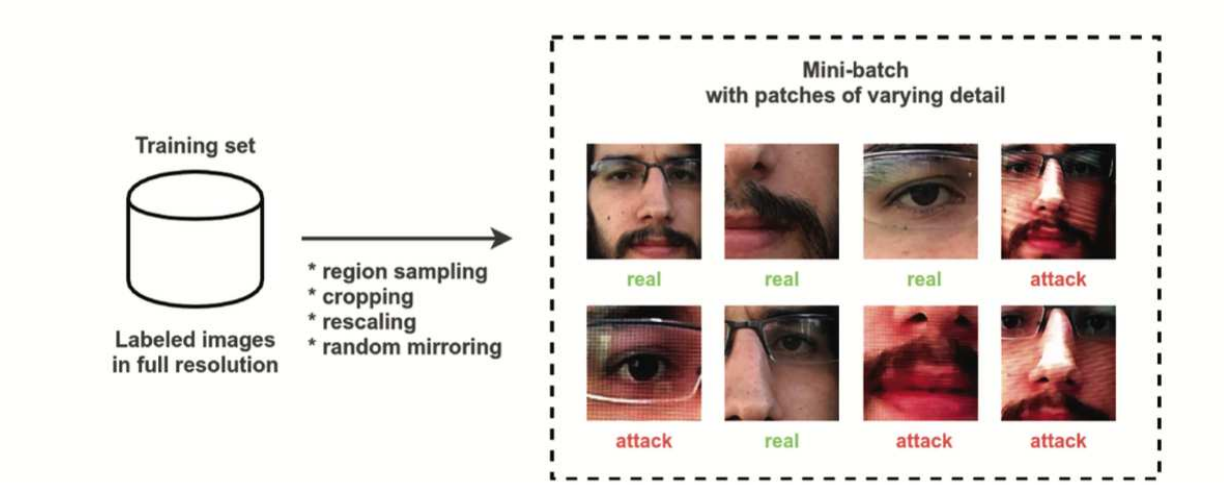


Figura 6 - Construção do *bath* para o método treinado com fragmentos da imagem

Fonte: Waldir Rodrigues de Almeida, 2018

3.1.3. Método III: Rede neural convolucional treinada com perda multi-objetivo

Inspirado por alguns questionamentos como:

- É razoável imaginar que amostras reais de diferentes dispositivos tenham características semelhantes?
- A classificação binária pura é a melhor forma de modelar o problema?
- Como orientar o treinamento de maneira que as representações aprendidas sejam propensas a identificar diferenças reais de amostras

reais versus amostras de ataque, ao invés do ruído dependente do conjunto de dados?

É proposto um terceiro método, onde o problema original é reformulado adicionando outro termo ao objetivo de treinamento. A ideia é que amostras reais de um determinado dispositivo sejam localizadas mais próximas em espaços de atributos intermediários, mas mais distantes de amostras de ataque para o mesmo dispositivo. A hipótese é que isso criaria melhores representações ao não confundir as informações para dispositivos diferentes, como nas estratégias de treinamento tradicionais.

O princípio da arquitetura da rede é o mesmo dos anteriores, e o pré-processamento de imagens é feito distinguindo fragmentos da imagem, exatamente como é feito no método II. A Figura 7 ilustra as diferenças de arquitetura e procedimento de treinamento para o método III.

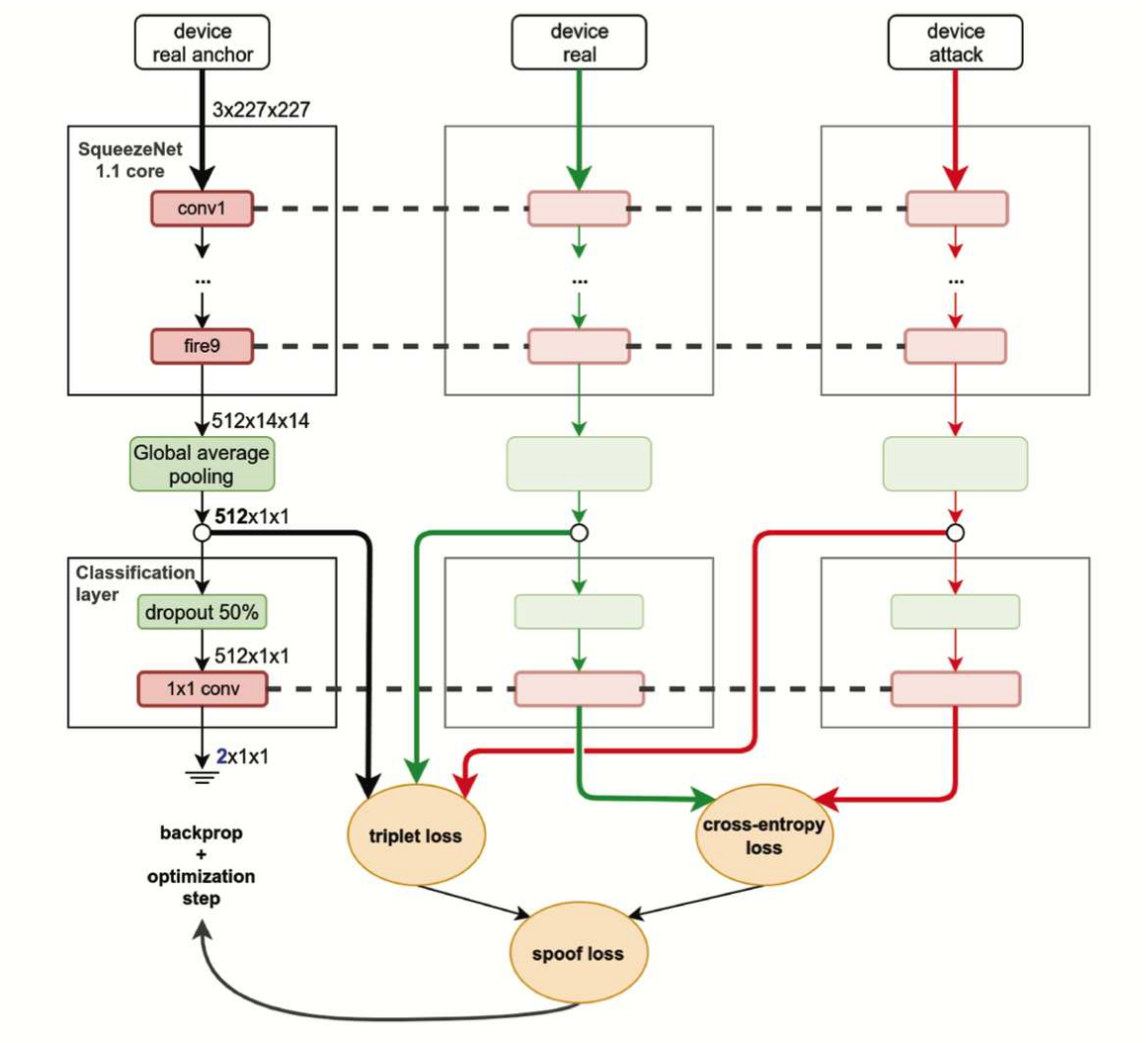


Figura 7 - Arquitetura utilizada para o método rede neural convolucional treinada com perda multi-objetivo

Fonte: Waldir Rodrigues de Almeida, 2018

Três colunas diferentes são formadas, uma com âncoras de acesso real, uma com amostras de acesso real e uma com amostras de ataque. A perda é calculada para cada trio e incentiva a compactação de um determinado dispositivo na camada de incorporação, enquanto afasta as amostras de ataque, até uma margem.

Para a construção dos modelos foram utilizados dois *datasets*, o RECOD-MPAD (RECOD *Mobile Presentation-Attack Dataset*), conjunto de dados coletados como parte deste trabalho, e o OULU-NPU, outro conjunto de dados visando o cenário de dispositivos móveis.

- Dataset I: RECOD-MPAD - O principal objetivo da criação deste conjunto de dados era ter dados que fossem representativos no cenário de desbloqueio rápido em dispositivos móveis, além disso cobrir o maior número possível de variações de iluminação, pois é um ponto faltante em conjuntos de dados públicos.

  Para a coleta dos dados foram usados dois *smartphones* com câmeras frontais bem diferentes entre si, os vídeos foram capturados em diferentes ambientes com cenários de iluminação diferentes, e os usuários foram instruídos a manter o dispositivo frontalmente em relação ao seu rosto e em seguida girar lentamente em torno de si, a fim de tornar o *dataset* o mais representativo possível dentro do escopo do problema.

Já as fotos foram obtidas coletando frames dos vídeos capturados.

- Dataset II: OULU-NPU - É um conjunto de dados público que visa a detecção de ataques de apresentação em dispositivos móveis. Seu principal mérito é incluir vídeos de acesso real e de ataques feitos com 6 câmeras diferentes de smartphones. Foi utilizado como um *benchmark* adicional, que permite comparar os métodos apresentados com outros métodos.

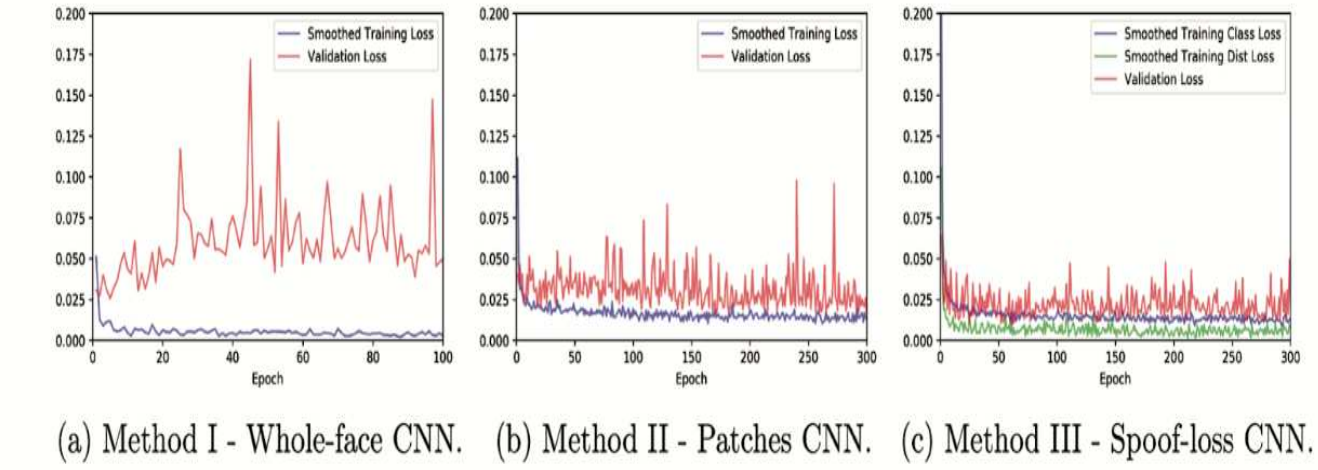
3.1.3. Avaliação e Resultados

Para avaliar a eficácia de modelos treinados, foram utilizadas as métricas padrão da literatura de detecção de ataque de apresentação. Para uma determinada amostra, a saída de cada algoritmo é uma pontuação que representa sua confiança de que a entrada deve ser classificada como ataque em vez de real. Todas as pontuações podem ser interpretadas como probabilidades no intervalo [0, 1].

No protocolo geral do RECOD-MPAD, são incluídos 143.997 frames abrangendo 45 usuários, 2 dispositivos sensores, 5 cenários de iluminação e 4 tipos de ataque. Esses frames são divididos em 3 subconjuntos separados:

- Um conjunto de treinamento - utilizado para realizar o treinamento do modelo;
- Um conjunto de validação - usado para ajuste de hiperparâmetros, como critério de parada ou para monitorar o procedimento de treinamento;
- Um conjunto de teste - usado exclusivamente para relatar os resultados finais de um método.

A figura 8 mostra as curvas gerais de treinamento para os três métodos propostos, em termos de treinamento e perda de validação em função das épocas de treinamento.



(a) Method I - Whole-face CNN. (b) Method II - Patches CNN. (c) Method III - Spoof-loss CNN.

Figura 8 - Curvas gerais de treinamento

Fonte: Waldir Rodrigues de Almeida, 2018

Em geral, todos os métodos superaram as *baselines* por uma grande margem. A tabela 1 demonstra os resultados:

	Conjunto de Validação			Conjunto de Testes		
Método	HTER	FAR	FRR	HTER	FAR	FRR
Color-LBP	3,87	1,02	6,72	4,94	0,72	9,16
Pre-trained CNN	3,62	2,62	4,61	6,59	2,32	10,86
Whole-face CNN (I)	0,91	0,40	1,43	0,82	0,35	1,29
Patches CNN (II)	0,56	0,67	0,44	1,14	0,34	1,94
Spoof-loss CNN (III)	0,35	0,29	0,42	0,63	0,12	1,15

Tabela 1 - Desempenho das diferentes propostas nos conjuntos de validação e testes

Fonte: Waldir Rodrigues de Almeida, 2018

O desempenho do conjunto de validação sugere que, os métodos II e III podem modelar melhor o problema, no sentido de que atingem pontos de menor erro de validação durante o treinamento. Este padrão também fica evidente em outros experimentos.

Na tese, são propostas três maneiras diferentes de treinar uma rede neural convolucional para modelar e resolver o problema de detecção de ataque de apresentação facial de uma maneira totalmente orientada a dados, além de uma nova base de dados representativa para este contexto. Focada nas restrições do cenário de dispositivos móveis, com suas peculiaridades de aquisição de dados e limitações de *hardware*.

Ao usar uma arquitetura convolucional profunda e poderosa, mas leve, como base, a abordagem orientada a dados permite que a concentração seja na definição do problema em si. Os modelos mostraram resultados promissores em termos de taxas de erro se comparadas ao estado da arte, porém os métodos de detecção de ataques de apresentação baseados em *software* único ainda não são suficientemente bons para problemas do mundo real. O maior desafio está nas limitações dos conjuntos de dados disponíveis.

3.2. LiveNet: Improving features generalization for face liveness detection using convolutional neural networks

Com a utilização geral dos recursos de biometria facial em sistemas de autenticação, tentativas de enganar o sistema forjando imagens da face de algum

indivíduo precisam ser levadas em consideração. Neste contexto as principais contribuições do artigo são:

- Apresentar uma estratégia eficiente de treinamento para redes neurais convolucionais em conjuntos de dados relacionados ao problema que possuem amostras de treinamento limitadas;
- Uma revisão do estado da arte para algoritmos anti falsificação de face;
- Os detalhes das redes neurais propostas, bem como seus resultados.

Ao revisar o estado da arte, é possível identificar que os algoritmos são projetados para a classificação binária, mas não têm um bom desempenho nos testes de validação cruzada. Além disso, os algoritmos não utilizam totalmente o aprendizado de ponta a ponta. (REHMAN et al., 2018) A figura 9 mostra o modelo proposto (c) e os convencionais (a) e (b) para o treinamento da rede.

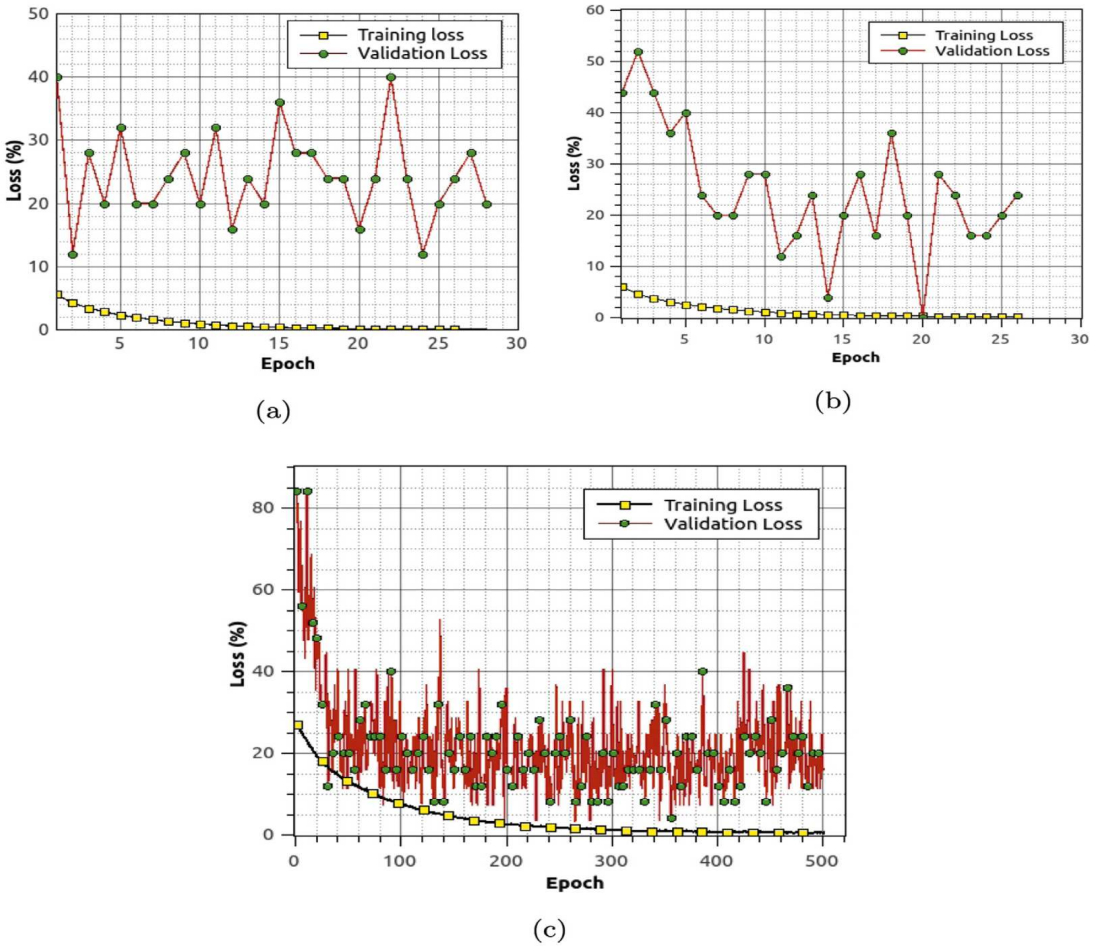


Figura 9 - Comparação entre o modelo proposto (c) e os convencionais (a) e (b)

Fonte: Rehmanet al., 2018

Como pode ser observado no método convencional, os dados são randomizados apenas uma vez antes do treinamento da rede. Porém na abordagem proposta que é semelhante ao *boot-strapping*, os dados são continuamente randomizados antes de serem aplicados a rede na forma de *mini-baches* individuais, o que reduz o problema de *overfitting* causado pela baixa quantidade de dados de treinamento. Os principais motivos de diminuição de *overfitting* na abordagem proposta são:

- A rede pode ser treinada em mais de um *frame* de vídeo como nas abordagens tradicionais, fazendo com que a rede possa aprender mais sobre recursos discriminativos nos dados de entrada;
- A rede pode ser treinada do zero utilizando aprendizado de ponta a ponta;
- Fornecer mais dados dos quadros de vídeo aumenta a capacidade de generalização da rede, algumas amostras dos dados de entrada podem não ser fornecidas como entrada durante uma época completa, mas são introduzidas após algumas épocas;
- O tempo de treinamento é reduzido significativamente,uma vez que no esquema de treinamento proposto o tamanho do *batch* é pequeno e as sub-épocas são 60.

O aspecto mais importante desse estudo é avaliar a capacidade de generalização da rede neural convolucional proposta em comparação com outras abordagens recentes.

Intra-database results: Comparison with other state-of-the-art method (Liveness Detection).

Method	Intra-database	
	CASIA (Test) HTER (%)	Replay-Attack (Test) HTER (%)
DPCNN (Li et al., 2016)	4.5	6.1
SpoofNet (Menotti et al., 2015)	–	0.75
LSTM + CNN (Xu et al., 2015)	–	5.93
Non-Linear Diffusion (Alotaibi & Mahmood, 2017)	–	10
Multi-cues Integration + NN (Feng, Po, Li, Xu et al., 2016)	5.83[a]	0
Pinto et al. (2015)	14.3	2.8
Siddiqui et al. (2016)	3.8	0
DDGL (Manjani et al., 2017)	1.3	0
LiveNet	4.59[a]	5.74

[a] EER = HTER

Figura 10 - Comparação do LiveNet com outras abordagens

Fonte: Rehman et al., 2018

A partir da figura 10 observamos que o HTER final em CASIA é um pouco maior ao de Li et al. (2016), Siddiqui et al. (2016) e Manjani et al. (2017) e da mesma forma para o banco de dados Replay-Attack o HTER é maior que o de Menotti et al. (2015), Feng, Po, Li, Xu et al. (2016), Pinto, Pedrini, Schwartz e Rocha (2015), Siddiqui et al. (2016) e Manjani et al. (2017). Indicando que, a randomização contínua de dados pode ajudar a treinar as redes profundas em bancos de dados de pequena escala e sem superajuste, porém pode impedir que algumas amostras passem pela rede, o que resulta num leve aumento de HTER nesse tipo de teste.

Cross-database Results: Comparison with other state-of-the-art method (Liveness Detection).

Method	Cross-database	
	CASIA (Test) HTER (%)	Replay-Attack (Test) HTER %
Pinto et al. (2015)	50	34.4
Siddiqui et al. (2016)	44.6	35.4
CNN (Yang et al., 2014)	38.11	23.78
DDGL (Manjani et al., 2017)	27.4	22.8
LiveNet	**19.12**	**8.39**

Figura 11 - Comparação em avaliação cruzada

Fonte: Rehmanet al., 2018

A figura 11 mostra uma comparação da abordagem proposta com outras abordagens do estado da arte para avaliação cruzada. Claramente as abordagens com baixo HTER na avaliação anterior, tem um valor mais alto na avaliação cruzada. As

redes propostas fornecem valores HTER significativamente mais baixos na avaliação cruzada, superando outras abordagens do estado da arte. Isso mostra que a rede proposta tem melhor capacidade de generalização. Além disso, a randomização de dados nos conjuntos, ou seja, uma combinação diferente de amostras alimentando cada passagem, permite que a rede aprenda mais sobre a variação de classe, melhorando a generalização e robustez dos recursos aprendidos.

O método proposto fornece uma maneira eficaz de melhorar recursos para generalização de redes neurais profundas para detecção de fraudes em biometria facial. A abordagem de randomização dos dados é uma forma de prevenir *overfitting* causados por poucos dados de treinamento. Por outro lado, ao mostrar aleatoriamente pequenos *batches* do conjunto de dados, existe a possibilidade de algumas amostras representativas, não passarem pela rede. Isso fica evidente por um aumento do HTER nos testes da primeira tabela, porém nos testes cruzados, o HTER é significativamente menor em relação a outras abordagens.

3.3. A lite convolutional neural network built on permuted Xceptio-inception and Xceptio-reduction modules for texture based facial liveness recognition

O reconhecimento da vivacidade facial tem principalmente dois tipos, dependendo da interação com o usuário. Um é o método intrusivo que leva em consideração as ações do usuário, como movimento da cabeça, olhos ou boca. Porém métodos não intrusivos não dependem de nenhuma interação no processo de tomada

de decisão. No mesmo contexto, o reconhecimento de vivacidade facial é categorizado em três tipos, levando em conta os atributos que serão utilizados. O primeiro utiliza propriedades da pele e iluminação da superfície da face para diferenciar uma amostra genuína de uma amostra falsa. O segundo método usa a imagem térmica e infravermelha para provar a vivacidade, porém o alto custo limita a utilização para contextos em aplicações altamente seguras. Já o terceiro método usa recursos de movimentos da cabeça, olhos e boca, o caracterizando como intrusivo. Como requer menos parâmetros e custo computacional em comparação com técnicas intrusivas, o modelo proposto utilizará o primeiro método que leva em consideração atributos relacionados a propriedades da pele e iluminação.(SATAPATHY et al., 2020).

Na figura 12 vemos a esquerda um exemplo de imagem real e a direita um exemplo de tentativa de fraude.

Figura 12 - Exemplos de imagem genuína (esquerda) e de ataque de apresentação (direita)

Fonte: Satapathy et al., 2020

3.3.1. Inception V2

Esta rede neural consiste em dois tipos de módulos inception, onde cada convolução 5 × 5 nos módulos inception foi substituída por duas convoluções 3 × 3 simétricas. A função de ativação ReLU é usada para resolver o problema do gradiente zero. O *GoogleNet* usa dez módulos iniciais de redução dimensional de camadas de convolução 1 × 1, 3 × 3, 5 × 5 e uma camada de pool 3 × 3 Max com valor de passo 1. O Inception V2 terminou com uma camada média global que substitui camadas totalmente conectadas e não possui parâmetros a serem otimizados. Classificadores suplementares são incluídos após algumas camadas intermediárias para resolver o problema de gradientes que desaparecem.

3.3.2. Xception

A camada de convolução separável em profundidade realiza a convolução em um único canal ou em um grupo de canais para produzir sua saída de destino, igual ao número de filtros. Ele divide uma convolução normal em dois conjuntos distintos de núcleos, e o primeiro executa a convolução separável em profundidade, enquanto o segundo faz a convolução pontual para produzir o mesmo número de canais do mesmo tamanho como uma convolução normal, mas com muito menos operações e pesos. O módulo Xception usa convolução pontual para extrair a correlação entre os canais, e as saídas dos módulos convolucionais separáveis em profundidade foram concatenadas para produzir o resultado final. O módulo Xception torna a operação simples e mais

produtiva, segregando as correlações intracanal e intercanal existentes nos mapas de recursos de uma camada convolucional trivial.

3.3.3. Arquiteturas Propostas

3.3.3.1. Permuted Xceptio-reduction module

O módulo de redução no Inception V2 é aprimorado em precisão usando camadas de permuta, nivelamento e remodelação para estabelecer correlação cruzada e espacial entre os canais, mesmo após a redução do tamanho do mapa de atributos. O módulo pega a saída de uma camada anterior e aplica uma camada de agrupamento máximo de 3x3 e duas convoluções pontuais diferentes com tamanhos de filtro variados. Os mapas de atributos resultantes são inseridos em camadas convolucionais separáveis em profundidade 3x3. Cada camada convolucional 3x3 é substituída por uma convolução 3x3 em profundidade seguida por uma convolução pontual. As saídas dessas camadas são permutadas, achatadas e concatenadas para gerar um vetor 1D. A camada de remodelação cria um tensor 3D a partir do vetor, mantendo a correlação cruzada e espacial entre os canais. Essas camadas ajudam a manter a correlação entre dois módulos de iniciação, que podem ser afetados pelo uso de convolução separável em profundidade em um módulo Xception.

3.3.3.2. Permuted Xceptio-inception module

O módulo Permuted Xceptio-Inception é uma reestruturação do módulo Inception da rede Inception v2 com convoluções em profundidade, permutação, nivelamento, adição e remodelação de camadas. Cada módulo de iniciação foi atualizado para suportar a ideia de uma rede residual para resolver o problema dos gradientes de fuga durante a retropropagação e para resolver a maldição dos problemas de dimensionalidade. A arquitetura de rede neural proposta é uma versão redesenhada do Inception V2 para reconhecimento facial de vivacidade, que se concentra na convolução separável em profundidade e na rede residual para construir uma rede neural leve e mais eficiente. Ele usa uma imagem 224 × 224 de três canais como entrada e extrai recursos usando os módulos Permuted Xceptio-Inception e Reduction. A saída dos módulos é então passada por uma camada de agrupamento para obter um vetor de recursos 1D que é usado para decidir os valores de probabilidade para diferentes classes. A arquitetura proposta reduz o número de conexões com os neurônios na rede e minimiza o número de pesos, mantendo a correlação cruzada e espacial entre os canais com a ajuda de camadas intermediárias de permuta, achatamento e remodelação.

3.3.4. Resultados

O artigo apresenta a avaliação de desempenho de diferentes modelos de redes neurais convolucionais para autenticação facial usando diferentes conjuntos de dados. O modelo de rede neural proposto é avaliado em relação a modelos de redes neurais

populares, em conjuntos de dados FRAUD2, NUAA e CASIA-FASD. O artigo apresenta métricas de desempenho como exatidão, precisão e valores de recall, com diferentes taxas de aprendizado inicial e otimizadores para diferentes esquemas de particionamento desses conjuntos de dados. Como demonstrado na figura 13, o modelo proposto mostra melhor desempenho nos conjuntos de dados FRAUD2 e NUAA do que outros modelos. Os resultados mostram que o modelo proposto superou outros modelos em termos de exatidão, precisão e valores de *recall* nesses conjuntos de dados. No entanto, o desempenho dos modelos varia para diferentes esquemas de particionamento e conjuntos de dados.

Related Work	NUAA [ACC./ HTER/EER][%]	CASIA-FASD [ACC./HTER/EER][%]
Boulkenafet, Z., Komulainen, J., & Hadid, A. (2015)	–	6.2% EER
Parveen, S. et al. (2016)	94.5% ACC.	94.4% ACC.
Beham, M. P. et al. (2017)	3.02% HTER	9.21% HTER
Alotaibi, A., & Mahmood, A. (2017)	99.0% ACC.	–
Wang, S. et al. (2017)	98.56% ACC.	93.24% ACC.
Luan, X. et al. (2017)	98.8% ACC.	–
Dong, J., Tian, C., & Xu, Y. (2017)	–	5.68% EER
Qin, Le, et al. (2017)	–	10.23% EER
Benlamoudi, A. et al. (2017)	–	4.62% EER
Li, L., Correia, P. L. & Hadid, A. (2017)	–	6.2% EER
Mhou, K., Haar, D. V. & Leung, W. S. (2017)	–	72% ACC.
Zhao, X., Lin, Y. & Heikkila, J. (2017)	–	6.48% EER
Rehman, Y. A. U., Po, L. M. & Liu, M. (2017)	–	92.52% ACC.
Beham, M. P., & Roomi, S. M. M. (2018)	99.78% ACC.	–
Angadi, S. A., & Kagawade, V. C. (2018)	98.97% ACC.	–
Şengur, A. et al. (2018)	88.09% ACC.	94.01% ACC.
Vanitha, A., Vaidehi, V., & Vasuhi, S. (2018)	92% ACC.	–
Sthevanie, F., & Ramadhani, K. N. (2018)	99.07% ACC.	–
Kusuma, I. et al. (2018)	87.22% ACC.	–
Larbi, K. et al. (2018)	–	10.68% HTER
Zhang, L. et al. (2018)	–	8.0% EER
Lee, C. E. et al. (2018)	–	94.68% ACC.
Van der Haar, D. T. (2018)	–	12.1% EER
Hao, H., Pei, M., & Zhao, M. (2019)	1.96% HTER	–
Koshy, R., & Mahmood, A. (2019)	100% ACC.	–
Song, L. & Ma, H. (2019)	2.16% HTER	7.22% HTER
Pan, S., & Deravi, F. (2019)	–	4.8% EER
Pérez-Cabo, D. et al. (2019)	–	16.74% HTER
Yang, T. et al. (2020)	98.2% ACC.	–
Yılmaz, A. G., Turhal, U., & Nabiyev, V. V. (2020)	12.18% HTER	–
Raghavendra, R. J., & Kunte, R. S. (2020)	96.41% ACC.	–
Song, X. et al. (2020)	–	96.10% ACC., 5.06% EER, 4.41% HTER
Yao, C. et al. (2020)	–	2.96% HTER
Zhang, W., & Xiang, S. (2020)	–	94.44% ACC., 5.6% EER
Pujol, F. A. et al. (2020)	–	9.5% EER
PXIR CNN (Our model)	100% ACC.	94.44% ACC., 5.56% HTER, 5.59% EER

Figura 13 - Comparação do método PXIR CNN com outras abordagens.

Fonte: Satapathy et al., 2020

3.3.5. Conclusão

O artigo discute a importância do reconhecimento de vivacidade facial na tecnologia de reconhecimento facial para diferenciar entre rostos vivos e falsos, o que geralmente é comprometido por ataques de impostores. O artigo enfoca o uso de redes neurais convolucionais para reconhecimento de vivacidade facial, discutindo especificamente o Inception V2 e seus diferentes módulos. O artigo também propõe um modelo de rede neural convolucional lite baseado nos módulos Xceptio-Inception e Xceptio-Reduction permutados e o compara com o Inception V2 em termos de exatidão, precisão, *recall* e número de parâmetros ponderados. O modelo proposto tem um desempenho melhor do que o Inception V2 e outras redes neurais padronizadas e requer menos espaço de memória, tornando-o mais adequado para implementação em tempo real.

3.4. You Only Look Once: Unified, Real-Time Object Detection

O artigo de Redmon et al.(2016) faz um estudo sobre a rede YOLO. Esta abordagem enxerga a detecção de objetos como um problema de regressão, do pixel da imagem as coordenadas das caixas delimitadoras e as probabilidades de cada uma das classes, esta rede convolucional faz a predição das múltiplas caixas delimitadoras e suas respectivas classes simultaneamente. Treinando em imagens completas, otimiza diretamente o desempenho de detecção. Dentre as principais vantagens desta abordagem podem ser citadas a rapidez, raciocínio global, ou seja, analisar toda a imagem durante o treinamento e a inferência, incluindo informações de contexto sobre

cada classe. Essa característica resulta em menos erros de fundo comparados a outros métodos tradicionais. Apesar desses pontos existem algumas limitações na precisão, especialmente ao localizar objetos pequenos, porém outra vantagem é que esta abordagem segue em código aberto.

Para o treinamento a pesquisa o modelo foi pré-treinado no dataset ImageNet para melhorar a detecção de objetos, utilizando uma arquitetura que alcança 88% de precisão. Após o pré-treinamento, são adicionadas camadas convolucionais e totalmente conectadas, com a resolução de entrada aumentada para 448 × 448. A função de custo é ajustada para balancear erros de localização e classificação, e técnicas de estabilização do treinamento, como aumento de dados e dropout, são empregadas para evitar o overfitting. A rede é treinada por 135 épocas nos conjuntos de dados PASCAL VOC 2007 e 2012, utilizando uma taxa de aprendizado que varia ao longo do treinamento. Esse processo ilustra a eficácia do pré-treinamento e adaptação de modelos para detecção de objetos em tempo real.

Nesse estudo, foi realizada uma comparação entre a YOLO e outros sistemas de detecção em tempo real, com o dataset PASCAL VOC 2007. Destacando diferenças entre YOLO e variantes do R-CNN, explorando os erros. Com base nos diferentes perfis de erros, é demonstrado neste estudo que o YOLO pode ser utilizado para reavaliar as detecções do Fast R-CNN reduzindo erros de falsos positivos de fundo. Também são apresentados resultados no VOC 2012 comparando a precisão média (mAP) com outros métodos.

Para examinar as diferenças entre YOLO e outros detectores no estado da arte, foi utilizado especificamente o Fast R-CNN, utilizando a metodologia de Hoiem et al.

Classificando cada predição em diferentes categorias de erro: correto, erro na localização, classe similar, erro na classe e fundo. Nesse contexto observou-se que YOLO tem dificuldades significativas em erros de localização, e o Fast R-CNN em erros de fundo. Analisando estas características, YOLO foi combinado com Fast R-CNN para melhorar o desempenho em detecções de fundo, como resultado a mAP do Fast R-CNN teve um aumento de 71.8% para 75.0% no conjunto de testes VOC 2007, como pode ser visto na figura 14. Levando em consideração que os modelos são executados separadamente , a combinação não se beneficia da velocidade da YOLO, apesar de não adicionar tempo computacional significativo em relação ao Fast R-CNN. Conforme demonstrado na figura 15, nos testes realizados utilizando VOC 2012, YOLO consegue uma mAP de 57,9%, inferior a outros métodos do estado da arte, graças a sua maior dificuldade com objetos pequenos, apresentando um desempenho de 8% a 10% inferior nesses casos, porém a combinação entre Fast R-CNN e YOLO resulta em uma das detecções mais eficientes, aumentando em 2,3% a mAP do Fast R-CNN, elevando em 5 lugares sua posição.

YOLO é um modelo unificado para detecção de objetos, simples de construir e treinado diretamente em imagens completas. Diferentemente de outras abordagens baseadas em classificadores, o YOLO é treinado em uma função de perda que corresponde diretamente ao desempenho da detecção, e todo o modelo é treinado conjuntamente. Além disso, o YOLO também generaliza bem para novos domínios, sendo ideal em aplicações que necessitam de uma detecção de objetos rápida e robusta.

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**

Figura 14 - Resultados ao combinar vários modelos com a melhor versão do Fast R-CNN.

Fonte: Redmon et al., 2016

Outras versões do Fast R-CNN fornecem apenas um pequeno benefício, enquanto o YOLO oferece um aumento significativo no desempenho.

VOC 2012 test	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
MR_CNN_MORE_DATA [11]	**73.9**	**85.5**	**82.9**	**76.6**	**57.8**	**62.7**	**79.4**	77.2	86.6	**55.0**	**79.1**	**62.2**	87.0	**83.4**	**84.7**	78.9	45.3	73.4	65.8	80.3	74.0
HyperNet_VGG	71.4	84.2	78.5	73.6	55.6	53.7	78.7	**79.8**	87.7	49.6	74.9	52.1	86.0	81.7	83.3	**81.8**	**48.6**	**73.5**	59.4	79.9	65.7
HyperNet_SP	71.3	84.1	78.3	73.3	55.5	53.6	78.6	79.6	87.5	49.5	74.9	52.1	85.6	81.6	83.2	81.6	48.4	73.2	59.3	79.7	65.6
Fast R-CNN + YOLO	70.7	83.4	78.5	73.5	55.8	43.4	79.1	73.1	**89.4**	49.4	75.5	57.0	**87.5**	80.9	81.0	74.7	41.8	71.5	68.5	**82.1**	67.2
MR_CNN_S_CNN [11]	70.7	85.0	79.6	71.5	55.3	57.7	76.0	73.9	84.6	50.5	74.3	61.7	85.5	79.9	81.7	76.4	41.0	69.0	61.2	77.7	72.1
Faster R-CNN [28]	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
DEEP_ENS_COCO	70.1	84.0	79.4	71.6	51.9	51.1	74.1	72.1	88.6	48.3	73.4	57.8	86.1	80.0	80.7	70.4	46.6	69.6	**68.8**	75.9	71.4
NoC [29]	68.8	82.8	79.0	71.6	52.3	53.7	74.1	69.0	84.9	46.9	74.3	53.1	85.0	81.3	79.5	72.2	38.9	72.4	59.5	76.7	68.1
Fast R-CNN [14]	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	**87.5**	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
UMICH_FGS_STRUCT	66.4	82.9	76.1	64.1	44.6	49.4	70.3	71.2	84.6	42.7	68.6	55.8	82.7	77.1	79.9	68.7	41.4	69.0	60.0	72.0	66.2
NUS_NIN_C2000 [7]	63.8	80.2	73.8	61.9	43.7	43.0	70.3	67.6	80.7	41.9	69.7	51.7	78.2	75.2	76.9	65.1	38.6	68.3	58.0	68.7	63.3
BabyLearning [7]	63.2	78.0	74.2	61.3	45.7	42.7	68.2	66.8	80.2	40.6	70.0	49.8	79.0	74.5	77.9	64.0	35.3	67.9	55.7	68.7	62.6
NUS_NIN	62.4	77.9	73.1	62.6	39.5	43.3	69.1	66.4	78.9	39.1	68.1	50.0	77.2	71.3	76.1	64.7	38.4	66.9	56.2	66.9	62.7
R-CNN VGG BB [13]	62.4	79.6	72.7	61.9	41.2	41.9	65.9	66.4	84.6	38.5	67.2	46.7	82.0	74.8	76.0	65.2	35.6	65.4	54.2	67.4	60.3
R-CNN VGG [13]	59.2	76.8	70.9	56.6	37.5	36.9	62.9	63.6	81.1	35.7	64.3	43.9	80.4	71.6	74.0	60.0	30.8	63.4	52.0	63.5	58.7
YOLO	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
Feature Edit [33]	56.3	74.6	69.1	54.4	39.1	33.1	65.2	62.7	69.7	30.8	56.0	44.6	70.0	64.4	71.1	60.2	33.3	61.3	46.4	61.7	57.8
R-CNN BB [13]	53.3	71.8	65.8	52.0	34.1	32.6	59.6	60.0	69.8	27.6	52.0	41.7	69.6	61.3	68.3	57.8	29.6	57.8	40.9	59.3	54.1
SDS [16]	50.7	69.7	58.4	48.5	28.3	28.8	61.3	57.5	70.8	24.1	50.7	35.9	64.9	59.1	65.8	57.1	26.0	58.8	38.6	58.9	50.7
R-CNN [13]	49.6	68.1	63.8	46.1	29.4	27.9	56.6	57.0	65.9	26.5	48.7	39.5	66.2	57.3	65.4	53.2	26.2	54.5	38.1	50.6	51.6

Figura 15 - Comparação da YOLO e Fast R-CNN + YOLO em diversas métricas.

Fonte: Redmon et al., 2016

3.5. Biometric Face Presentation Attack Detection With Multi-Channel Convolutional Neural Network

Em geral, a maioria dos métodos de Detecção de Autenticidade de Imagens (PAD) baseados em espectro visível têm como objetivo identificar diferenças sutis na qualidade da imagem quando esta é recapturada. No entanto, à medida que a qualidade de captura de dispositivos e impressoras melhora, esse método pode apresentar falhas. Com o intuito de mitigar esse problema, propõe-se uma abordagem baseada em rede neural convolucional multicanal para detecção de autenticidade (ANJITH et al., 2020). A ideia principal consiste em utilizar a representação conjunta de múltiplos canais para a detecção de autenticidade, empregando aprendizado de transferência a partir de uma rede de reconhecimento facial pré-treinada.

Anjith et al. (2020) estenderam essa ideia para a tarefa de Detecção de Ataque de Apresentação (PAD) multicanal, aproveitando informações complementares de representações conjuntas obtidas de vários canais. A estrutura proposta permite a reutilização de dados de reconhecimento facial disponíveis, quando há uma escassez de dados para o treinamento de sistemas PAD. Foi utilizado um modelo denominado Light CNN, pré-treinado em imagens faciais para reconhecimento facial, o qual possui um número menor de parâmetros em comparação a outras redes. Essa rede utiliza a operação *Max-Feature Map* em vez de unidades lineares retificadas (ReLU) para suprimir neurônios de baixa ativação.

3.6. Integration of image quality and motion cues for face anti-spoofing: A neural network approach

A abordagem *antispoofing* facial proposta utiliza a integração multi-sugestões para combinar recursos de vivacidade de três aspectos: qualidade de imagem baseada em *shearlet* (SBIQF), movimento facial baseado em fluxo óptico e movimento de cena baseado em fluxo óptico. O processo envolve a extração de um vetor SBIQF de uma imagem facial normalizada, a coleta de um vídeo facial com as mesmas coordenadas faciais e o cálculo do fluxo óptico denso entre os quadros faciais. A média das informações do fluxo óptico é usada para obter um mapa médio de movimento facial, que é alimentado em uma sub-rede para extrair uma representação de gargalo. Além disso, um mapa médio de movimento de cena é calculado a partir do vídeo da cena original. Esse mapa é utilizado como entrada para outra sub-rede que também extrai uma representação de gargalo. As três representações de gargalo são concatenadas e alimentadas em uma rede neural subsequente para detecção de vivacidade, que usa um classificador softmax de duas classes. As sub-redes antes da camada oculta II são conectadas localmente com entradas visuais diferentes e treinadas separadamente. O recurso de gargalo fundido na camada oculta II está totalmente conectado com a rede seguinte e é treinado em camadas com a camada oculta I.

3.7. Tabela Comparativa

A tabela 2 apresenta uma comparação entre os modelos dos trabalhos relacionados, utilizando a métrica HTER, a qual é calculada por meio da média entre as

taxas de falsos positivos e falsos negativos). Essa métrica é amplamente empregada em sistemas biométricos, sendo que um valor menor indica um desempenho superior do sistema, uma vez que reflete uma taxa de erro mais baixa.

As demais colunas da tabela correspondem a informações relevantes para os resultados e para este trabalho, tais como o dataset utilizado e sua natureza (acesso).

Trabalho	Rede	dataset	Acesso	Métrica	Resultado
Almeida 2018	Whole-face CNN	RECOD-MPAD	Criado para o trabalho	HTER	0,82%
Almeida 2018	Patches CNN	RECOD-MPAD	Criado para o trabalho	HTER	1,14%
Almeida 2018	Spoof-Loss CNN	RECOD-MPAD	Criado para o trabalho	HTER	0,63%
Rehman et al. 2018	LiveNet	CASIA-FASD	Público	HTER	19,12%
Satapathy et al. 2020	Xceptio-reduction	CASIA-FASD	Público	HTER	5,56%
Redmon et al. 2016	Fast R-CNN + YOLO	PASCAL VOC 2007	Público	-	-
Anjith et al. 2020	MC - CNN	WMCA DATASET	Restrito	HTER	0,76%
Feng et al. 2016	SBIQF	REPLAY-ATTACK	Público	HTER	6,13%

Tabela 2 - Síntese das técnicas utilizadas nos trabalhos correlatos

3.8. Considerações Sobre os Trabalhos Relacionados

Analisando os trabalhos relacionados e a tabela, a rede *Spoof-Loss* CNN teve bons resultados e possibilidade de melhora do seu desempenho ao utilizar uma base de dados mais heterogênea. E o método de treinamento proposto para a rede LiveNet permitindo a redução do overfitting mesmo em bases de dados menores, tem um potencial de aprimoramento variando seus hiperparâmetros ou utilizando outras redes neurais convolucionais como a inceptionV2 que também foi utilizada no trabalho da seção 4.3.

4. DESENVOLVIMENTO

A partir do conhecimento vindo da fundamentação teórica e dos trabalhos relacionados, as decisões para implementação serão de utilizar detecção de objetos a fim de identificar artefatos nas imagens que possam denunciar uma tentativa de fraude, classificando como ataque caso alguma instância das classes definidas no processo de rotulação seja identificada, e como imagem legítima caso nenhum objeto seja encontrado, para assim avaliar e comparar com os trabalhos do capítulo 3. A figura 16 a seguir demonstra a estrutura do desenvolvimento.

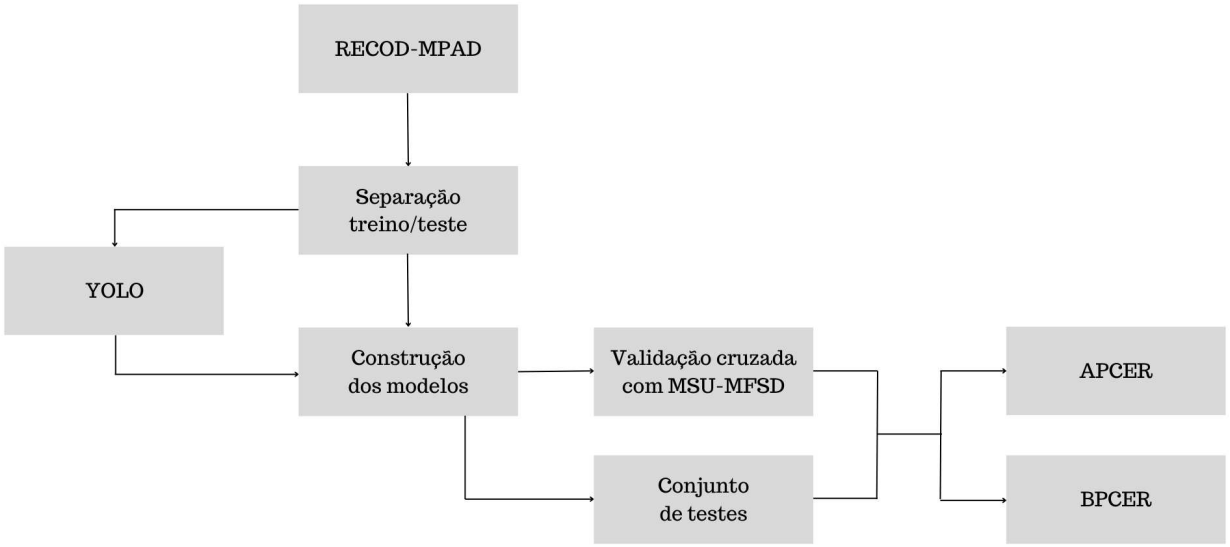


Figura 16 - Diagrama geral

Fonte: Elaborado pelo autor

4.1. Rede Selecionada - YOLOv8

A YOLOv8 é uma rede de visão computacional desenvolvida pela Ultralytics que suporta diversas tarefas como detecção, segmentação, classificação, detecção orientada de objetos e pontos chave de objeto.

Dentre as características relevantes do objetivo deste trabalho presentes nesta abordagem podem ser citadas:

- Disponibilidade de implementações: A YOLO segue em código aberto, existindo diversas implementações disponíveis publicamente, facilitando sua utilização.
- Robustez: Diferente de propostas baseadas em regiões e classificação, que dividem as tarefas, a YOLO trata a detecção de objetos como um problema de regressão único, permitindo aprender características robustas e discriminativas

diretamente dos dados. (REDMON et al., 2016). Além disso, versões mais recentes da YOLO como a utilizada neste trabalho, utilizam backbones otimizados para capturar características em diferentes condições de iluminação, que é uma característica relevante dentro do contexto da vivacidade.

- Eficiência de tempo e Capacidade de Detecção rápida: Como explicado na seção 2.5.2, por tratar a detecção como um problema de regressão único, em vez de dividir a tarefa em estágios diferentes, dividindo a imagem em uma grade e fazer previsões diretas das caixas delimitadoras e classes em cada grade da célula em uma única etapa, fazendo uma abordagem direta de detecção, a YOLO evita sobrecarga computacional. Essa combinação de fatores torna a YOLO uma ótima opção para detecção em tempo real, que é ideal para integrar em aplicações de segurança como a proposta neste trabalho.

  A figura 17 mostra a arquitetura detalhada da rede.

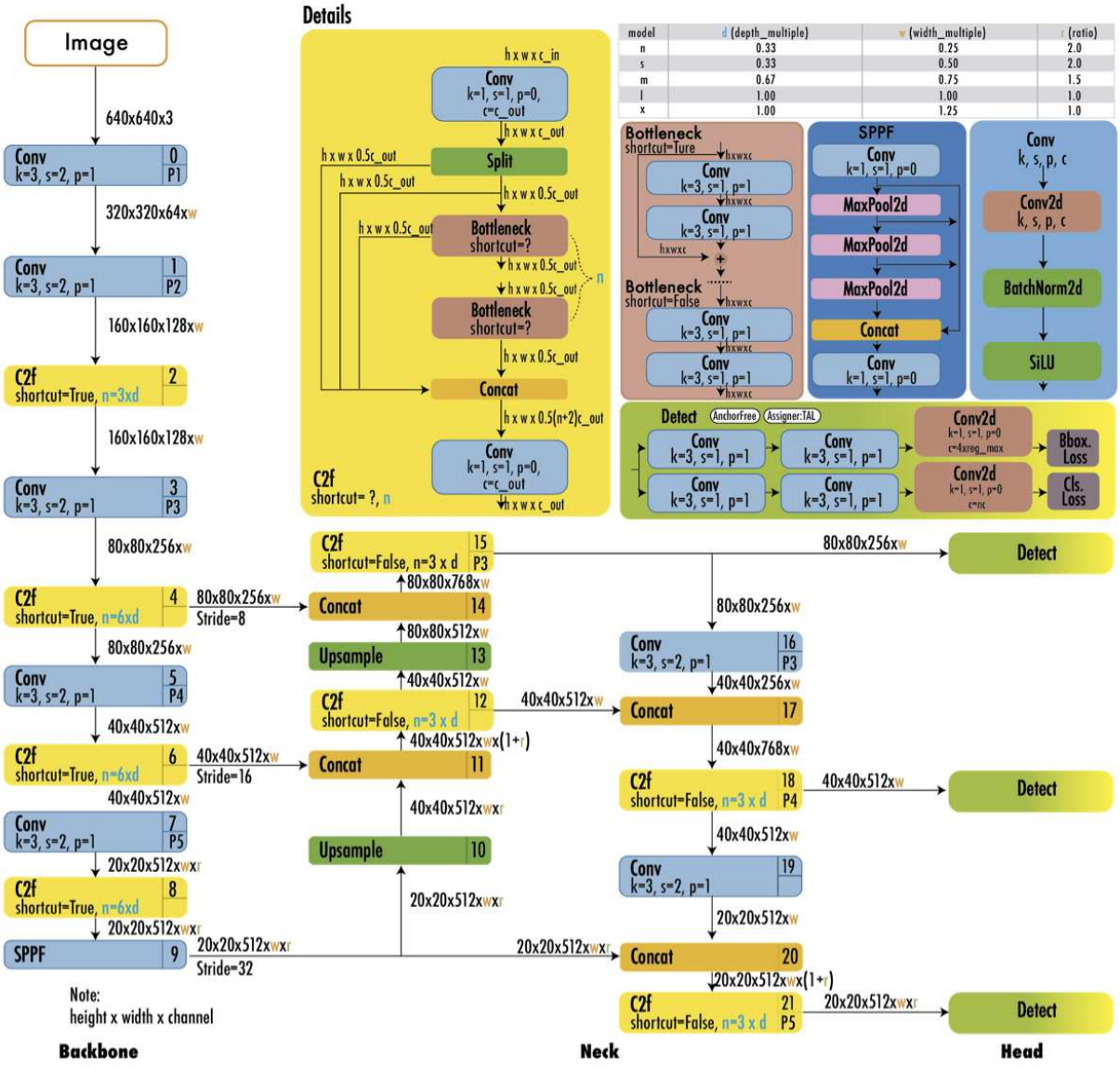


Figura 17 - Arquitetura da rede YOLOv8.

Fonte: Terven ,2024.

A arquitetura da YOLOv8n pode ser dividida em três partes principais:

- *Backbone*: Extrai características fundamentais das imagens de entrada;

- *Neck*: Combinar as características extraídas pelo backbone para melhorar a detecção dos objetos;
- *Head*: Realiza a previsão final das caixas delimitadoras.

A YOLOv8 utiliza um modelo sem âncoras, ou seja, sem caixas delimitadoras predefinidas como referências para prever a localização e tamanho dos objetos, com a head desacoplada, assim processa independente a tarefas como classificação e regressão, permitindo que cada ramo se concentre na sua tarefa.

4.2. Métricas Utilizadas

A avaliação será realizada utilizando três métricas principais que são comumente usadas na análise de desempenho de sistemas biométricos sendo elas:

- $APCER = \frac{Número\ de\ Apresentações\ de\ ataque\ Incorretamente\ Aceitas}{Número\ Total\ de\ Apresentações\ de\ Ataque}$
- $BPCER = \frac{Número\ de\ Apresentações\ Autênticas\ Incorretamente\ Rejeitadas}{Número\ Total\ de\ Apresentações\ de\ Autênticas}$
- $HTER = \frac{APCER + BPCER}{2}$

Para avaliação dos resultados do treinamento, foram utilizadas duas métricas adicionais:

- Precisão - mede o quão confiável é um modelo de detecção ao identificar um objeto, ou seja, mede a proporção de detecções corretas feitas pelo modelo, em relação ao total de predições. Pode ser calculado pela fórmula:
    - $Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos}$

- *Recall* - mede a capacidade do modelo em identificar todas as instâncias positivas dos objetos, através da relação entre os verdadeiros positivos e o total de instâncias presentes nas imagens. Pode ser calculado pela fórmula:
    - $Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos}$

4.3. Datasets utilizados

4.3.1. RECOD-MPAD

O *dataset* utiliza de dois dispositivos com câmeras frontais diferentes para fazer a captura: o smartphone Moto G5 lançado em 2017 e o smartphone Moto X Style XT1572 de finais de 2015, em cinco cenários de iluminação diferentes que englobam sessões ao ar livre, com luz solar direta ou sombra, e em ambientes internos com iluminação artificial ou natural e também com luzes apagadas, como demonstrado nas figuras 17 e 18. Os ataques estão divididos entre ataques de exibição (*displays*) e ataques de foto impressa. O *dataset* é construído com 64 quadros igualmente espaçados de cada vídeo de resolução 1920x1080 gerando um número final de 142.997 quadros, contendo 45 usuários diferentes.

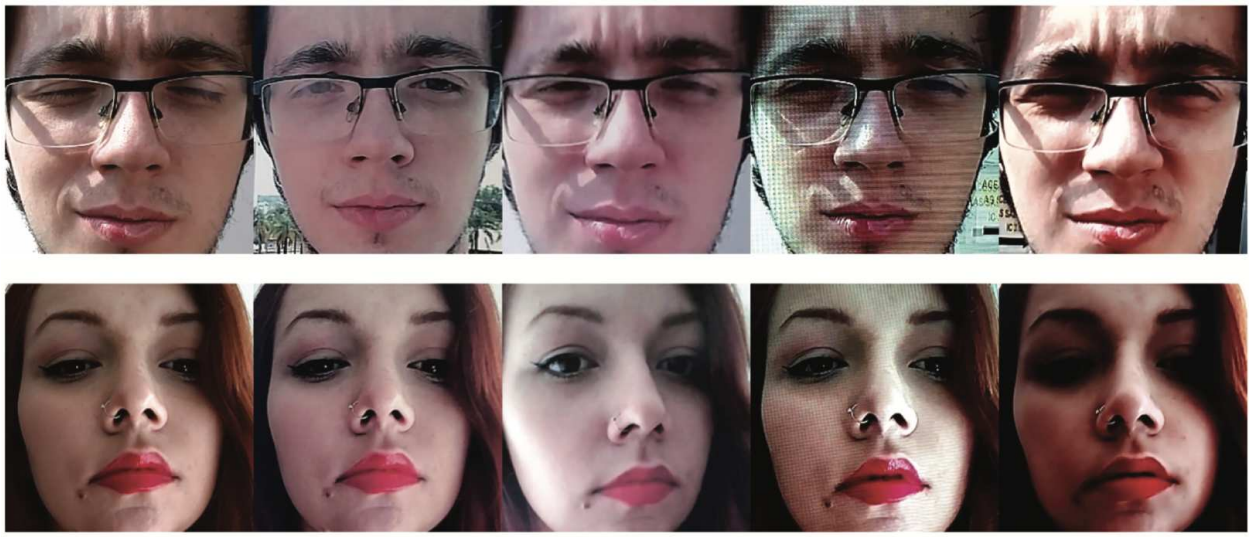


Figura 18 - Exemplo de ataques nos diferentes ambientes no RECOD-MPAD. Da esquerda para a direita: real, ataque impresso 1, ataque impresso 2, ataque com *display* 1, ataque com *display* 2.

Fonte: Waldir Rodrigues de Almeida, 2018

Figura 19 - Exemplo de imagens genuínas nos diferentes ambientes no RECOD-MPAD.

Fonte: Waldir Rodrigues de Almeida,2018

4.3.2. MSU-MFSD

O conjunto MSU-MFSD contém 280 gravações de vídeos, entre ataques e vídeos genuínos, feitos com dois tipos de câmeras em diferentes resoluções (720x480 e 640x480) com 35 indivíduos diferentes. Foram produzidos ataques de *display* a partir da tela de um iPad Air e de um iPhone, além de ataques impressos dos 35 indivíduos em papéis A3 utilizando uma impressora colorida. Como mostrado na figura 20.

Figura 20 - Exemplos de amostras do MSU-MFSD. A primeira linha corresponde a imagens tiradas com um telefone Android, enquanto a segunda linha mostra imagens capturadas com uma câmera de laptop. Da esquerda para a direita: rostos reais e os respectivos ataques de iPad, iPhone e impressão.

Fonte: Boulkenafet, 2016.

4.4. Rotulação

Para fazer a rotulação foi utilizado o Roboflow, que facilita esse processo fornecendo as ferramentas para delimitar os objetos nas imagens, além de recursos para acelerar o processo. Também foi possível fazer o download dos dados anotados já no formato compatível com a YOLOv8. As classes definidas são explicadas na tabela 3:

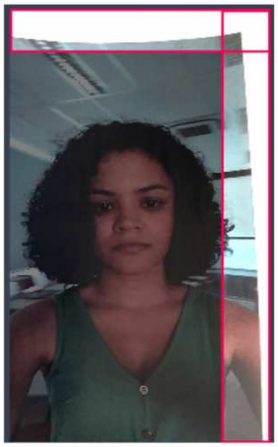
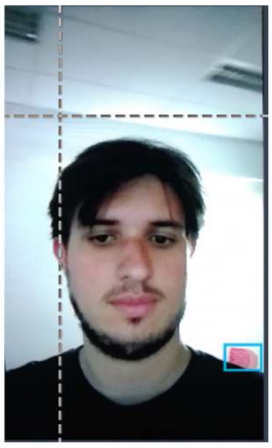
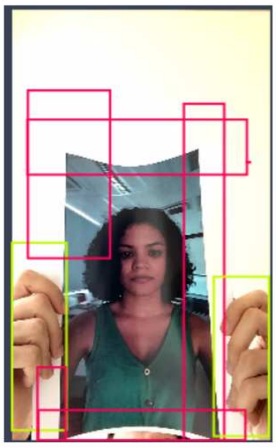
IMAGEM EXEMPLO	IMAGEM EXEMPLO	IMAGEM EXEMPLO	IMAGEM EXEMPLO
			
RÓTULO	RÓTULO	RÓTULO	RÓTULO
Border: bordas aparentes de uma imagem impressa ou display provenientes de um ataque.	Ghosting: efeito de rastro que ocorre em vídeos, devido a demora na mudança de cor dos pixels adjacentes, indicando tentativa de ataque.	Hand: mãos aparentes ao segurar o instrumento utilizado para o ataque, como imagens impressas ou displays.	Iconplay: ícones de play aparentes em quadros de vídeos utilizados em ataques.
IMAGEM EXEMPLO	IMAGEM EXEMPLO	IMAGEM EXEMPLO	
			
RÓTULO	RÓTULO	RÓTULO	
Lightingreflex: reflexo na superfície da imagem devido a exposição de luz, indicando um ataque.	Reflex: reflexo do rosto na imagem, dando a sensação de duplicação, em fraudes no formato de exibição.	Sensorreflex: "ondas" formadas ao capturar imagens de um display em ataques de exibição.	

Tabela 3 - Rótulos utilizados com seus respectivos exemplos

Fonte: Elaborada pelo autor

Um ponto importante a salientar na rotulação de classes como "sensorreflex", "lightiningreflex" é que levando em consideração a natureza da rede utilizada para detecção de objetos, foram escolhidos pontos da imagem onde esses efeitos seriam mais representativos e não necessariamente em todos os pontos em que aparecem na imagem.

4.5. Conjunto de dados para treinamento

Devido a limitações da plataforma de anotações que permite carregar no máximo 10.000 imagens, foram selecionadas para os experimentos 7.659 imagens do conjunto RECOD-MPAD em todos os diferentes cenários apresentados no conjunto. Dentre as classes selecionadas, a com mais instâncias detectadas foi a classe border, enquanto a com menos foi a classe ghosting, como mostrado em mais detalhes na figura 21:

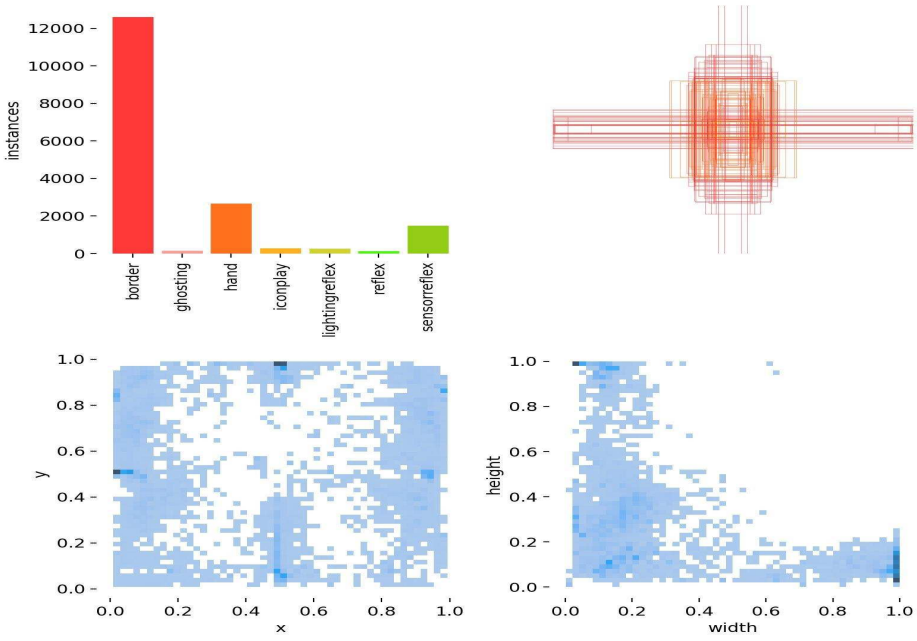


Figura 21 - Gráficos barras, distribuição, coordenadas e tamanhos dos rótulos e caixas delimitadoras

Fonte: Treinamento do trabalho

O gráfico de barras no canto superior esquerdo da imagem dá uma ideia clara de como as classes estão distribuídas, destacando uma disparidade entre a classe border comparada com as demais. O gráfico de distribuição no canto superior direito mostra a distribuição das caixas delimitadoras, onde observa-se uma concentração no centro da imagem, indicando a maior possibilidade de encontrar objetos nesta região. Já os dois mapas da região inferior representam as coordenadas e tamanhos normalizados das caixas delimitadoras, a maior concentração em valores menores (áreas mais escuras ou com maior densidade) no mapa *width x height* indicam uma maior parte das classes sendo objetos pequenos. .

4.6. Treinamento

O conjunto ficou separado em 6113 imagens no conjunto de treinamento, 780 para o conjunto de validação e 766 para o conjunto de testes.

Nessa etapa de treinamento, com objetivo de avaliar o desempenho padrão da arquitetura, o treinamento foi feito em condições padrão, sem aumento de dados, com uma taxa de aprendizagem de 0.01 e utilizando o modelo pré treinado YOLOv8n. Foram realizadas 100 épocas de treino com batches de tamanho 16 e tamanho das imagens 640x640.

4.7. Resultados do treino

Com o treinamento realizado na seção 4.6 foram obtidos os resultados abaixo de precisão e *recall*.

Conjunto de validação

Classe	Imagens	Instâncias	Precisão	Recall
Geral	760	2388	0,838	0,790
border	760	1632	0,962	0,888
ghosting	760	10	0,256	0,200
hand	760	293	0,993	0,915
iconplay	760	33	0,995	0,939
lightingreflex	760	28	0,887	0,841
reflex	760	6	0,885	1,000
sensorreflex	760	386	0,889	0,746

Tabela 4 - Resultados de precisão e *recalll* com o conjunto de validação

De acordo com os resultados apresentados pela tabela 4, o modelo mostrou um bom desempenho geral, destacando as classes "border", "hand", e "iconplay" com um desempenho excelente, passando de 0.95 em diversos resultados. Para classes como "sensorreflex", "lightiningreflex", verificou-se que em virtude da natureza da rede

utilizada, especializada em detecção, ela pode identificar as classes em posições diferentes das rotuladas ou até mesmo detectar em mais posições do que as originalmente rotuladas. Essas variações impactam sensivelmente os valores de precisão e recall, mesmo em situações onde a saída da rede é satisfatória. Outro ponto que pode ser destacado é o desempenho bem abaixo da classe ghosting, destoando dos outros resultados.

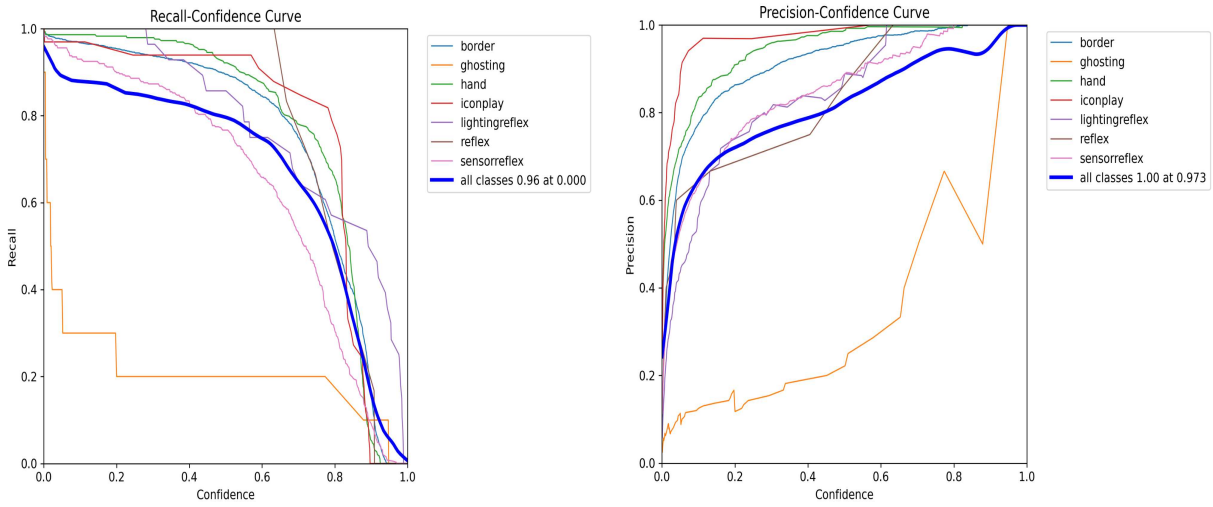


Figura 22 - gráficos de precisão e *recall* com o conjunto de validação

Nos gráficos de precisão e *recall* da figura 22 é possível observar o comportamento destas métricas em diferentes níveis de confiança. As curvas mostram um bom equilíbrio entre as métricas; especialmente nas classes "border",”hand" e "iconplay", mantendo alta precisão e recall em uma alta gama de valores de confiança. É também um destaque como a classe “reflex”, tem um bom recall inicialmente, mas tem uma queda brusca a partir de valores mais altos de confiança, além do desempenho bem abaixo da classe "ghosting" já observada na tabela anterior, e agora ilustrada mostrando um comportamento anômalo em relação às outras classes.

Conjunto de testes:

Classe	Imagens	instâncias	Precisão	Recall
Geral	752	2220	0,814	0,849
border	752	1624	0,907	0,948
ghosting	752	11	0,416	0,390
hand	752	338	0,967	0,953
iconplay	752	34	0,968	0,904
lightingreflex	752	33	0,727	0,879
reflex	752	7	0,888	1,000
sensorreflex	752	173	0,823	0,867

Tabela 5 - Resultados de precisão e *recall* com o conjunto de testes

Observando os resultados do conjunto de testes na tabela 5, nota-se um comportamento parecido com o visto anteriormente no conjunto de validação, o que indica um comportamento consistente do modelo. Houve uma ligeira queda na precisão, mas com um aumento do *recall*, indicando que o modelo identificou mais instâncias positivas, tanto verdadeiras quanto falsas. Houve um aumento significativo nos resultados das métricas relacionadas a classe "ghosting", mas com o desempenho ainda muito baixo.

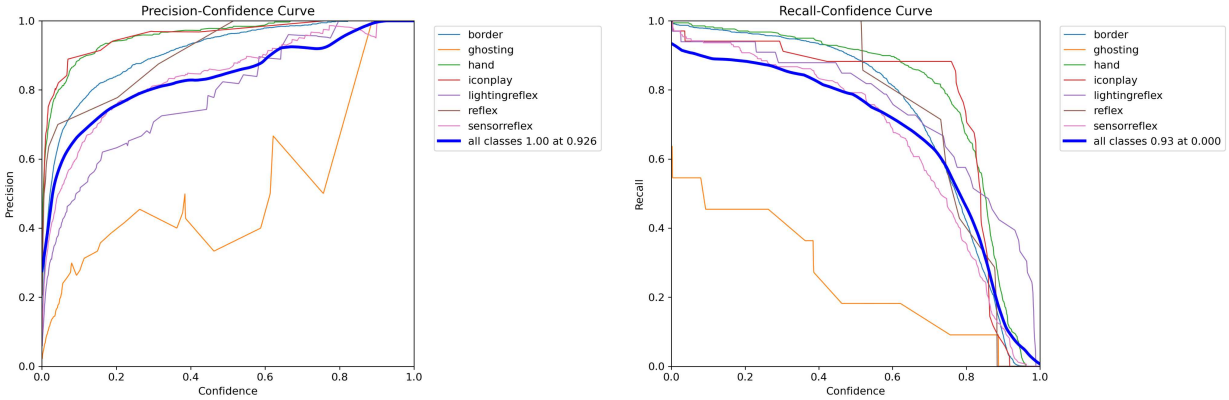


Figura 23 - gráficos de precisão e *recall* com o conjunto de testes

Os gráficos de precisão e *recall* da figura 23 mostram que no conjunto de testes houve um comportamento bem semelhante ao obtido no conjunto de validação. A confiança onde o conjunto alcança a precisão de 1.0 é um pouco mais baixa nesse conjunto, o que pode indicar uma diferença na variabilidade dos dados dos conjuntos de validação e testes, já que de acordo com as tabelas o valor da precisão seria ligeiramente menor no conjunto de testes. É possível notar também uma queda menos brusca nos valores de recall para a classe "reflex" em níveis de confiança mais altos.

4.8. Avaliação

No contexto da detecção de objetos, para a avaliação foi considerado uma apresentação genuína quando o modelo não encontra nenhum objeto que sugira uma tentativa de ataque. Por outro lado, uma apresentação falsa é identificada quando o sistema detecta um objeto que indica uma tentativa de fraude.

Primeiramente, foram utilizados 2496 frames com exemplos de ataques e 2304 apresentações genuínas, também retirados da base RECOD-MPAD, de 17 usuários

distintos, utilizando dados que não foram utilizados nos conjuntos de treinamento, validação e testes. Como resultado foi obtido um APCER de 6,13% e BPCER de 2,98% utilizando o valor de 0,25 de confiança com 153 apresentações falsas tratadas como genuínas e 69 apresentações genuínas tratadas como falsas.

4.9. Avaliação Cruzada

Para avaliação cruzada, foram utilizados 105 vídeos de exemplos de ataques e 70 vídeos de exemplos reais em formato mp4 e mov, retirados da base MSU-MFSD, também com o valor de 0,25 de confiança.

Como resultado, foram obtidos um APCER de 34,75% e um BPCER de 14,35%. Este desempenho ocorreu pois a maioria das classes onde o treinamento teve melhores resultados como "hand", "iconplay", "sensorreflex", não estão representativas na base MSU-MFSD. Para a classe "border", foi obtido um resultado melhor de instâncias detectadas, enquanto algumas instâncias da classe "ghosting" foram identificadas erroneamente como esperado devido ao baixo desempenho com esta classe no treinamento. Outro ponto que chamou atenção nos resultados foi em relação a classe "reflex" não ter identificado nenhum reflexo nesse experimento, mesmo obtendo valor de 1 no valor de *recall*, o que pode sinalizar *overfitting*.

5. COMPARAÇÃO ENTRE O MODELO PROPOSTO E OS TRABALHOS RELACIONADOS

A tabela 6 compara os valores de HTER do modelo deste trabalho, que utiliza as bases MSU-MFSD e RECOD-MPAD, aos resultados dos trabalhos relacionados.

Trabalho	Rede	Dataset	Acesso	Métrica	Resultados
Almeida 2018	Whole-face CNN	RECOD-MPAD	Criado para o trabalho	HTER	0,82%
	Patches CNN			HTER	1,14%
	Spoof-Loss CNN			HTER	0,63%
Rehman et al. 2018	LiveNet	CASIA-FASD	Público	HTER	19,12%
Satapathy et al. 2020	Xceptio-reduction	CASIA-FASD	Público	HTER	5,56%
Redmon et al. 2016	Fast R-CNN + YOLO	PASCAL VOC	Público	-	-
Anjith et al. 2020	MC - CNN	WMCA DATASET	Restrito	HTER	0,76%
Feng et al. 2016	SBIQF	REPLAY-ATTACK	Público	HTER	6,13%
Detecção de Objetos	YOLOv8	MSU-MFSD	Público	HTER	24,55%
	YOLOv8	RECOD-MPAD	Público	HTER	4,55%

Tabela 6 - Comparação entre o modelo proposto e os trabalhos relacionados.

Ao analisar os resultados da tabela observa-se um resultado competitivo na base de dados RECOD-MPAD, utilizando dados diferentes dos que foram utilizados para o treinamento do modelo. Apesar de ter apresentado um desempenho inferior comparado aos trabalhos relacionados que utilizam o mesmo dataset. Porém houve um desempenho significativamente inferior nos resultados em validação cruzada utilizando o MSU-MFSD, o que pode indicar um baixo poder de generalização do modelo.

6. CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho propôs desenvolver um modelo de aprendizado de máquina através de redes neurais para identificar vivacidade em imagens.

Inicialmente seriam utilizados algoritmos de classificação a fim de identificar essas fraudes. Posteriormente houve uma mudança de foco para a utilização de algoritmos de detecção de objetos, capazes de identificar e localizar algum objeto ou artefato na imagem que denuncie uma tentativa de fraude.

Utilizou-se a rede YOLOv8, para uma abordagem mais direta e de fácil implementação.

Nos experimentos realizados, o modelo apresentou bons resultados de treinamento no desempenho das classes "border", "hand" e "iconplay", com precisão e *recall* superiores a 0.95 a maioria do tempo. Contudo, houve desafios na detecção precisa das classes "sensorreflex" e "lightiningreflex", devido à natureza da rede em identificar objetos em posições variáveis, impactando os valores de precisão e *recall*. A classe "ghosting" apresentou desempenho inferior, destacando a necessidade de aprimoramentos futuros.

Na avaliação utilizou-se a base RECOD-MPAD, mesma do treinamento, porém com dados distintos. O modelo alcançou um HTER de 4,55%, enquanto na avaliação cruzada com vídeos da base MSU-MFSD, os resultados foram um HTER de 24,55%.

Houve uma grande diferença nas avaliações das duas bases de dados, o que indica baixo poder de generalização do modelo, porém a ausência de representatividade de algumas classes na base MSU-MFSD que obtiveram os melhores resultados no treinamento do modelo são um indicativo do baixo desempenho, já que mesmo em avaliação cruzada houve um desempenho mais satisfatório em encontrar classes como "border" por exemplo.

Esses resultados evidenciam tanto o potencial quanto às limitações da abordagem proposta, capaz de atingir bons resultados quando há um objeto indicativo de fraude fisicamente, como as mãos que seguram o objeto utilizado para o ataque ou bordas que ficam sobressalentes.

O desafio é maior ao tentar identificar artefatos na imagem, como reflexos, elementos gerados pela câmera e diferenças de iluminação, devido à natureza da rede. Porém há espaço para melhora aqui visto que a rede foi treinada com os valores padrões da YOLO.

Ajustes de hiperparâmetros e aumento dos dados de treinamento podem gerar resultados melhores, visto que o desempenho de precisão e recall da maioria das classes dessa categoria durante o treinamento foi satisfatório em alguns valores, tendo seu menor resultado de 0,725, ignorando os obtidos pela classe "ghosting" que destoam das demais.

Para trabalhos futuros, existem alternativas como aumentar e diversificar o conjunto de dados de treinamento, aumentando os exemplos de classes, principalmente as que representam artefatos na imagem e tiveram menor desempenho, coletando mais dados em diferentes cenários. Ajustes nos hiperparâmetros da YOLOv8n como taxa de aprendizado, confiança e número de épocas também podem trazer bons resultados, uma vez que os experimentos realizados utilizaram valores padrão, a fim de identificar o potencial do modelo. Aplicar técnicas de regularização como *dropout* e normalização do *batch* também são alternativas para melhorar a capacidade de generalização.

Além disso, avaliações cruzadas com conjuntos de dados diferentes adicionais pode fornecer uma visão mais abrangente do desempenho do modelo, uma vez que o RECOD-MPAD não favorece os pontos fortes que foram observados no treinamento e na avaliação. Estudar a viabilidade de integrar classificação para casos como reflexos de luz, e reflexos de sensor, que podem ter desempenho e avaliações afetados devido a natureza de detecção do modelo.

REFERÊNCIAS

LIVENESS. Disponível em: https://www.liveness.com/. Acesso em: jul. 2024.

RUSSEL, Stuart; NORVIG, Peter. Inteligência artificial. Rio de Janeiro: Elsevier, 2004.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. MIT Press, 2016.

RUSSELL, S.; NORVIG, P. Artificial Intelligence: A Modern Approach. 3. ed. Prentice Hall, 2010.

LUGER, G. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 2008.

RAMACHANDRA, R.; BUSCH, C. Presentation attack detection methods for face recognition systems: A comprehensive survey. ACM Computing Surveys (CSUR), v. 50, n. 1, p. 8, 2017. Disponível em: mai.2024 https://www.researchgate.net/publication/312937243_Presentation_Attack_Detection_Methods_for_Face_Recognition_Systems_-_A_Comprehensive_Survey. Acesso em: mai. 2024.

ALMEIDA, Waldir Rodrigues. Detecção de ataques de apresentação por faces em dispositivos móveis. Orientador: Anderson de Rezende Rocha. 2018. Dissertação (Mestrado) – Universidade Estadual de Campinas, Instituto de Computação, Campinas,

SP, 2018. Disponível em: https://repositorio.unicamp.br/acervo/detalhe/1082810. Acesso em: mar. 2024.

REHMAN, Yasar; MAN PO, Lai; LIU, Mengyang. Improving features generalization for face liveness detection using convolutional neural networks. ScienceDirect, 2018. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0957417418302811. Acesso em: mar. 2024.

SATAPATHY, Ashutosh; LIVINGSTON, L. M. Jenila. A lite convolutional neural network built on permuted Xception-inception and Xception-reduction modules for texture based facial liveness recognition. Springer, 2020. Disponível em: https://link.springer.com/article/10.1007/s11042-020-10181-4. Acesso em: jul. 2024.

ANJITH, George; ZOHRE, Mostaani; DAVID, Geissenbuhler; OLEGS, Nikisins; ANDRÉ, Anjos; SÉBASTIEN, Marcel. Biometric Face Presentation Attack Detection With Multi-Channel Convolutional Neural Network. IEEE, 2020. Disponível em: https://ieeexplore.ieee.org/abstract/document/8714076. Acesso em: jul. 2024.

LITONG, Feng; LAI-MAN, Po; YUMING, Li; XUYUAN, Xu; FANG, Yuan; TERENCE, Chun-Ho Cheung; KWOK-WAI, Cheung. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. ScienceDirect, 2016. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1047320316300244. Acesso em: jul. 2024.

OMAR, Luma, Qassam, Abedalqader. Face Liveness Detection under Processed Image Attacks. Durham theses, Durham University, 2018. Disponível em: http://etheses.dur.ac.uk/12812/. Acesso em: jul. 2024.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, 2012. p. 1097-1105. Disponível em: https://www.researchgate.net/publication/267960550_ImageNet_Classification_with_Deep_Convolutional_Neural_Networks. Acesso em: mar.2024.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. Nature, v. 521, n. 7553, p. 436-444, 2015. Disponível em: https://www.researchgate.net/publication/277411157_Deep_Learning. Acesso em: mar.2024.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition, 2014. Disponível em: https://arxiv.org/abs/1409.1556. Acesso em: mai. 2024.

TERVEN, Juan R.; CORDOVA-ESPARZA, Diana M. Publicado como um artigo de revista em Machine Learning and Knowledge Extraction. 4 de fevereiro de 2024. Instituto Politécnico Nacional CICATA-Qro; Universidad Autónoma de Querétaro Facultad de Informática.

BOULKENAFET, Zinelabidine; KOMULAINEN, Jukka; HADID, Abdenour. Face Spoofing Detection Using Colour Texture Analysis. IEEE Transactions on Information Forensics and Security, v. 11, n. 8, p. 1-1, ago. 2016. Disponível em: https://www.researchgate.net/publication/301571761_Face_Spoofing_Detection_Using_Colour_Texture_Analysis. Acesso em: jul. 2024.

ZOU, Z.; SHI, Z.; GUO, Y.; YE, J. Object Detection in 20 Years: A Survey. 2019. Disponível em: https://arxiv.org/abs/1905.05055. Acesso em: mar. 2024.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. University of Washington, Allen Institute for AI, Facebook AI Research, 2016. Disponível em: https://arxiv.org/abs/1506.02640. Acesso em: jul. 2024.

Detecção de Objetos em Imagens Para Verificação de Vivacidade

Pedro Henrique Azevedo[1]

[1]Departamento de Informática e Estatística

Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC - Brasil

Abstract. *This article describes the completion of an Information Systems undergraduate thesis that proposes a machine learning model based on object detection using the YOLOv8 network. The model aims to identify objects or artifacts in images that may indicate biometric fraud in the context of identity validation through facial recognition. Experiments were conducted using the MSU-MFSD and RECOD MPAD databases, and the results were evaluated. Despite challenges in identifying some classes and discrepancies in HTER results, the proposed object detection method showed potential in identifying certain types of artifacts that indicate biometric fraud. This approach could be utilized in future research.*

Resumo. *Este artigo descreve a realização de um trabalho de conclusão de curso de Sistemas de Informação que propõe um modelo de aprendizado de máquina, baseado em detecção de objetos, utilizando a rede YOLOv8 para identificar objetos ou artefatos em imagens que possam indicar uma fraude biométrica no contexto da validação de identidades através de reconhecimento facial. Foram realizados experimentos com as bases de dados MSU-MFSD e RECOD MPAD assim como avaliação dos resultados. Apesar dos desafios na identificação de algumas classes e dos resultados discrepantes de HTER, a proposta de detecção de objetos para identificar vivacidade mostrou potencial na identificação de alguns tipos de artefatos que denunciam uma fraude biométrica, podendo ser utilizada em pesquisas futuras.*

1. Introdução

Ferramentas de reconhecimento facial estão se tornando cada vez mais relevantes e necessárias para promover segurança e eficiência na validação de identidades, com aplicações em análise forense, controles de acesso e comércio eletrônico. No entanto, a aplicação generalizada desses sistemas também traz vulnerabilidades, pois artefatos faciais podem ser criados facilmente para fraudes (RAMACHANDRA; BUSCH, 2017). A verificação de vivacidade, que impede o uso de *deep fakes*, fotos ou outras falsificações, é essencial para garantir a presença de um ser humano real.

Técnicas para essa verificação incluem a análise de padrões em microtexturas e a exploração de informações temporais em vídeos. Além disso, métodos como a detecção de objetos podem identificar e localizar instâncias de objetos de interesse dentro de uma imagem ou vídeo, que podem ser indicativos de uma tentativa de fraude. Segundo Zou et al. (2019) a tarefa de detecção diz respeito a identificar e localizar instâncias de objetos de interesse dentro de uma imagem ou vídeo, a detecção não apenas identifica, como determina a posição de algum objeto por meio de caixas delimitadoras.

Analisando essas características com algoritmos de inteligência artificial é possível indicar automaticamente se o indivíduo está presente na biometria, ou se existe uma tentativa de fraude. O conceito de inteligência artificial é amplo e não possui uma definição exata, uma vez que a própria inteligência em si não possui um conceito totalmente definido, o termo pode ser conceituado de diversas formas. Segundo Luger (2008), a inteligência artificial pode ser definida como o ramo da ciência da computação

preocupada com a automação do comportamento inteligente. Dentro desse contexto, para que seja possível identificar a vivacidade em imagens a partir de uma solução computacional eficiente, este trabalho propõe uma solução utilizando redes neurais. Redes neurais são uma área da inteligência artificial, onde o estudo é focado em desenvolver técnicas para solução de problemas com base na estrutura dos neurônios humanos.

No trabalho de conclusão de curso é proposto um modelo de rede neural com aprendizagem supervisionada, que com uma base de dados consolidada para o treinamento da rede e testes de validação do modelo, se propõe uma solução para o problema da detecção de vivacidade em imagens, utilizando a abordagem de detecção de objetos.

1.1 Objetivos

O trabalho de conclusão de curso tem como finalidade desenvolver um modelo de aprendizado de máquina através de redes neurais, a fim de detectar objetos em imagens que sinalizem uma tentativa de ataque por fotografia ou vídeo, indicando uma fraude.

Além disso, o trabalho tem como objetivos específicos:

- Realizar pesquisa teórica sobre o estado da arte em machine learning com abordagem supervisionada, focado em detecção de objetos em imagens, buscando obter uma visão geral do que está sendo utilizado atualmente;
- Analisar algoritmos e metodologias de redes neurais, capazes de solucionar o problema proposto;

- Propor um modelo de rede neural para resolução do problema apresentado, a partir da análise feita no objetivo anterior;
- Realizar experimentos com a abordagem proposta;
- Avaliar os resultados obtidos.

2. Fundamentação Teórica

2.1 Biometria

O reconhecimento biométrico identifica exclusivamente uma pessoa através de traços biológicos distintivos como rosto, impressões digitais, retina, íris, geometria da mão, voz, DNA ou assinaturas manuscritas (OMAR et al., 2018). Em comparação com outros métodos biométricos, o reconhecimento facial utiliza dados facilmente acessíveis em domínio público, tornando-o vulnerável a ataques. Além disso, a biometria facial pode ser realizada de forma passiva, capturando dados de maneira direta e não intrusiva, como em câmeras de segurança.

2.2 Testes de Vivacidade

Testes de vivacidade utilizam algoritmos de classificação binária que fazem a distinção entre um indivíduo frente a uma câmera e algum tipo de fraude, que nos casos de detecção de face podem ser imagens, vídeos, ou até modelos em 3D (OMAR et al., 2018). Esses testes têm ganhado mais relevância com o crescimento de deep fakes e sistemas de biometria facial.

2.3 Redes Neurais

Segundo a neurociência, a atividade mental consiste principalmente em atividade eletroquímica nas redes de células cerebrais chamadas neurônios. Inspirados por essa hipótese, trabalhos no campo da inteligência artificial visam criar redes neurais artificiais (RUSSELL; NORVIG, 2016). O objetivo é simular através de camadas de nós, também chamados de neurônios, as conexões feitas no processo de aprendizagem orgânica, a fim de desenvolver sistemas inteligentes capazes de classificar e identificar padrões em diferentes conjuntos de dados.

2.4 Detecção de Objetos

Detecção de objetos diz respeito a uma tarefa de visão computacional de modelos de aprendizado profundo. Aprendizado profundo é um subcampo do aprendizado de máquina que utiliza redes neurais artificiais com muitas camadas. Segundo Goodfellow, Bengio e Courville (2016), o aprendizado profundo é particularmente eficaz devido à sua capacidade de lidar com grandes conjuntos de dados e modelar relações complexas. O objetivo principal da detecção de objetos é realizar a identificação e localizar as coordenadas das instâncias de objetos dentro de imagens ou vídeos.

3. Trabalhos Relacionados

Para os trabalhos relacionados foram selecionados 5 trabalhos que utilizam diferentes abordagens para testes de vivacidade, e também um trabalho que discute sobre a abordagem de detecção de objetos.

A tese de Almeida (2018), Detecção de Ataques de Apresentação por Faces em Dispositivos Móveis, aborda a detecção de ataques de apresentação em sistemas de biometria facial em dispositivos móveis, focando em ataques de fotos impressas e em tela. Foram propostas três metodologias baseadas em redes neurais convolucionais: (I) treinamento com imagens alinhadas de toda a face usando SqueezeNet, (II) treinamento com fragmentos de imagens para aumentar a robustez a desfoques e iluminação adversa, e (III) uma abordagem de perda multi-objetivo que diferencia amostras reais e de ataque em espaços de atributos intermediários. Para a avaliação, foram utilizados os *datasets* RECOD-MPAD e OULU-NPU. Resultados mostraram que os métodos II e III superaram as *baselines*, com destaque para a robustez e capacidade de generalização. No trabalho LiveNet: Improving Features Generalization for Face Liveness Detection Using Convolutional Neural Networks de Rehman et al. (2018) foi proposto uma abordagem com a rede LiveNet, que usa randomização contínua dos dados de treinamento para reduzir *overfitting*. Satapathy et al. (2020) em A Lite Convolutional Neural Network Built on Permuted Xception-Inception and Xception-Reduction Modules for Texture Based Facial Liveness Recognition, desenvolveram uma rede leve baseada nos módulos Xception-Inception e Xception-Reduction, focando em propriedades da pele e iluminação para diferenciar imagens genuínas de ataques. Anjith et al. (2020) propuseram uma abordagem baseada em rede neural convolucional multicanal para detecção de autenticidade no trabalho Biometric Face Presentation Attack Detection With Multi-Channel Convolutional Neural Network, aproveitando informações complementares de múltiplos canais e utilizando o modelo Light CNN pré-treinado em imagens faciais. Além disso,

uma abordagem *antispoofing* proposta por Feng et al. (2018) em Integration of image quality and motion cues for face anti-spoofing: A neural network approach integra múltiplas sugestões para combinar recursos de vivacidade de três aspectos: qualidade de imagem baseada em *shearlet*, movimento facial baseado em fluxo óptico e movimento de cena baseado em fluxo óptico, alimentando uma rede neural subsequente para detecção de vivacidade. Por fim, a abordagem YOLO de Redmon et al. (2016) foi destacada pela detecção de objetos em tempo real, sendo rápida e eficiente, mas com limitações na precisão de localização de objetos pequenos.

4. Desenvolvimento

Baseado na fundamentação teórica e nos trabalhos relacionados, este estudo opta por utilizar um algoritmo de detecção, especificamente a YOLOv8, para identificar objetos em imagens que indiquem tentativas de fraude. A YOLOv8, desenvolvida pela Ultralytics, é uma rede de visão computacional que suporta detecção, segmentação e classificação, tratando a detecção de objetos como um problema de regressão único, o que permite aprender características robustas e discriminativas diretamente dos dados (REDMON et al., 2016). Suas implementações de código aberto, alta precisão, eficiência e capacidade de detecção rápida tornam-na ideal para aplicações de segurança em tempo real.

4.1 Métricas

A avaliação será realizada utilizando três métricas principais que são comumente usadas na análise de desempenho de sistemas biométricos sendo elas:

- $APCER = \frac{Número\ de\ Apresentações\ de\ ataque\ Incorretamente\ Aceitas}{Número\ Total\ de\ Apresentações\ de\ Ataque}$
- $BPCER = \frac{Número\ de\ Apresentações\ Autênticas\ Incorretamente\ Rejeitadas}{Número\ Total\ de\ Apresentações\ de\ Autênticas}$
- $HTER = \frac{APCER + BPCER}{2}$

Para avaliação dos resultados do treinamento, foram utilizadas duas métricas adicionais:

- Precisão - mede o quão confiável é um modelo de detecção ao identificar um objeto, ou seja, mede a proporção de detecções corretas feitas pelo modelo, em relação ao total de predições. Pode ser calculado pela fórmula:
  - $Precisão = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Positivos}$
- Recall - mede a capacidade do modelo em identificar todas as instâncias positivas dos objetos, através da relação entre os verdadeiros positivos e o total de instâncias presentes nas imagens. Pode ser calculado pela fórmula:
  - $Recall = \frac{Verdadeiros\ Positivos}{Verdadeiros\ Positivos + Falsos\ Negativos}$

4.2 *Datasets* utilizados

4.2.1 - RECOD-MPAD

O *dataset* utiliza de dois dispositivos com câmeras frontais diferentes para fazer a captura: o *smartphone* Moto G5 lançado em 2017 e o *smartphone* Moto X Style XT1572 de finais de 2015, em cinco cenários de iluminação diferentes que englobam sessões ao ar livre, com luz solar direta ou sombra, e em ambientes internos com iluminação artificial ou natural e também com luzes apagadas, como demonstrado na figura 1. Os ataques estão divididos entre ataques de exibição (*displays*) e ataques de

foto impressa. O *dataset* é construído com 64 quadros igualmente espaçados de cada vídeo de resolução 1920x1080 gerando um número final de 142.997 quadros, contendo 45 usuários diferentes.

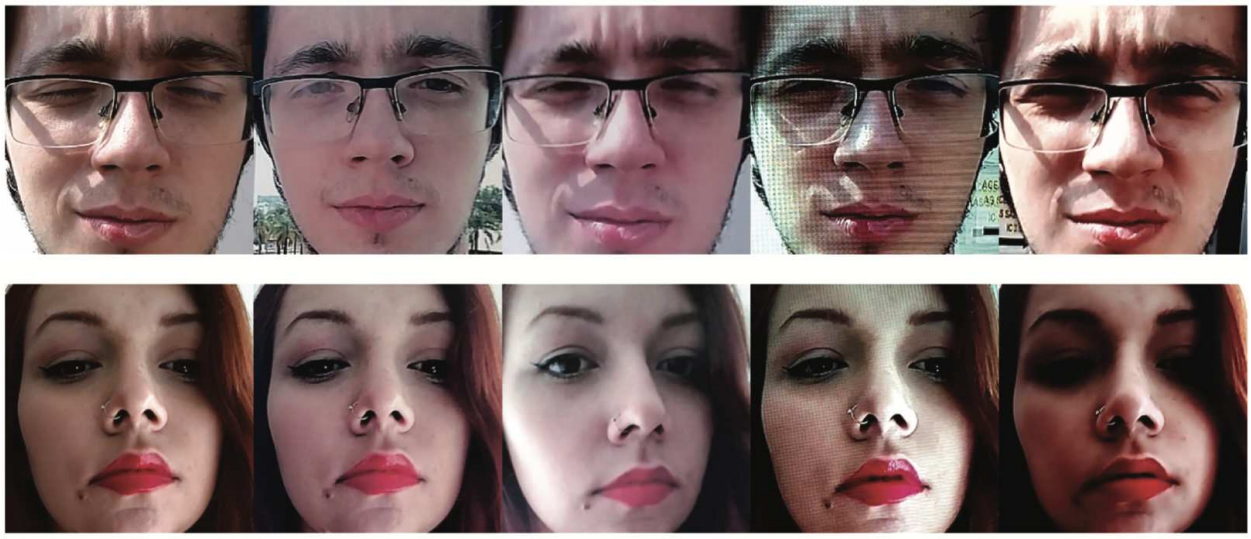


Figura 1 - Exemplo de ataques nos diferentes ambientes no RECOD-MPAD. Da esquerda para a direita: real, ataque impresso 1, ataque impresso 2, ataque com display 1, ataque com display 2.

Fonte: Waldir Rodrigues de Almeida, 2018

4.2.2 - MSU-MFSD

O conjunto MSU-MFSD contém 280 gravações de vídeos, entre ataques e vídeos genuínos, feitos com dois tipos de câmeras em diferentes resoluções (720x480 e 640x480) com 35 indivíduos diferentes. Foram produzidos ataques de *display* a partir da tela de um iPad Air e de um iPhone, além de ataques impressos dos 35 indivíduos em papéis A3 utilizando uma impressora colorida. Como demonstrado na figura 2.

Figura 2 - Exemplos de amostras do MSU-MFSD. A primeira linha corresponde a imagens tiradas com um telefone Android, enquanto a segunda linha mostra imagens capturadas com uma câmera de laptop. Da esquerda para a direita: rostos reais e os respectivos ataques de iPad, iPhone e impressão.

Fonte: Boulkenafet et al., 2016

4.3 Rotulação

Para fazer a rotulação foi utilizado o Roboflow, que facilita esse processo fornecendo as ferramentas para delimitar os objetos nas imagens, além de recursos para acelerar o processo. Também foi possível fazer o *download* dos dados anotados já no formato compatível com a YOLOv8. As classes definidas são explicadas na tabela 1.

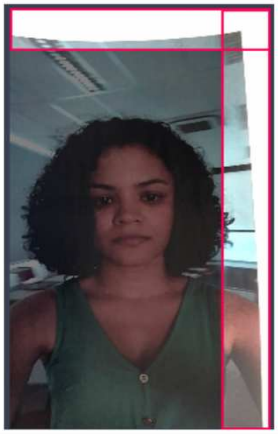
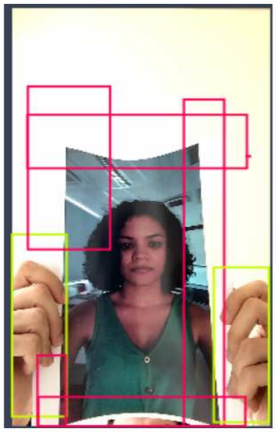
IMAGEM EXEMPLO	IMAGEM EXEMPLO	IMAGEM EXEMPLO	IMAGEM EXEMPLO
			
RÓTULO	RÓTULO	RÓTULO	RÓTULO
Border: bordas aparentes de uma imagem impressa ou display provenientes de um ataque.	Ghosting: efeito de rastro que ocorre em vídeos, devido a demora na mudança de cor dos pixels adjacentes, indicando tentativa de ataque.	Hand: mãos aparentes ao segurar o instrumento utilizado para o ataque, como imagens impressas ou displays.	Iconplay: ícones de play aparentes em quadros de vídeos utilizados em ataques.
IMAGEM EXEMPLO	IMAGEM EXEMPLO	IMAGEM EXEMPLO	
			
RÓTULO	RÓTULO	RÓTULO	
Lightingreflex: reflexo na superfície da imagem devido a exposição de luz, indicando um ataque.	Reflex: reflexo do rosto na imagem, dando a sensação de duplicação, em fraudes no formato de exibição.	Sensorreflex: "ondas" formadas ao capturar imagens de um display em ataques de exibição.	

Tabela 1 - Rótulos utilizados com seus respectivos exemplos

Fonte: Elaborada pelo autor

Um ponto importante a salientar na rotulação de classes como "sensorreflex", "lightiningreflex" é que levando em consideração a natureza da rede utilizada para detecção de objetos, foram escolhidos pontos da imagem onde esses efeitos seriam mais representativos e não necessariamente em todos os pontos em que aparecem na imagem.

4.4 Conjunto de dados para treinamento

Devido a limitações da plataforma de anotações que permite carregar no máximo 10.000 imagens, foram selecionadas para os experimentos 7.659 imagens do conjunto RECOD-MPAD em todos os diferentes cenários apresentados no conjunto. O conjunto ficou separado em 6113 imagens no conjunto de treinamento, 780 para o conjunto de validação e 766 para o conjunto de testes.

Nessa etapa de treinamento, com objetivo de avaliar o desempenho padrão da arquitetura, o treinamento foi feito em condições padrão, sem aumento de dados, com uma taxa de aprendizagem de 0,01 e utilizando o modelo pré treinado YOLOv8. Foram realizadas 100 épocas de treino com batches de tamanho 16 e tamanho das imagens 640x640.

4.5 Resultados do treino

Com o treinamento realizado na seção 4.6 foram obtidos os resultados da tabela 2 nas métricas de precisão e *recall*.

Conjunto de validação:

Classe	Imagens	Instâncias	Precisão	Recall
Geral	760	2388	0,838	0,790
border	760	1632	0,962	0,888
ghosting	760	10	0,256	0,200
hand	760	293	0,993	0,915
iconplay	760	33	0,995	0,939
lightingreflex	760	28	0,887	0,841
reflex	760	6	0,885	1,000
sensorreflex	760	386	0,889	0,746

Tabela 2 - Resultados de precisão e *recall* com o conjunto de validação

De acordo com os resultados apresentados pela tabela 2, o modelo mostrou um bom desempenho geral, destacando as classes "border", "hand", e "iconplay" com um desempenho excelente, passando de 0.95 em diversos resultados. Para classes como "sensorreflex", "lightiningreflex", verificou-se que em virtude da natureza da rede utilizada, especializada em detecção, ela pode identificar as classes em posições diferentes das rotuladas ou até mesmo detectar em mais posições do que as originalmente rotuladas. Essas variações impactam sensivelmente os valores de

precisão e *recall*, mesmo em situações onde a saída da rede é satisfatória. Outro ponto que pode ser destacado é o desempenho bem abaixo da classe "ghosting", destoando dos outros resultados.

Conjunto de testes:

Classe	Imagens	instâncias	Precisão	Recall
Geral	752	2220	0,814	0,849
border	752	1624	0,907	0,948
ghosting	752	11	0,416	0,390
hand	752	338	0,967	0,953
iconplay	752	34	0,968	0,904
lightingreflex	752	33	0,727	0,879
reflex	752	7	0,888	1,000
sensorreflex	752	173	0,823	0,867

Tabela 3 - Resultados de precisão e *recall* com o conjunto de testes

Observando os resultados do conjunto de testes na tabela 3, nota-se um comportamento parecido com o visto anteriormente no conjunto de validação, o que indica um comportamento consistente do modelo. Houve uma ligeira queda na precisão, mas com um aumento do recall, indicando que o modelo identificou mais instâncias positivas, tanto verdadeiras quanto falsas. Houve um aumento significativo

nos resultados das métricas relacionadas a classe "ghosting", mas com o desempenho ainda muito baixo.

4.6 Avaliação

No contexto da detecção de objetos, para a avaliação foi considerado uma apresentação genuína quando o modelo não encontra nenhum objeto que sugira uma tentativa de ataque. Por outro lado, uma apresentação falsa é identificada quando o sistema detecta um objeto que indica uma tentativa de fraude.

Primeiramente, foram utilizados 2496 *frames* com exemplos de ataques e 2304 apresentações genuínas, também retirados da base RECOD-MPAD, de 17 usuários distintos, utilizando dados que não foram utilizados nos conjuntos de treinamento, validação e testes. Como resultado foi obtido um APCER de 6,13% e BPCER de 2,98% utilizando o valor de 0,25 de confiança com 153 apresentações falsas tratadas como genuínas e 69 apresentações genuínas tratadas como falsas.

4.7 Avaliação cruzada

Para avaliação cruzada, foram utilizados 105 vídeos de exemplos de ataques e 70 vídeos de exemplos reais em formato mp4 e mov, retirados da base MSU-MFSD, também com o valor de 0,25 de confiança.

Como resultado, foram obtidos um APCER de 34,75% e um BPCER de 14,35%. Este desempenho ocorreu pois a maioria das classes onde o treinamento teve melhores resultados como "hand", "iconplay", "sensorreflex", não estão representativas na base MSU-MFSD. Para a classe "border", foi obtido um resultado melhor de

instâncias detectadas, enquanto algumas instâncias da classe "ghosting" foram identificadas erroneamente como esperado devido ao baixo desempenho com esta classe no treinamento. Outro ponto que chamou atenção nos resultados foi em relação a classe "reflex" não ter identificado nenhum reflexo nesse experimento, mesmo obtendo valor de 1 no valor de *recall*, o que pode sinalizar *overfitting*.

5 - COMPARAÇÃO ENTRE O MODELO PROPOSTO E OS TRABALHOS RELACIONADOS

A tabela 4 abaixo compara os valores de HTER do modelo deste trabalho, que utiliza as bases MSU-MFSD e RECOD-MPAD, aos resultados dos trabalhos relacionados.

Trabalho	Rede	Dataset	Acesso	Métrica	Resultados
Almeida 2018	Whole-face CNN	RECOD-MPAD	Criado para o trabalho	HTER	0,82%
	Patches CNN			HTER	1,14%
	Spoof-Loss CNN			HTER	0,63%
Rehman et al. 2018	LiveNet	CASIA-FASD	Público	HTER	19,12%
Satapathy et al. 2020	Xceptio-reduction	CASIA-FASD	Público	HTER	5,56%
Redmon et al. 2016	Fast R-CNN + YOLO	PASCAL VOC	Público	-	-
Anjith et al. 2020	MC - CNN	WMCA DATASET	Restrito	HTER	0,76%
Feng et al. 2016	SBIQF	REPLAY-ATTACK	Público	HTER	6,13%
Detecção de Objetos	YOLOv8	MSU-MFSD	Público	HTER	24,55%
	YOLOv8	RECOD-MPAD	Público	HTER	4,55%

Tabela 4 - Comparação entre o modelo proposto e os trabalhos relacionados.

Ao analisar os resultados da tabela observa-se um resultado competitivo na base de dados RECOD-MPAD, utilizando dados diferentes dos que foram utilizados para o treinamento do modelo. Apesar de ter apresentado um desempenho inferior comparado aos trabalhos relacionados que utilizam o mesmo *dataset.* Porém houve um desempenho significativamente inferior nos resultados em validação cruzada utilizando o MSU-MFSD, o que pode indicar um baixo poder de generalização do modelo.

6. Conclusão e Trabalhos Futuros

O trabalho de conclusão de curso propôs desenvolver um modelo de aprendizado de máquina com redes neurais para identificar vivacidade em imagens, inicialmente seriam utilizados algoritmos de classificação posteriormente houve uma mudança para algoritmos de detecção de objetos, como a rede YOLOv8, pela sua abordagem direta e fácil implementação. Nos experimentos, o modelo mostrou bons resultados nas classes "border", "hand" e "iconplay", com precisão e *recall* superiores a 0,95. No entanto, teve dificuldades nas classes "sensorreflex" e "lightiningreflex" devido à variabilidade das posições dos objetos, e a classe "ghosting" apresentou desempenho inferior, indicando necessidade de melhorias.

Na avaliação com a base RECOD-MPAD, o modelo obteve um HTER de 4,55%, enquanto na avaliação cruzada com a base MSU-MFSD, alcançou um HTER 24,55%, revelando um baixo poder de generalização, e desafios na detecção com a base MSU-MFSD que tem baixa representatividade nas classes onde o modelo se saiu melhor na etapa de treinamento. Esses resultados evidenciam o potencial da abordagem para detectar fraudes físicas, como mãos segurando objetos ou bordas

salientes, mas destacam desafios na identificação de artefatos na imagem, como reflexos e diferenças de iluminação. Para futuros trabalhos, sugere-se aumentar e diversificar o conjunto de dados de treinamento, ajustar hiperparâmetros e aplicar técnicas de regularização para melhorar a capacidade de generalização do modelo. Avaliações cruzadas adicionais e a integração de classificação para os casos que obtiveram pior desempenho podem também aprimorar os resultados.

7. Referências

RAMACHANDRA, R.; BUSCH, C. Presentation attack detection methods for face recognition systems: A comprehensive survey. ACM Computing Surveys (CSUR), v. 50, n. 1, p. 8, 2017. Disponível em: mai.2024https://www.researchgate.net/publication/312937243_Presentation_Attack_Detection_M ethods_for_Face_Recognition_Systems_-_A_Comprehensive_Survey. Acesso em: mai. 2024.

ZOU, Z.; SHI, Z.; GUO, Y.; YE, J. Object Detection in 20 Years: A Survey. 2019. Disponível em: https://arxiv.org/abs/1905.05055. Acesso em: mar. 2024.

LUGER, G. Artificial Intelligence: Structures and Strategies for Complex Problem Solving. 2008.

OMAR, Luma, Qassam, Abedalqader. Face Liveness Detection under Processed Image Attacks. Durham theses, Durham University, 2018. Disponível em: http://etheses.dur.ac.uk/12812/. Acesso em: jul. 2024.

RUSSEL, Stuart; NORVIG, Peter. Inteligência artificial. Rio de Janeiro: Elsevier, 2004.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. MIT Press, 2016.

ALMEIDA, Waldir Rodrigues. Detecção de ataques de apresentação por faces em dispositivos móveis. Orientador: Anderson de Rezende Rocha. 2018. Dissertação (Mestrado) – Universidade Estadual de Campinas, Instituto de Computação, Campinas, SP, 2018. Disponível em: https://repositorio.unicamp.br/acervo/detalhe/1082810. Acesso em: mar. 2024

REHMAN, Yasar; MAN PO, Lai; LIU, Mengyang. Improving features generalization for face liveness detection using convolutional neural networks. ScienceDirect, 2018. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0957417418302811. Acesso em: mar. 2024.

SATAPATHY, Ashutosh; LIVINGSTON, L. M. Jenila. A lite convolutional neural network built on permuted Xception-inception and Xception-reduction modules for texture based facial liveness recognition. Springer, 2020. Disponível em: https://link.springer.com/article/10.1007/s11042-020-10181-4. Acesso em: jul. 2024.

ANJITH, George; ZOHRE, Mostaani; DAVID, Geissenbuhler; OLEGS, Nikisins; ANDRÉ, Anjos; SÉBASTIEN, Marcel. Biometric Face Presentation Attack Detection With Multi-Channel Convolutional Neural Network. IEEE, 2020. Disponível em: https://ieeexplore.ieee.org/abstract/document/8714076. Acesso em: jul. 2024.

LITONG, Feng; LAI-MAN, Po; YUMING, Li; XUYUAN, Xu; FANG, Yuan; TERENCE, Chun-Ho Cheung; KWOK-WAI, Cheung. Integration of image quality and motion cues for face anti-spoofing: A neural network approach. ScienceDirect, 2016. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S1047320316300244. Acesso em: jul. 2024.

REDMON, J.; DIVVALA, S.; GIRSHICK, R.; FARHADI, A. You Only Look Once: Unified, Real-Time Object Detection. University of Washington, Allen Institute for AI, Facebook AI Research, 2016. Disponível em: https://arxiv.org/abs/1506.02640. Acesso em: jul. 2024.