Marcos Henrique Fernandes Marcone

# Análise da Efetividade de um Sistema Anti-Spoofing para Reconhecimento Facial com a Classificação de Imagens Estéreo através de uma Rede Neural Convolucional

Natal – RN

Julho de 2023

Marcos Henrique Fernandes Marcone

# Análise da Efetividade de um Sistema Anti-Spoofing para Reconhecimento Facial com a Classificação de Imagens Estéreo através de uma Rede Neural Convolucional

Trabalho de Conclusão de Curso na modalidade Monografia, submetido como parte dos requisitos necessários para conclusão do curso de Engenharia de Computação pela Universidade Federal do Rio Grande do Norte (UFRN/CT)

Orientador: Dr. Bruno Marques Ferreira da Silva

Universidade Federal do Rio Grande do Norte – UFRN

Departamento de Engenharia de Computação e Automação – DCA

Curso de Engenharia de Computação

Natal – RN

Julho de 2023

Universidade Federal do Rio Grande do Norte - UFRN
Sistema de Bibliotecas - SISBI
Catalogação de Publicação na Fonte. UFRN - Biblioteca Central Zila Mamede

Marcone, Marcos Henrique Fernandes.
Análise da efetividade de um sistema anti-spoofing para reconhecimento facial com a classificação de imagens estéreo através de uma rede neural convolucional / Marcos Henrique Fernandes Marcone. - 2023.
42f.: il.

Monografia (Graduação) - Universidade Federal do Rio Grande do Norte, Centro de Tecnologia, Graduação em Engenharia de Computação, Natal, 2023.
Orientador: Dr. Bruno Marques Ferreira da Silva.

1. Face Anti-Spoofing - Monografia. 2. Visão Estéreo - Monografia. 3. Convolutional Neural Network - Monografia. 4. Transfer Learning - Monografia. 5. Ataques de Apresentação Facial - Monografia. I. Silva, Bruno Marques Ferreira da. II. Título.

RN/UF/BCZM CDU 004

Elaborado por Raimundo Muniz de Oliveira - CRB-15/429

Marcos Henrique Fernandes Marcone

# Análise da Efetividade de um Sistema Anti-Spoofing para Reconhecimento Facial com a Classificação de Imagens Estéreo através de uma Rede Neural Convolucional

Trabalho de Conclusão de Curso na modalidade Monografia, submetido como parte dos requisitos necessários para conclusão do curso de Engenharia de Computação pela Universidade Federal do Rio Grande do Norte (UFRN/CT)

Orientador: Dr. Bruno Marques Ferreira da Silva

Trabalho aprovado. Natal – RN, 07 de Julho de 2023:

---

**Prof. Dr. Bruno Marques Ferreira da Silva - Orientador**
UFRN

---

**Prof.ª Dra. Fabiana Tristão de Santana - Examinadora Interna**
UFRN

---

**Prof. Dr. Helton Maia Peixoto - Examinador Interno**
UFRN

Natal – RN
Julho de 2023

# RESUMO

O aumento no uso de tecnologias de reconhecimento facial para autenticação e segurança tem acelerado a necessidade de desenvolver métodos eficazes para combater a falsificação facial, conhecida como Ataques de Apresentação Facial. Este trabalho aborda o desafio do *Face Anti-Spoofing* (FAS) e propõe uma abordagem baseada na visão estéreo para melhorar a detecção de ataques de *spoofing* facial 2D. A pesquisa detalha a implementação de uma *Convolutional Neural Network* (CNN) construída a partir da técnica de *Transfer Learning*, que usa imagens de disparidade para adicionar uma dimensão de profundidade à análise. Ao longo do trabalho também comparamos o desempenho do modelo proposto com um modelo de código aberto existente, o *Silent-Face-Anti-Spoofing*, e oferecemos insights sobre possíveis melhorias. Os resultados mostraram que a exploração de imagens de disparidade pode aumentar a robustez e a generalização em relação às abordagens tradicionais baseadas apenas em imagens RGB, atingindo uma acurácia de 100% em um ambiente controlado.

**Palavras-chaves**: *Face Anti-Spoofing*; Visão Estéreo; Convolutional Neural Network; Transfer Learning; Ataques de Apresentação Facial.

# ABSTRACT

The rise in the use of facial recognition technologies for authentication and security has hastened the need to develop effective methods to combat facial spoofing, known as face Presentation Attacks (PAs). This work addresses the challenge of Face Anti-Spoofing (FAS) and proposes a stereo vision-based approach to enhance the detection of 2D facial spoofing attacks. The research details the implementation of a Convolutional Neural Network (CNN) built from the Transfer Learning technique, which uses disparity images to add a depth dimension to the analysis. Throughout the work, the performance of the proposed model is also compared with an existing open-source model, the Silent-Face-Anti-Spoofing, providing insights into potential improvements. The results show that the exploration of disparity images can increase robustness and generalization compared to traditional approaches based solely on RGB images, achieving an accuracy of 100% in a controlled environment.

**Keywords**: Face Anti-Spoofing; Stereo Vision; Convolutional Neural Network; Transfer Learning; Face Presentation Attacks.

# LISTA DE ILUSTRAÇÕES

# LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| FAS | *Face Anti-Spoofing* |
| PA | *Presentation Attack* |
| IA | *Inteligência Artificial* |
| ML | *Machine Learning* |
| DL | *Deep Learning* |
| CNN | *Convolutional Neural Network* |
| FPS | *Frame Per Second* |
| VPU | *Visual Processing Unit* |
| API | *Application Programming Interface* |

# SUMÁRIO

# 1 INTRODUÇÃO

Visão computacional, de acordo com Brown e Ballard (1982), é a ciência responsável pela visão de uma máquina englobando a forma como os computadores interpretam o ambiente circundante. Por meio da extração de informações relevantes de imagens captadas por câmeras de vídeo, sensores e outros dispositivos, o computador pode reconhecer, manipular e refletir sobre os objetos presentes na imagem.

Os sistemas biométricos se beneficiam consideravelmente das técnicas de visão computacional, com o reconhecimento facial se destacando como um método importante dessa interação. Em tal sistema, é aplicado um conjunto de algoritmos de visão computacional para identificar e reconhecer características faciais. A operação usual desses sistemas implica na detecção de uma face em uma imagem, seguida pela normalização dessa face para ajustar as variações de iluminação e pose. Posteriormente, ocorre a extração de atributos da face e, finalmente, é feita a comparação desses atributos com um repositório de dados para identificar uma correspondência (THORAT; NAYAK; DANDALE, 2010).

Nas últimas décadas, as técnicas de reconhecimento facial receberam uma atenção significativa das pesquisas dedicadas à biometria. Juntamente com outras formas de reconhecimento biométrico amplamente difundidas - como impressões digitais e identificação pela íris - ela vem sendo empregada, principalmente, para autenticação e segurança. A sua utilização em uma variedade de aplicações eletrônicas deve-se à sua conveniência e interação intuitiva com o usuário (GALBALLY; MARCEL; FIERREZ, 2014).

Na ausência de uma funcionalidade *Face Anti-Spoofing* (FAS), os sistemas de reconhecimento facial se tornam suscetíveis a uma variedade de ataques de falsificação facial, os quais são conhecidos como *Face Presentation Attacks* (PAs) ou Ataques de Apresentação Facial. Esses ataques podem ser facilmente gerados para obter acesso ilegal a dispositivos do usuário, como celular, computador ou ativos intangíveis, como contas bancárias (REHMAN; PO; LIU, 2020).

De acordo com Rehman, Po e Liu (2020), os principais tipos de ataques de *face spoofing* são: fotos; replay de videos e máscaras de rostos. Os autores ressaltam que embora os ataques com máscaras não sejam comumente acessíveis devido ao custo de produção mais elevado, as outras duas técnicas, que usam fotos impressas e vídeos reproduzidos, podem ser gerados com facilidade utilizando recursos de fotografia ou filmagem de alta resolução.

A obtenção de uma fotografia ou um vídeo do rosto de uma pessoa para gerar um Ataque de Apresentação Facial tornou-se uma tarefa mais simples. Principalmente, considerando a disponibilidade de equipamentos avançados de câmeras e impressoras, e o acesso facilitado a plataformas de mídia social como Twitter, Facebook e Instagram.

Assim como ressalta Yu et al. (2022), ao contrário de outras tarefas da área de visão computacional, o Sistema de *Face Anti-Spoofing* apresenta uma natureza auto-evolutiva. Isto é, a dinâmica de ataque *versus* defesa evolui de maneira iterativa, tornando-o um desafio adicional.

As principais pesquisas desenvolvidas nessa área tem como principal foco mono câmeras RGB comerciais (YU et al., 2022). Entretanto, como pontua Wu et al. (2020), a maioria desses métodos se mostram instáveis quando lidam com imagens que carecem de contexto, nas quais elementos planos, tais como telas em alta definição e impressões em papel, ocupam todo o campo de visão. Durante o processo de inferência, os modelos se limitam a entregar um resultado binário, não sendo capazes de prover explicações ou justificativas para as decisões tomadas.

Dessa forma, cresce cada vez mais o interesse por métodos *anti-spoofing* que utilizam informações adicionais além da imagem RGB. Um exemplo é a Visão Estéreo, que consiste em capturar uma cena a partir de duas câmeras e através de propriedades e manipulações matemáticas é possível obter a profundidade dos elementos em relação à câmera (HAMZAH; IBRAHIM, 2016).

Nesse contexto, o objetivo deste trabalho é desenvolver e avaliar uma *Convolutional Neural Network* (CNN), construída a partir da técnica de *Transer Learning*, para a detecção de ataques de *spoofing* facial 2D. Em particular, a abordagem apresentada difere das técnicas tradicionais ao aplicar a visão estéreo para capturar imagens de disparidade, adicionando assim uma dimensão de profundidade à análise e potencialmente aumentando a eficácia na detecção de *spoofing*.

O trabalho também busca comparar a eficácia do modelo proposto com a de uma rede neural de código aberto existente, o *Silent-Face-Anti-Spoofing* (MINIVISION, 2020), a fim de contextualizar a performance do modelo proposto e explorar possíveis melhorias. A validação e a comparação dos modelos são realizadas em um conjunto de dados composto por imagens de rostos reais e de spoofing facial.

Os resultados obtidos demonstraram que a abordagem proposta pôde lidar efetivamente com o *spoofing* facial 2D em um ambiente controlado e a uma distância específica, alcançando uma acurácia de 100%. Além disso, os resultados sugeriram que a exploração de imagens de disparidade pôde oferecer benefícios substanciais em termos de robustez e generalização em relação às abordagens tradicionais baseadas em imagens RGB.

Este artigo está estruturado da seguinte maneira: o capítulo 2 fornece uma base sólida sobre os conceitos centrais de detecção de *spoofing facial*, redes neurais convolucionais e visão estéreo. Em seguida, capítulo 3 oferece uma visão geral sobre os trabalhos desenvolvidos para *Face Anti-Spoofing*. O capítulo 4 detalha o design e a implementação da CNN proposta, assim como o processo de construção do conjunto de dados utilizados. Já o capítulo 5 apresenta os resultados do treinamento, validação e teste da CNN, além de comparar o desempenho do nosso modelo com o do Silent-Face-Anti-Spoofing. Por fim, o capítulo 6 sintetiza as principais

conclusões obtidas, além de propor direções para pesquisas futuras na detecção de spoofing facial utilizando imagens de disparidade.

# 2 REFERENCIAL TEÓRICO

Nesta seção são apresentados os principais conceitos teóricos que foram utilizados para o desenvolvimento do trabalho. Inicia-se com explicação do que se trata *Face Anti-Spoofing*, Ataques de Apresentação Facil e seus diferentes tipos. Em seguida, são abordadas as definições que envolvem a visão estéreo e como é possível obter a disparidade e a profundidade a partir de uma câmera estéro. Por fim, são explicadas a teoria e as concepções sobre *Deep Learning*, Redes Neurias Convolucionais e *Transfer Learning*.

## 2.1 *Face anti-spoofing*

Sistemas de reconhecimento facial são aplicações feitas para identificar automaticamente uma pessoa a partir de uma imagem digital ou de um *frame* de um vídeo. Comumente, as técnicas mais utilizadas comparam as características faciais específicas de uma imagem com um *dataset* facial, o qual mantém essas características armazenadas. Tais sistemas são tipicamente empregados na autenticação de usuários para garantir a segurança no acesso a serviços como: pagamentos online; batidas de ponto; desbloqueio de dispositivos; entre outras aplicações (THORAT; NAYAK; DANDALE, 2010).

Entretanto, esses sistemas estão vulneráveis a ataques de *spoofing*, os quais são conhecidos na literatura por *face Presentation Attacks* (PAs), que pode ser traduzido como Ataques de Apresentação facial. Esses ataques podem ser provenientes de imagens, vídeos, maquiagem, máscaras 3D, etc e são amplamente utilizados para ganhar acesso ilegal/não autorizado a serviços específicos de um determinado usuário, como por exemplo a contas bancárias (REHMAN; PO; LIU, 2020).

Assim, as tecnologias de *Face Anti-Spoofing* (FAS), também conhecidas por *face liveness detection*, fornecem suporte aos sistemas de reconhecimento facial na prevenção dos ataques de apresentação facial (REHMAN; PO; LIU, 2020) e são um componente crucial na segurança desses sistemas, pois cumpre justamente a tarefa de prevenir que uma verificação facial fraudulenta seja feita (YU et al., 2022).

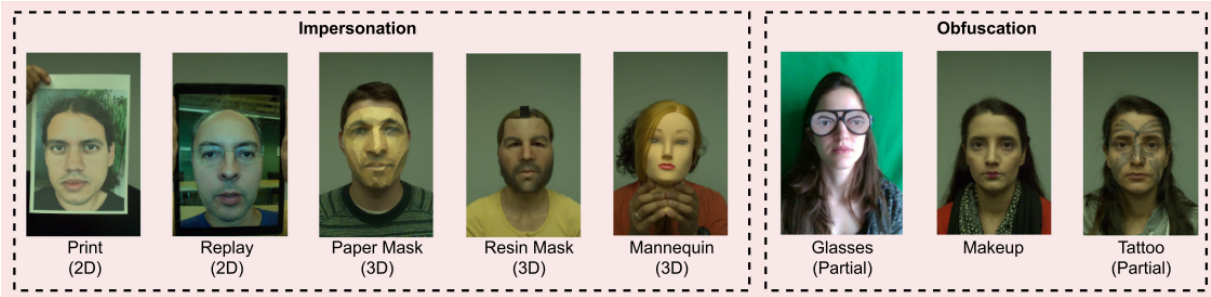
Os principais tipos de ataques de *face spoofing* são: fotos; replay de videos e máscaras de rostos. De acordo com Yu et al. (2022), PAs faciais podem ser classificados através de duas características: a intenção do atacante e a propriedade geométrica do ataque. A Figura 1 demonstra exemplos de diferentes tipos de ataques e suas respectivas classificações.

Em relação à intenção do atacante, existem dois casos típicos. O primeiro é a *personificação*, que envolve o uso de falsificação para ser reconhecido como outra pessoa, copiando

atributos faciais de um usuário genuíno para fotos, telas eletrônicas ou máscaras. O segundo caso é a *ofuscação*, a qual busca ocultar ou remover a identidade do próprio atacante usando métodos, como óculos, maquiagem, peruca e rosto disfarçado.

Já de acordo com a propriedade geométrica do ataque, os PAs faciais são (amplamente) classificados em ataques 2D e 3D. Os PAs faciais 2D são realizados apresentando atributos faciais usando fotos ou vídeos para o sensor. Fotos impressas, fotos cortadas de olhos/boca e reprodução digital de vídeos são variantes comuns de ataques 2D. Com a maturidade da tecnologia de impressão 3D, a máscara facial 3D tornou-se um novo tipo de PA para ameaçar os sistemas de Reconhecimento Facial Automático. Comparadas com os PAs 2D tradicionais, as máscaras faciais são mais realistas em termos de cor, textura e estrutura geométrica. As máscaras 3D são feitas de diferentes materiais, por exemplo, máscaras duras/rígidas podem ser feitas de papel, resina, gesso ou plástico, enquanto máscaras macias e flexíveis são geralmente compostas de silicone ou látex (YU et al., 2022).

Figura 1 – Tipos de ataques de face spoofing.




Fonte: Yu et al. (2022)

É válido ressaltar que este trabalho busca detectar PAs focados na personificação com propriedades geométricas em 2D, que são os ataques mais comuns, geralmente realizados através de fotos ou vídeos.

## 2.2 Visão estéreo

De acordo com Hamzah e Ibrahim (2016), Visão Estéreo é um ramo da visão computacional que aborda um importante problema: a reconstrução das coordenadas tridimensionais de pontos para a estimativa de profundidade. Um sistema de visão estéreo consiste em uma câmera estéreo, ou seja, duas câmeras acopladas geralmente horizontalmente (uma a esquerda e outra a direita). As duas imagens capturadas simultaneamente por essas câmeras são então processadas para a recuperação de informações visuais de profundidade.

Visão Estéreo é uma das maneiras em que os humanos percebem a profundidade. A palavra estéreo significa justamente “dois”. Sendo assim, o sistema de visão estéreo humano inspirou os sistemas de visão estéreo computacionais. É possível encontrar diferentes tipos de

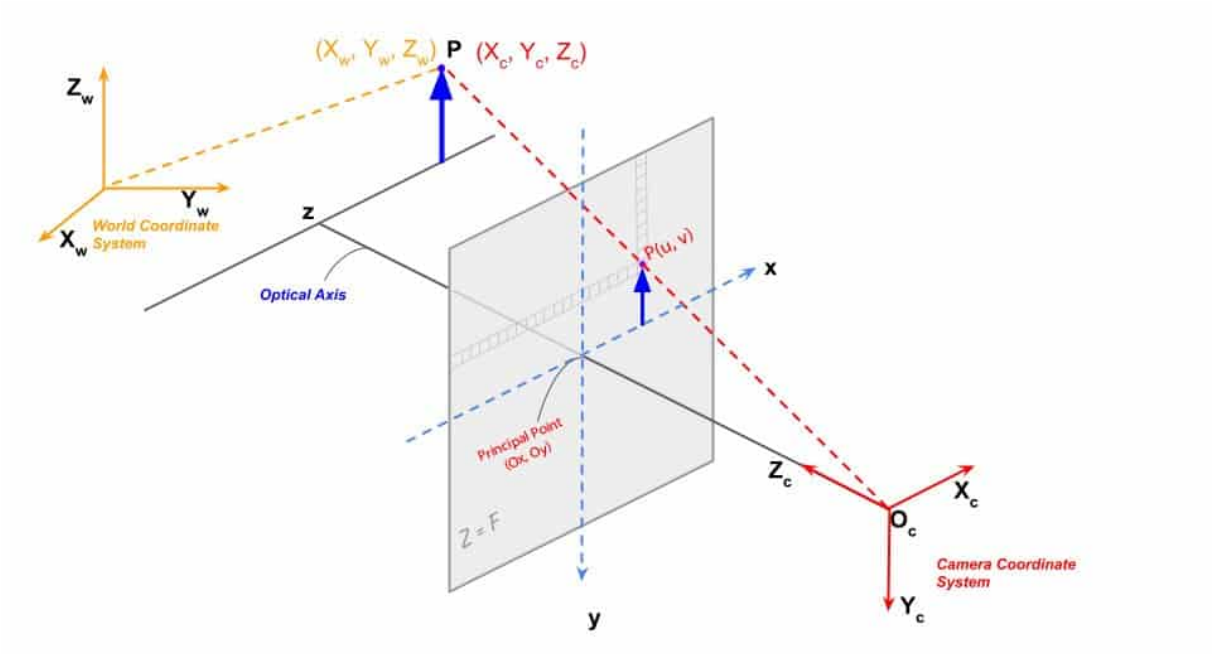
aplicações que utilizam a visão estéreo em tempo real, como por exemplo: carros autônomos, jogos 3D e navegação autônoma de robôs (HAMZAH; IBRAHIM, 2016).

### 2.2.1 A geometria da visão estéreo

Segundo Bradski e Kaehler (2008), uma imagem é uma projeção 2D de um objeto 3D do mundo real para um plano de uma imagem. Como mostra a Figura 2, o processo de formação de uma imagem pode ser descrito através dos seguintes sistemas de coordenadas:

1. Coordenadas de Mundo (3D, unidade: metros);
2. Coordenadas da Câmera (3D, unidade: metros);
3. Coordenadas do Plano da Imagem (2D, unidade: *pixels*).

Figura 2 – Sistemas de coordenadas para a formação de imagens.




Fonte: Mallick e Kukil (2021)

Ao falar sobre câmeras, é crucial entender o processo de geração de uma imagem. No entanto, para compreender essa operação, é preciso conhecer os parâmetros da câmera. O procedimento para determinar os parâmetros das lentes e do sensor de imagem é denominado calibração da câmera. (BRADSKI; KAEHLER, 2008).

Os parâmetros da câmera podem ser divididos em internos e externos. Os parâmetros internos se referem às características internas do sistema de câmera/lente, como por exemplo: a distância focal, o centro óptico e coeficiente radial de distorção das lentes. Já os parâmetros

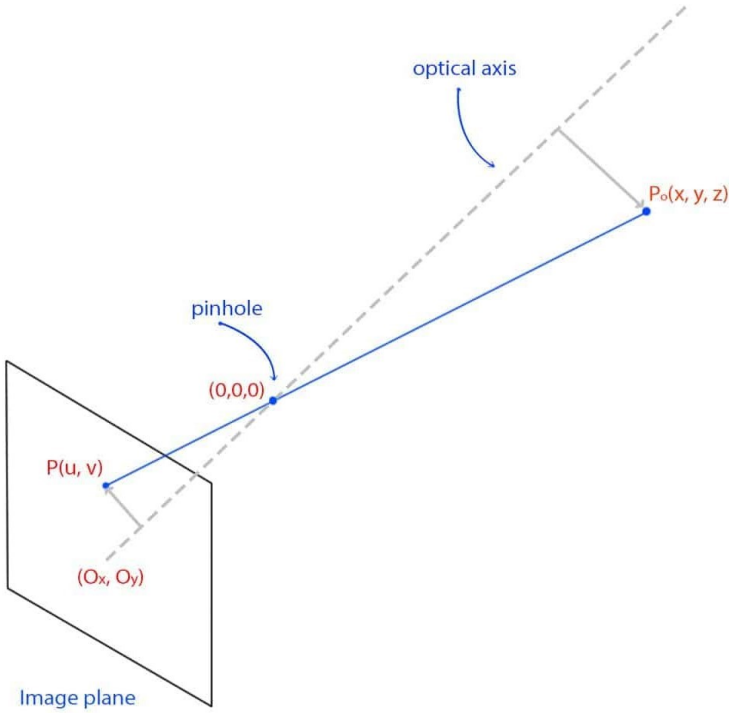
externos estão ligados à orientação (rotação e translação) da câmera em relação a algum sistema de coordenadas de mundo.

Com a obtenção desses parâmetros é possível construir uma matriz chamada Matriz de Transformação, através da qual realizamos o mapeamento de coordenadas do mundo real para as coordenadas em pixel de uma imagem (BRADSKI; KAEHLER, 2008).

Sabendo agora que um ponto 3D no mundo real pode ser mapeado para uma imagem (sistema 2D) através de uma Matriz de Transformação, ainda fica o questionamento: "como obter a profundidade de um determinado ponto a partir da perspectiva da câmera?".

Considere o seguinte cenário da Figura 3: uma câmera *pinhole* que captura a imagem de um ponto $P_0$ do mundo real, onde $P_0(x,y,z)$ é a posição desse ponto no sistema de coordenadas de mundo e $P(u,v)$ é o seu correspondente no plano da imagem.

Figura 3 – Formação de imagem em uma câmera *pinhole*.




Fonte: Mallick e Kukil (2021)

Segundo Bradski e Kaehler (2008), as coordenadas $u$ e $v$ do ponto $P(u,v)$ são obtidas pelas equações de projeção de perspectiva:

$$u = f_x \frac{x}{z} + o_x \tag{2.1}$$

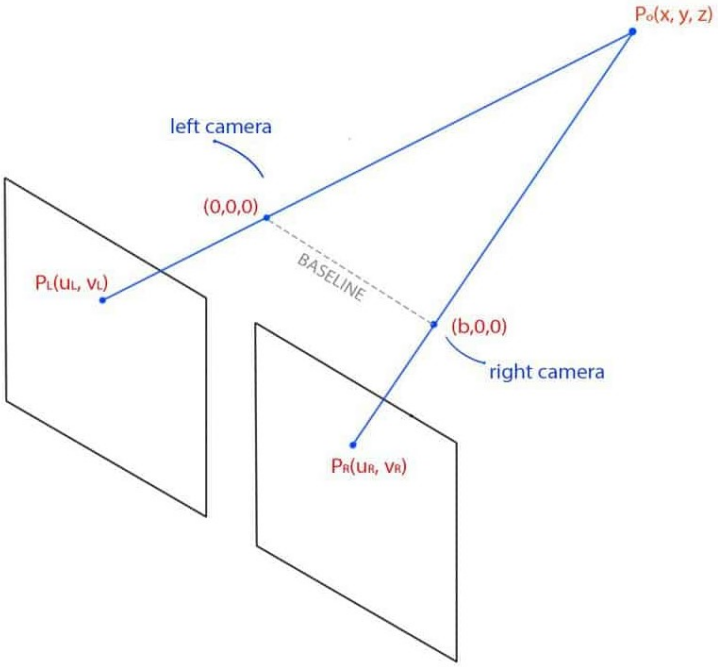
$$v = f_y \frac{y}{z} + o_y \tag{2.2}$$

em que,

- $f_x$, $f_y$, $u$, $v$, $o_x$, $o_y$ são parâmetros conhecidos em unidades de pixel;
- $f_x$ e $f_y$ são as distâncias focais ao longo dos eixos $x$ e $y$ respectivamente;
- $(o_x, o_y)$ é o ponto em que o eixo óptico intercepta o plano da imagem.

Uma vez que se tem apenas as duas equações (2.1) e (2.2), não é possível calcular os valores para $x$, $y$ e $z$. Para encontrá-los, se faz necessário duas câmeras. Assim, na Figura 4, outra câmera idêntica é posicionada formando um sistema estéreo. Por simplificação, assume-se que ambas as câmeras não possuem distorção nas lentes.

Figura 4 – Formação de imagens em um *setup* estéreo




Fonte: Mallick e Kukil (2021)

A partir da Figura 4, pode-se observar os seguintes aspectos: a distância entre os dois centros das câmeras, denotada por $b$, é chamada de linha de base; $P_L(u_L, v_L)$ e $P_R(u_R, v_R)$ são as projeções do ponto $P_0$ no plano das imagens esquerda e direita, respectivamente. Assim, de acordo com Bradski e Kaehler (2008), esse cenário fornece as seguintes equações:

$$u_L = f_x \frac{x}{z} + o_x \tag{2.3}$$

$$u_R = f_x \frac{x-b}{z} + o_x \tag{2.4}$$

$$v_L = f_y \frac{y}{z} + o_y \tag{2.5}$$

$$v_R = f_y \frac{y}{z} + o_y \tag{2.6}$$

Resolvendo o sistema formado pelas equações (2.3), (2.4), (2.5) e (2.6), obtém-se:

$$x = \frac{b(u_L - o_x)}{u_L - u_R} \tag{2.7}$$

$$y = \frac{b f_x (v_L - o_y)}{f_y (u_L - u_R)} \tag{2.8}$$

$$z = \frac{b f_x}{u_L - u_R} \tag{2.9}$$

onde $z$ representa a profundidade do ponto $P_0$ em relação a câmera, a qual é diretamente proporcional à linha de base.

### 2.2.2 Disparidade

Ao olhar de perto as duas imagens capturadas por mono câmeras, observa-se que elas não são idênticas. A disparidade é facilmente notável ao combinar as duas imagens em uma única imagem com a contribuição de 50% de cada uma, como mostra a Figura 5. Existe uma diferença de posições entre os pontos correspondentes, essa diferença é chamada de disparidade (HAMZAH; IBRAHIM, 2016).

Figura 5 – Exemplo de disparidade

Fonte: Elaborada pelo autor

A disparidade e a profundidade são inversamente proporcionais. Isso que dizer que quando um objeto está mais próximo da câmera a disparidade será maior, sendo a profundidade menor, e vice-versa. Isso pode ser notado através da Figura 5 em que a mão está bem próxima a câmera e apresenta uma grande diferença entre as suas sobreposições.

Encontrar os pontos correspondentes entre imagens é uma das tarefas mais desafiadoras nesse cenário de visão estéreo. Alguns algoritmos, que fazem correspondências entre *features* e métodos similares podem ser utilizados. Contudo, imagens capturadas de câmeras de alta-resolução possuem milhões de pixels, fazendo com que esse processo se torne altamente intensivo ao relizar a busca em toda a imagem.

Todavia, a câmera utilizada nesse trabalho é a OAK-D Lite, cujas características serão detalhadas na Seção 4.1, é uma câmera estéreo calibrada em que as imagens são retificadas (conceito apresentado na Seção 2.2.3). Assim, através da teoria da geometria epipolar, o espaço de busca para a corrêspodência entre dois pontos se reduz ao longo de uma linha horizontal entre as duas imagens (HARTLEY; ZISSERMAN, 2003).

### 2.2.3 Desafios na estimativa de profundidade

Apesar de toda a base teórica que existe por trás do cálculo de profundidade através de imagens estéreo, na prática a estimativa de profundidade não é tão simples. Bradski e Kaehler (2008) cita três pontos essenciais para que os cálculos mostrados na Seção 2.2.1 sejam aplicados:

- As câmeras devem estar niveladas;
- As imagens devem ser coplanares;
- Não deve existir distorção óptica.

É difícil obter o cenário ideal em um par estéreo. É comum encontrar câmeras que estão desalinhadas e que produzam imagens que são coplanares. Porém, isso pode ser corrigido através da retificação estéreo, que consiste em projetar os planos de imagem esquerdo e direito em um plano comum, paralelo à linha de base. Já as distorções ópticas são minimizadas através dos parâmetros da câmera obtidos através do processo de calibração.

Entretanto, estas não são as únicas barreiras associadas à visão estéreo. A mensuração de profundidade também possui suas limitações, provocando resultados não desejados em situações que carecem de texturas ou são repetitivas. Adicionalmente, há uma distância específica dentro da qual o trabalho deve ser realizado. Os objetos não podem estar muito próximos nem muito distantes da câmera estéreo (MALLICK; KUKIL, 2021).

## 2.3 *Deep learning*

*Machine Learning* (ML), ou Aprendizado de Márquina, é um ramo da inteligência artificial que provê aos sistemas a habilidade de aprender automaticamente e evoluir a sua performance através de experiências sem ser explicitamente programado. O foco está no desenvolvimento de aplicações que são capazes de acessar dados e usá-los para aprender. O processo de aprendizagem inicia-se através da observação de dados com o objetivo de detectar padrões e assim fazer uma decisão. Dessa forma, esses sistemas são projetados para emular a inteligência humana aprendendo com o ambiente ao redor (NAQA; MURPHY, 2015).

Já *Deep Learning* (DL), ou Aprendizado Profundo, é um ramo de *machine learning* que é particulamente focado em redes neurais artificiais com múltiplas camadas. De acordo com LeCun, Bengio e Hinton (2015), DL permitiu que os modelos computacionais que são compostos de diversas camadas de processamento pudessem aprender representações de dados com diferentes níveis de abstração. Esses métodos aumentaram de forma significativa o estado-da-arte no que se refere a reconhecimento de fala; reconhecimento visual de objetos; detecção de objetos, entre outros domínios.

O aprendizado profundo descobriu estruturas complexas em grandes conjuntos de dados usando o algoritmo de retropropagação para indicar como uma máquina deve alterar seus parâmetros internos. Esses parâmetros são usados para calcular os resultado em cada camada a partir do resultado da camada anterior (LECUN; BENGIO; HINTON, 2015).

### 2.3.1 Redes neurais convolucinais

Entre os diferentes tipos de redes neurais de aprendizado profundo, as *Convolutional Neural Networks* (CNNs), ou Redes Neurais Convolucionais, se destacam sendo uma das mais estudadas intensivamente. Graças ao rápido crescimento na quantidade de dados anotados e as grandes melhorias na capacidade computacional de processadores gráficos, as pesquisas com CNNs emergiram de forma notável e ocuparam o estado-da-arte nos resultados de várias tarefas, com destaque para as que possuem imagens como dados de entrada (GU et al., 2018).

CNNs são otimizadas para o processamento de dados com uma topologia do *grid*, como as imagens, as quais podem ser compreendidas como um *grid* de pixels de duas dimensões. Elas foram inspiradas na organização do córtex visual animal e projetadas para imitar o padrão de conectividade dos neurônios no cérebro humano (LECUN et al., 1998).

A principal característica de uma CNN é a utilização de camadas convolucionais, as quais são compostas por vários filtros, também conhecidos como *kernels*. Esses filtros se "movem" através da imagem ou da saída da camada, calculando a convolução entre os pesos dos *kernels* e a entrada da camada e produz um mapa de *features*. Essas operações ajudam a capturar características locais na imagem, como formatos e bordas (GOODFELLOW; BENGIO;

COURVILLE, 2016).

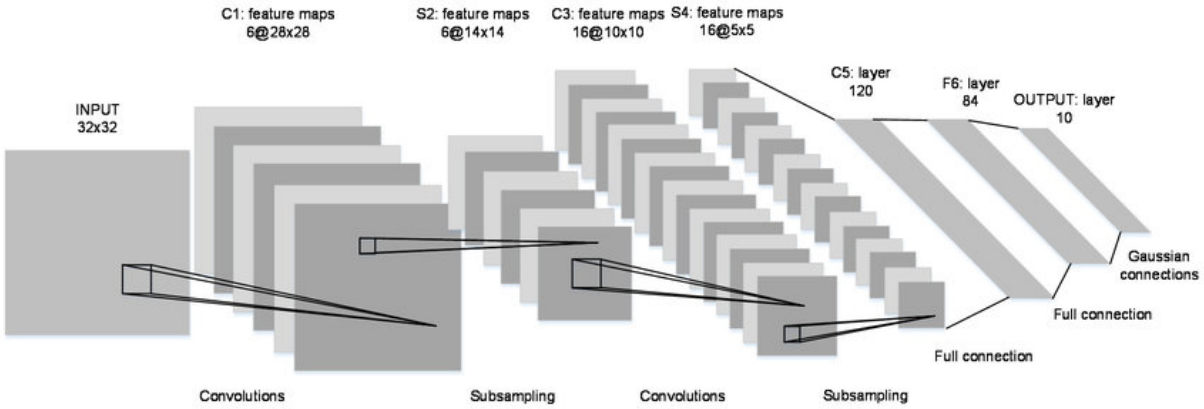
Outro importante aspecto das Redes Neurais Convolucionais é a utilização das camadas de *pooling*, muitas vezes referidas como camadas de subamostragem (*downsampling*). Conforme Zeiler e Fergus (2014), as camadas de pooling são usadas para reduzir a dimensionalidade de cada mapa de *features* resguardando a sua informação mais importante. Isso torna o processamento da rede menos custoso computacionalmente e aumenta sua robutez a ruídos e variações.

É comum em Redes Neurais Convolucionais ter a presença de *fully-connected layers*, que pode ser traduzido como camadas totalmente conectadas. Elas são similares à forma em que os neurônios são organizados em uma rede neural tradicional. Portanto, cada nó na camada totalmente conectada está diretamente conectado a todos os nós presentes na camada anterior e nos nós da próxima camada (ALBAWI; MOHAMMED; AL-ZAWI, 2017).

Assim, uma típica arquitetura para uma CNN consiste em uma camada de entrada, várias camadas alternadas de convolução e *pooling*, seguidas por camadas totalmente conectadas e uma camada de saída. Um exemplo é a arquitera da LeNet (LECUN et al., 1998), uma das CNNs mais tradicionais, a qual está ilustrada na Figura 6.

Nas CNNs, as camadas convolucionais iniciais são empregadas para captar características de baixo nível, como arestas, cantos e texturas. Já as camadas mais avançadas são capazes de identificar elementos de alto nível, como formas, objetos ou rostos (KRIZHEVSKY; SUTSKEVER; HINTON, 2017).

Figura 6 – Arquitetura da LeNet.




Fonte: LeCun et al. (1998)

O treinamento de uma Rede Neural Convolucional envolve aprender os parâmetros ótimos que minimizam a função de perda (*loss function*). O processo começa com a inicialização dos parâmetros do modelo com pequenos valores aleatórios. Em seguida, para cada entrada no conjunto de treinamento, a rede calcula uma saída por meio de um processo chamado propagação direta (*feedforward*). A diferença entre a saída prevista e a saída real é quantificada usando uma função de perda.

Depois, o algoritmo de retropropagação calcula quanto cada parâmetro contribuiu para o erro. Esses erros são usados em um algoritmo de descida de gradiente (ou uma variante do mesmo) para ajustar os parâmetros do modelo. O processo é repetido ao longo de várias épocas até que o desempenho do modelo pare de melhorar em um conjunto de validação. Técnicas como dropout, normalização em lote ou aumento de dados são frequentemente usadas durante o treinamento para melhorar a capacidade do modelo de generalizar (GOODFELLOW; BENGIO; COURVILLE, 2016).

### 2.3.2 Transfer learning

Conforme explica Yosinski et al. (2014) modelos de aprendizado profundo, como as Redes Neurais Convolucionais (CNNs) ou Redes Neurais Recorrentes (RNNs), podem ter milhões de parâmetros, o que requer uma grande quantidade de dados para treiná-los do zero de forma eficaz. No entanto, muitos projetos da área de DL possuem dados limitados. Nesse contexto, foram desenvolvidas técnicas que buscam superar essa limitação na quantitade de dados disponíveis para se treinar uma rede neural. Entre elas, uma das mais conhecidas e difundidas na comunidade de DL é o *Transfer Learning*.

*Transfer Learning* é um método de *machine learning* em que um modelo pré-treinado, geralmente para uma tarefa específica, é utilizado como ponto de partida para outro modelo em uma segunda tarefa relacionada. Essa abordagem ganhou muita força na comunidade de ML devido à sua capacidade de acelerar o tempo de treinamento da rede neural e melhorar o desempenho de modelos em tarefas com dados limitados (PAN; YANG, 2010).

Os modelos tradicionais de ML/DL geralmente iniciam o processo de aprendizado do zero, utilizando uma inicialização aleatória. Por outro lado, no *Transfer Learning*, o conhecimento adquirido durante a resolução de um problema é aplicado ou "transferido" para um problema diferente, que geralmente está relacionado com o problema inicial. Isso é particularmente benéfico quando uma tarefa possui dados de treinamento insuficientes, pois ele aproveita o aprendizado de uma tarefa relacionada que possui dados amplos (WEISS; KHOSHGOFTAAR; WANG, 2016).

No campo da visão computacional, é comum utilizar modelos de CNNs pré-treinadas em grandes conjuntos de dados, como o *ImageNet* (DENG et al., 2009), que consiste em usar milhões de imagens categorizadas em milhares de classes que em seguida são ajustadas para uma tarefa de classificação mais específica e menor, como por exemplo, treinar um classificador de gatos e cachorros (YOSINSKI et al., 2014).

*Transfer Learning* é comumente dividida em duas estratégias: *fine-tuning* e *feature extraction*. Ao utilizar a estratégia de *fine-tuning*, um modelo pré-treinado é adaptado a uma nova tarefa semelhante, com isso o processo de retropropagação continua e os pesos pré-existentes no modelo são atualizados, considerando os dados do novo problema. Isso significa que não

apenas as camadas superiores (que normalmente são específicas da tarefa), mas também as camadas inferiores (que geralmente são mais genéricas) são ajustadas durante o novo processo de treinamento.

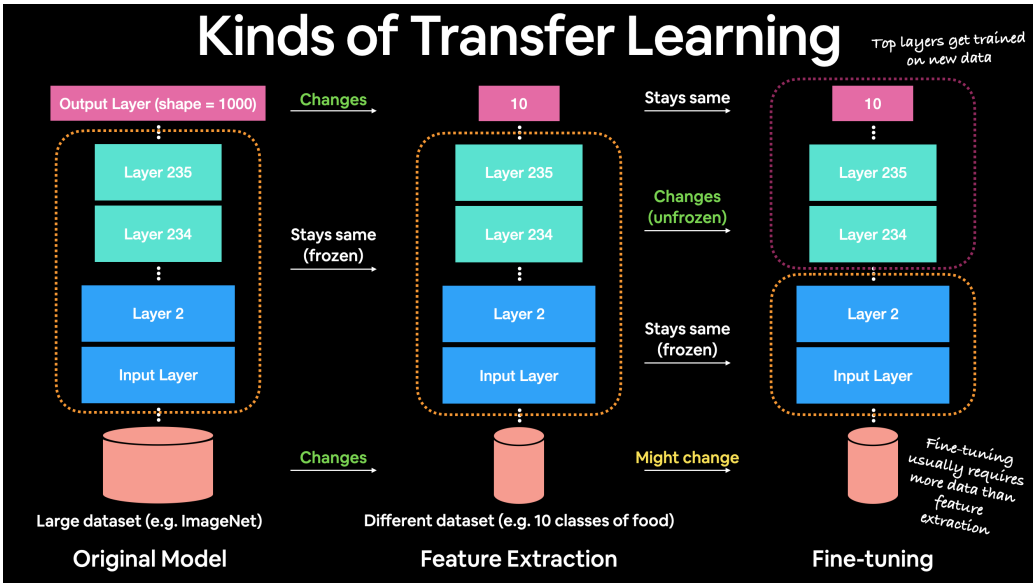
Assim, a ideia por trás dessa estratégia é que o modelo pré-treinado já tenha aprendido alguns recursos úteis que podem funcionar como uma boa inicialização para a nova tarefa, levando a um melhor desempenho e uma convergência mais rápida (YOSINSKI et al., 2014).

Já a estratégia de *feature extraction* trata o modelo pré-treinado como um extrator de características, nesse caso, os dados de entrada se propagam até um determinado ponto e, em seguida, um novo classificador é treinado a partir da saída das camadas pré-existentes. Os pesos do modelo pré-treinado são normalmente congelados ou mantidos estáticos durante todo o processo de treinamento (OQUAB et al., 2014).

A *feature extraction* é computacionalmente mais eficiente do que o *fine-tuning*, pois menos parâmetros precisam ser atualizados durante o treinamento. Muitas vezes, é o método preferível quando o novo conjunto de dados é pequeno e/ou muito semelhante ao conjunto de dados original usado para pré-treinamento.

A Figura 7 representa bem as diferenças que existem entre as estratégias de *fine-tuning* e *feature-extraction*. Enquanto que no *fine-tuning* parte da rede neural que é utilizada como base é retreinada para justamente fazer o "ajuste fino", na *feature extraction* somente a camada superior, geralmente composta por camadas totalmente conectadas e a camada de saída, são treinadas. Além disso, na primeira abordagem geralmente é necessário um *dataset* mais amplo.

Figura 7 – Os tipos de *Transfer Learning*




Fonte: Bourke (2021)

Nesse trabalho, que tem como objetivo treinar uma CNN capaz de distinguir imagens de disparidade de rostos reais de *spoofing*, utilizamos *Transfer Learning* através da abordagem de

*feature extraction*. A camada base da rede neural MobileNetV2 (SANDLER et al., 2018) será apresentada em detalhes na Seção 4.3 e será usada como um extrator de características onde a sua saída é acoplada para treinar uma nova rede totalmente conectada.

# 3 TRABALHOS RELACIONADOS

No trabalho "*Deep Learning for Face Anti-Spoofing: A Survey*", Yu et al. (2022) faz um *review* sobre a evolução das técnicas de *Face Anti-Spoofing* (FAS), destacando o crescimento do interesse nessa área ao longo dos anos. Com o surgimento de *datasets* acadêmicos de larga escala que possuem uma grande diversidade de tipos de *spoofing*, como por exemplo: CelebA-Spoof (ZHANG et al., 2020) e SiW-M (LIU et al., 2019) e o aumento do poder computacional, as metodologias FAS baseadas em *deep learning* atingiram uma performace notável e se tornaram dominante.

Cronologicamente, os primeiros métodos para a detecção de ataques de apresentação baseavam-se em características determinadas por humanos (*handcrafted feature*) (YU et al., 2022). A maioria dos algoritmos tradicionais foram projetos com base na vivacidade humana, tais como: o piscar de olhos (PAN et al., 2007), movimento da face e da cabeça (WANG; DING; FANG, 2009), rastreamento do olhar (ALI; DERAVI; HOQUE, 2012), dentre outras caraterísticas.

Outra classe de técnicas *anti-spoofing* considerada tradicional foi a utilização de descritores, como LBP (PEREIRA et al., 2013), SURF (BOULKENAFET; KOMULAINEN; HADID, 2016) e HOG (KOMULAINEN; HADID; PIETIKÄINEN, 2013). Eles foram usados para extração efetiva de padrões de *spoofing* de vários espaços de cores (RGB, HSV e YCbCr), baseando-se principalmente na qualidade das imagens e nas características das texturas locais. Essas metodologias provaram ser bastante robustas na detecção de diferentes tipos de ataques de apresentação facial (REHMAN; PO; LIU, 2020).

Subsequentemente, foram propostas técnicas de *Face Anti-Spoofing* baseadas em *deep learning* para ataques de aprensentação facial estáticos e dinâmicos. A maioria desses trabalhos (GEORGE; MARCEL, 2019) trata FAS como um problema de classificação binário, em que por exemplo um rosto real seria "0", enquanto um *spoofing* seria "1", e vice-versa. Nesse contexto de crescimento dos métodos baseados *deep learning*, surge a aplicação das Redes Neurais Convolucionais (CNNs) na extração de características para distinguir rostos "verdadeiros" de "falsos" (LI et al., 2016; YANG; LEI; LI, 2014).

Um ponto importante a ser ressaltado, é que a maioria dos estudos desse domínio são focados em imagens RGB obtidas de câmeras comercias (YU et al., 2022). Contudo, é notável o crescimento de pesquisas que utilizam sensores especializados, como é caso deste trabalho, que usa uma câmera estéreo. Por exemplo, Rehman, Po e Liu (2020) e Wu et al. (2020) aplicam os conceitos de dispariridade e profundidade que são obtidos através de imagens estéreo e treinam CNNs específicas para produrizem um detector de Face Anti-Spoofing. Em ambos os trabalhos, é possível notar a grande capacidade de detectar ataques de imagens 2D e de vídeos.

Assim, o presente trabalho situa-se em um cenário que ainda está sendo bastante explorado. Os estudos que envolvem a aplicação de CNNs com imagens estéreo são recentes e já produziram bons resultados. Todavia, a dinamicidade e a diversidade de ataques de apresentação facial tornam necessário o estudo contínuo e o desenvolvimento de novas metodologias para serem implementadas.

# 4 METODOLOGIA

Nesta seção são discutidas em detalhe as ferramentas e técnicas que foram empregadas para alcançar os objetivos do trabalho. Inicialmente, apresenta-se a câmera OAK-D Light, um dispositivo de visão computacional avançado que desempenhou um papel crucial na coleta e processamento de dados. Em seguida, descreve-se o processo de construção do conjunto de dados, formado por imagens de disparidade. Além disso, detalha-se a arquitetura da Rede Neural Convolucional (CNN) proposta, explicando as camadas, parâmetros e técnicas utilizadas para a sua construção. Por fim, apresenta-se os hardwares e softwares usados durante o projeto, especificando as suas configurações.

## 4.1 OAK-D Lite

Sendo o principal hardware utilizado para a construção do *dataset* desse trabalho, a *OAK Lite* é uma *spatial camera*, que faz parte da família *OAK*, desenvolvida pela empresa *Luxonis* em parceria com a *OpenCV*. Esta câmera é uma solução completa para a implementação de soluções de visão computacional e aprendizado profundo, pois ela possui múltiplas câmeras e aceleradores de inferência de rede neural incorporados diretamente em um dispositivo do compacto. Ela é alimentada pela Unidade de Processamento Visual (VPU) Intel Movidius Myriad X para inferência de rede neural (SHARMA, 2022a). A Figura 8 mostra um exemplar de uma *OAK Lite*.

Figura 8 – *OAK-D Lite.*

Fonte: Luxonis (2023b)

A *OAK-D Lite* é considerada uma Câmera Espacial de Inteligência Artificial (IA), o que significa que ela é capaz de tomar decisões baseadas em duas propriedades: a percepção visual e a percepção de profundidade. Na percepção visual, encontra-se a habilidade de IA “ver” e

"interpetrar" o ambiente de forma visual. Já a percepção de profundidade consiste na capacidade de compreender o quão longe os objetos estão.

Conforme apresenta Sharma (2022a), a *OAK-D Lite* é formada pelos seguintes componentes:

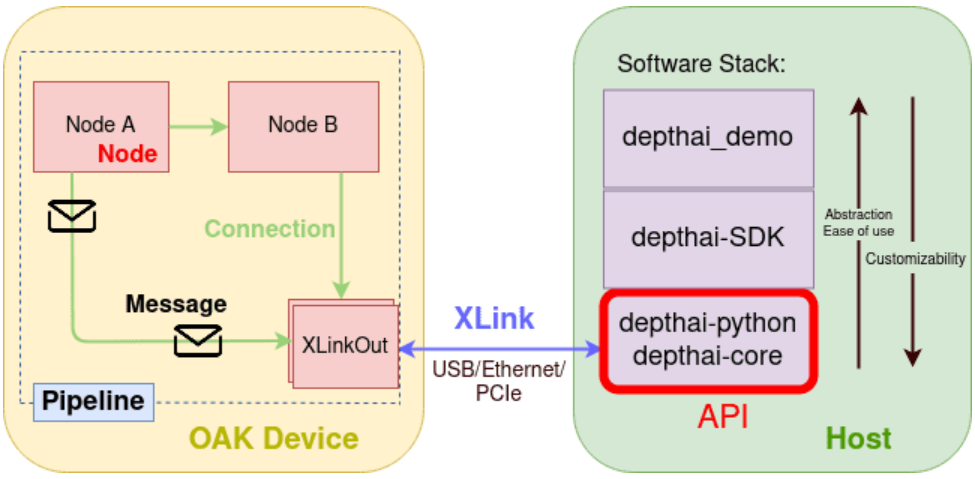
- Câmera RGB de alta-resolução de 12 Megapixel, 4K até 60 FPS;
- Um par estéreo - sistema de duas câmeras utilizado para a percepção de profundidade - com resolução de $640 \times 480p$ com 120 FPS;
- Intel® Myriad™ X Visual Processing Unit (VPU): podereso processador capaz de executar 4 trilhões de operações por segundo.

Além disso, a *OAK-D Lite* tem uma linha de base (distância entre o par estéreo) de $7,5cm$, o que é bem similar à distância os dois olhos de um ser humano.

A *API DepthAI* (LUXONIS, 2023a) permite que um dispositivo (por exemplo, um computador ou qualquer microprocessador) conecte, configure e comunique-se com *OAK-D Lite* usando tanto a API *Python (depthai-python)* quanto a *API C++ (depthai-core)*. A biblioteca *depthai-python* fornece uma espécie de "ponte" em *Python* para a biblioteca *C++ depthai-core*.

Dessa forma, a *API DepthAI* fornece um modelo de programação usando o conceito de um *pipeline*, que nada mais é do que um conjunto de nós, sendo um nó uma unidade com algumas entradas e saídas. As mensagens são enviadas de um nó para outro com links entre eles. A Figura 9 mostra uma arquitetura de alto nível da conexão entre um dispositivo e a câmera *OAK-D*, a pilha de software que o DepthAI fornece, e o que acontece dentro da câmera (SHARMA, 2022b).

Figura 9 – Arquitetura de alto-nível da *API DepthAI*.




Fonte: Sharma (2022b)

## 4.2 Construção do dataset

O principal objetivo deste trabalho consiste na confecção de uma CNN capaz de classificar rostos reais e spoofings faciais 2D a partir de imagens de disparidade. Para que isso possa ser atingido, foi necessária a importante etapa de construção do dataset de treinamento, isto é, a base de dados a partir da qual a CNN extrairá as informações necessárias para diferenciar os tipos imagens.

Como apresentado na seção 2.2.2, a disparidade consiste na distância entre os pontos correspondentes nas imagens esquerda e direita de um par estéreo. Assim, a partir dos valores de disparidade é possível construir uma imagem da cena com diferentes colorações. Um dos principais mapas de cor, frequentemente aplicados a imagens de disparidade, é uma escala em tons de cinza, em que a variação na tonalidade da imagem representa uma variação na disparidade/profundidade de um elemento da cena em relação à câmera.

Considerando que os valores dos *pixels* em uma imagem formada em tons de cinza variam de 0 à 255, em que 0 é o preto e 255 é o branco. Assim, originalmente quanto maior for a disparidade (menor a profundidade), mais próximo do branco um pixel é, e vice-versa. Entretanto, com objetivo de facilitar a visualização nesse tipo de imagem, foi aplicado um pré-processamento para inverter os bits dos valores dos pixels da imagem de disparidade, o que produziu o seguinte efeito: quanto mais próximo um elemento estiver da câmera, mais próximo do preto seu *pixel* é (maior disparidade), e quanto mais distante, o pixel será mais próximo do branco (menor disparidade). A Figura 10 retrata justamente esse cenário.

A Figura 10a mostra um exemplo de imagem de disparidade que foi utilizada para treinamento da rede neural, enquanto que a Figura 10b é a sua imagem RGB correspondente. Nota-se que por se tratar de imagens de um recorte de um rosto real, é possível observar na imagem de disparidade o contorno do rosto em um tom de cinza, enquanto que o fundo da imagem aparece totalmente em branco por estar mais distante. Assim, pode-se notar diferenças de profundidades que foram capturadas no próprio rosto. Observando a Figura 10b indentifica-se que o rosto está levemente inclinado produzindo uma diferença de disparidade entre as partes esquerda e direita do rosto.

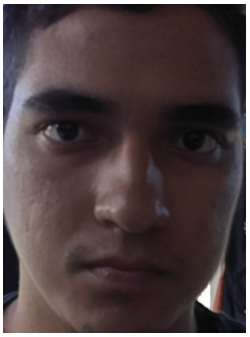
Nos casos em que há ataque de apresentação facial 2D, seja por rostos em telas (imagens e vídeos), em impressões, entre outras formas, a imagem de disparidade se comporta de maneira diferente, como mostrado na Figura 11. Na Figura 11a, pode-se ver uma imagem de disparidade totalmente uniforme com a mesma tonalidade de cinza, o que é teoricamente esperado, já que o rosto contido na Figura 11b está em um plano e, dessa forma, todos os pontos possuem a mesma profundidade, portanto eles possuem a mesma disparidade.

As imagens das Figuras 10 e 11 foram obtidas com a câmera *OAK-D Lite* (Seção 4.1), seguindo as funcionalidades definadas pela *API DepthAI* em *Python*. Foi construído um pipeline específico com o seguinte fluxo:

Figura 10 – Exemplo de imagem de disparidade de um rosto real em tons de cinza.

(a) Imagem de disparidade do rosto.

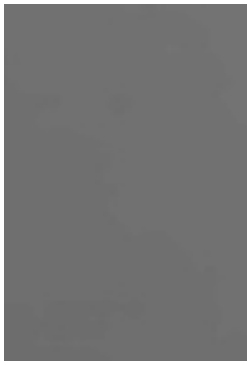
(b) Imagem RGB do rosto.



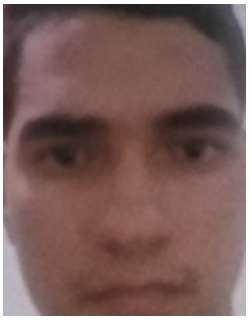
Fonte: Elaborada pelo autor.

Figura 11 – Exemplo de imagem de disparidade de um *spoofing* facial 2D em tons de cinza

(a) Imagem de disparidade de um *spoofing* 2D.



(b) Imagem RGB do*spoof* 2D.



Fonte: Elaborada pelo autor.

1. Na *OAK-D Lite*, através de um nó *"StereoDepth"* que tem como entrada as imagens do par de câmeras estéreo, obtém-se um mapa de disparidade e a imagem da câmera direita é retificada;

2. Em seguida, essa imagem retificada é pré-processada e passa por um detector facial (baseado na SqueezeNet (IANDOLA et al., 2016)), o qual fornece a *bounding box* do rosto;

3. O mapa de disparidade, a imagem retificada e a *bounding box* do rosto são enviados da *OAK-D Lite* para o computador;

4. Por fim, no computador, aplicava-se uma operação de inversão de bits do mapa de disparidade, e então recorta-se o rosto da imagem de disparidade formando imagens semelhantes às Figuras 10a e 11a.

É crucial destacar neste processo o emprego da imagem retificada capturada pela câmera direita do par estéreo para a detecção facial. A escolha desta abordagem se deve à correspondência entre as coordenadas em pixels da imagem retificada e da imagem de profundidade. Isso significa que, uma vez identificado um ponto na primeira imagem, para determinar sua disparidade, basta consultar as coordenadas correspondentes na imagem de disparidade.

Simultaneamente ao processo anterior, o rosto também é capturado nas imagens da câmera RGB da *OAK-D Lite*. Esse procedimento foi adotado para se obter um par RGB/disparidade para cada imagem. As imagens de disparidade foram utilzadas para o treinamento/teste da CNN e as imagens RGB foram utilizadas para comparação com outros modelos *anti-spoofing*.

### 4.2.1 Especificações do dataset

O primeiro aspecto a ser destacado é que os experimentos foram realizados em um ambiente controlado, onde a distância entre o rosto e a câmera variou de $0,5m$ a $1m$. Este detalhe é fundamental, pois foi precisamente neste intervalo de distância que se pôde discernir as maiores diferenças entre as imagens de disparidade de um rosto real e um spoofing.

Outro critério considerado para a montagem do conjunto de dados foi a condição de iluminação. Para a captura, tanto do rosto real quanto do spoofing, foram empregadas quatro condições distintas de iluminação:

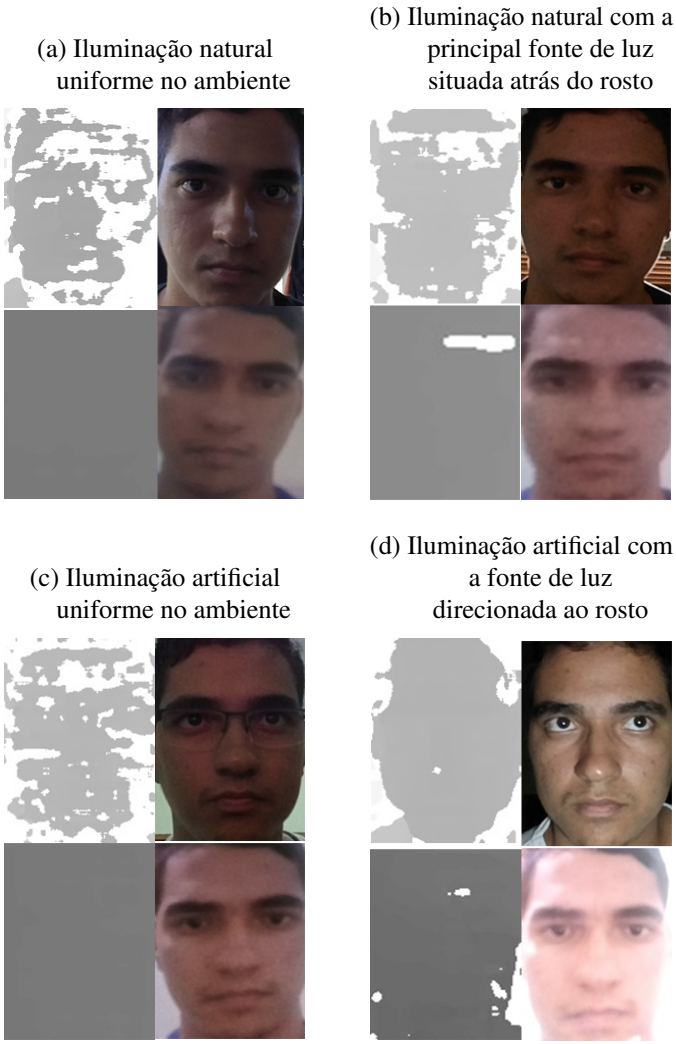
- iluminação natural uniforme no ambiente;
- iluminação natural com a principal fonte de luz situada atrás do rosto;
- iluminação artificial uniforme no ambiente;
- iluminação artificial com a fonte de luz direcionada ao rosto.

A intenção de variar as condições de iluminação era a de criar um conjunto de dados mais robusto e diversificado. Na Figura 12, mais especificamente nas Figuras 12a, 12b, 12c e 12d pode-se ver um par RGB/disparidade para cada condição de luz com um exemplo de rosto real e de um *spoofing* facial.

Visando a criação de um *dataset* mais abrangente e confiável, outra variação implementada foi a captura de fotos de *spoofing* facial com diversas inclinações de tela. A finalidade de inclinar a tela era gerar regiões com diferentes profundidades. No entanto, pode-se ver na 13 que as imagens de disparidade obtidas não correspondem à silhuetas faciais.

Por fim, é importante frisar que a CNN desenvolvida neste estudo foi treinada exclusivamente com imagens de disparidade, por exemplo as da Figura 14. No total, o conjunto de dados é construído por 3542 dessas imagens, subdivididas em: 1765 correspondentes a um rosto real

Figura 12 – Rostos capturados em diferentes condições de iluminação.




Fonte: Elaborada pelo autor.

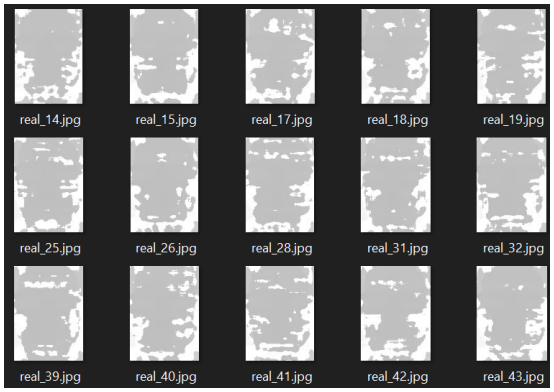
Figura 13 – Imagens de disparidade capturas através da inclinação da tela.

Fonte: Elaborada pelo autor.

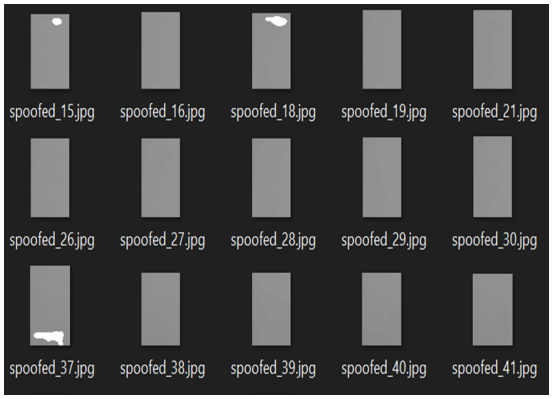
(Figura 14a) e 1777 relacionadas ao *spoofing* facial 2D (Figura 14b), obtidas a partir de uma tela de celular.

Figura 14 – Amostras do *dataset*

(a) Imagens de disparidade de rostos reais




(b) Imagens de disparidade de *spoofing* facial 2D




Fonte: Elaborada pelo autor.

## 4.3 Arquitetura da rede neural convolucional implementada

Com o objetivo de desenvolver uma Rede Neural Convolucional, construída a partir de uma aprendizagem supervisioda, capaz de classificar as imagens de disparidade como "spoofed" ou "real", adotou-se a técnica de *Transfer Learning*, especificamente a *feature extraction*. Nesta abordagem, a estrutura básica da rede neural MobileNetV2 (SANDLER et al., 2018) foi empregada como "extratora de características", enquanto uma nova rede totalmente conectada foi treinada com base nas características fornecidas pela saída da MobileNetV2.

A MobileNetV2, apresentada por Sandler et al. (2018), é uma evolução significativa da arquitetura MobileNet, projetada para dispositivos móveis e hardwares embarcados. É um modelo de aprendizado profundo leve e eficiente que permite a implantação de modelos de última geração em dispositivos com recursos computacionais limitados.

Por ser treinada na base de dados da ImageNet, que possui mais de 1,4 milhão de imagens e 1000 categorias para classificação, a MobileNetV2 se mostra bastante adequada para aplicação do *Transfer Learning* para problemas de classificação de imagens. A ideia por trás dessa técnica é que, se um modelo for treinado em um conjunto de dados grande e geral, esse modelo pode ser efetivamente considerado como um modelo genérico do mundo visual.

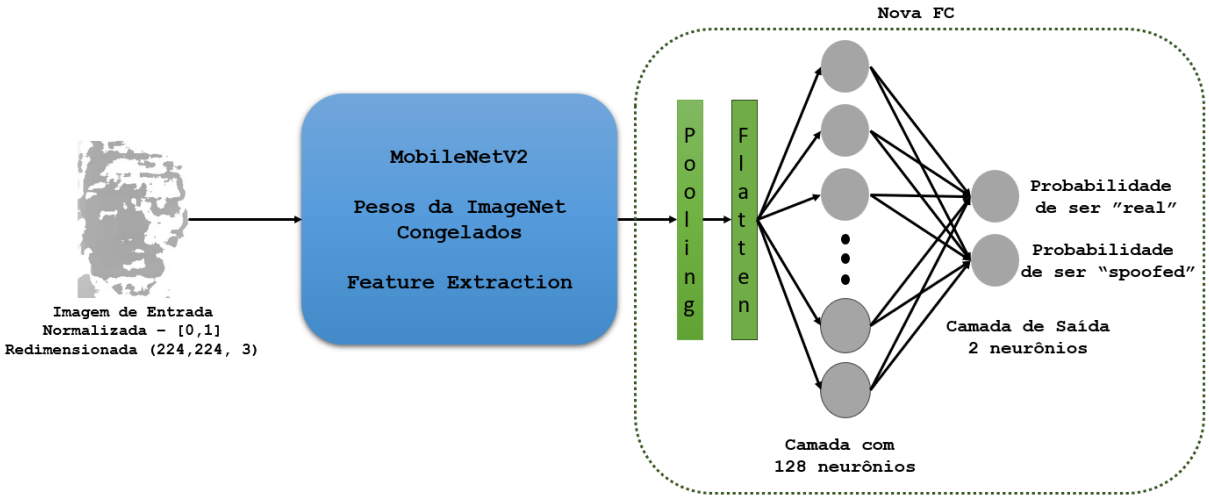
Assim, do ponto de vista da implementação, foi empregado apenas o modelo base da MobileNetV2, sem a utilização das camadas superiores, responsáveis pela classificação das 1000 categorias. No lugar dessas camadas, acoplou-se uma nova rede neural totalmente conectada e personalizada, apta a classificar apenas duas categorias de imagens. Essa nova rede neural apresenta a seguinte estrutura:

1. Camada de Pooling Médio 2D de tamanho $(7 \times 7)$;
2. Camada Flatten;

3. Camada Densa com 128 neurônios e função de ativação *ReLu*;
4. Dropout de 0.5;
5. Camada Densa com 2 neurônios e função de ativação $Softmax$.

Dessa forma, a estrutura final da CNN implementada é ilustrada na Figura 15. A entrada corresponde à imagem de disparidade normalizada, com valores no intervalo $[0, 1]$ e redimensionada para $(224, 224, 3)$, compatível com o formato utilizado pela MobileNetV2. Por outro lado, a saída é composta por dois neurônios, dos quais um fornece a probabilidade de ser 0 ("real") e o outro a probabilidade de ser 1 ("spoofed"). Importante destacar que essas probabilidades são complementares, isto é, a soma das inferências sempre resulta em 1.

Figura 15 – Arquitetura da CNN implementada.




Fonte: Elaborada pelo autor.

Um aspecto importante sobre essa arquitetura é o fato de que durante o treinamento todos os pesos da MobileNetV2 permancem congelados, significando que eles não são atualizados. Apenas os parâmetros relativos à nova camada totalmente conectada sofrem alterações pelo algoritmo de retropropagação. Assim, essa CNN totaliza 2.422.210 parâmetros, dos quais 2.257.984 não são treinados e 164.226 são treináveis.

## 4.4 Configurações de treinamento, validação e teste

A distribuição das imagens no conjunto de dados construído para as etapas de treinamento, validação, teste e comparação da CNN foi a seguinte: das 3542 imagens disponíveis, 2908 foram usadas para treinamento, teste e validação, em que 80% foram destinadas para treinamento e validação - subdivididas em 70% para treinamento (2036 imagens) e 10% para validação (290 imagens). Os 20% restantes, correspondendo a 582 imagens, foram reservados para o teste.

Além disso, 634 imagens restantes foram separadas para comparação com a rede "*Silent-Face-Anti-Spoofing*", sendo 313 imagens de rostos 'reais' e 321 imagens de ataque de apresentação facial.

Os hiperparâmetros adotados para o treinamento da CNN foram os seguintes:

- *Initial Learning Rate* (Taxa de Aprendizado Inicial): $10^{-4}$;
- *Batch Size* (Tamanho do Lote): 32;
- Épocas: 20.

Quanto ao hardware, foi usado um notebook da marca *Dell*, modelo Inspiron 14 Série 7000, equipado com um processador Intel(R) Core(TM) i7-7500U de 2.70GHz e 16 GB de memória RAM. Com esta configuração, o tempo de treinamento da CNN foi de aproximadamente 30 minutos.

Para a construção da CNN, utilizou-se a linguagem *Python*, com os *frameworks Tensorflow* e *Keras*. A manipulação e o processamento das imagens foram feitos com as bibliotecas *OpenCV* e *Numpy*. E, finalmente, o processamento e a visualização dos resultados foram realizados com o auxílio das bibliotecas *Scikit-learn* e *Matplotlib*.

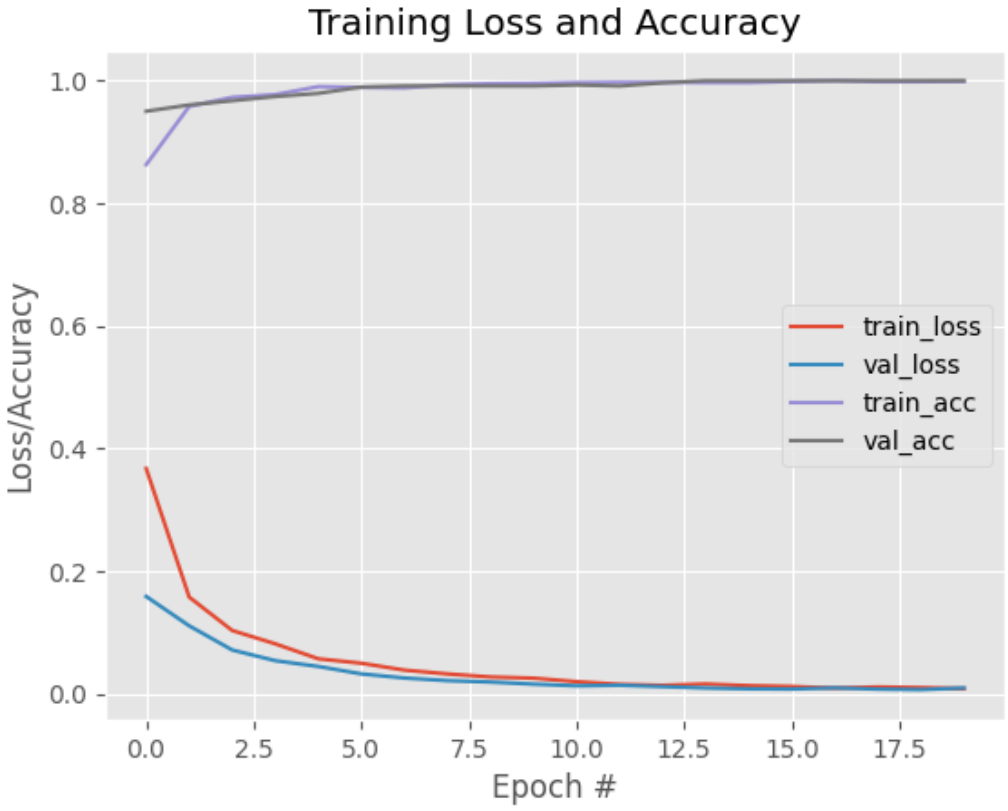
# 5 RESULTADOS E DISCUSSÕES

Esta seção inicia com a apresentação dos resultados das fases de treinamento, validação e teste da CNN proposta. Posteriormente, realiza-se uma comparação entre o modelo sugerido e a rede *open source* "*Silent-Face-Anti-Spoofing*".

## 5.1 Resultados de treinamento, validação e teste do modelo proposto

Durante as etapas de treinamento e validação, a CNN proposta alcançou uma acurácia de 100%, isto é, foi capaz de distinguir corretamente todos os casos de "rostos reais" e "*spoofing*" facial presentes nas imagens de disparidade. A Figura 16 ilustra a evolução da acurácia e do "*loss*" ao longo das 20 épocas, mostrando que as curvas de treinamento e validação exibem comportamentos similares até atingir a acurácia de 100% por volta da época 14.

Figura 16 – Curva de aprendizagem - Treinamento e Validação.



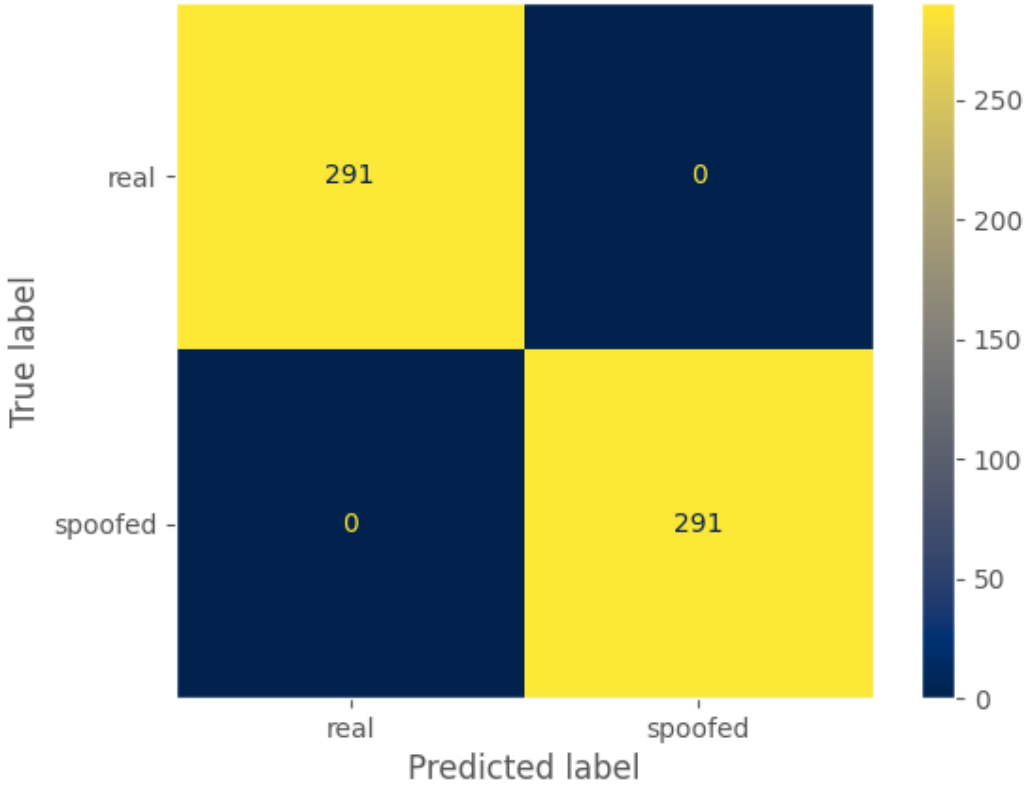

Fonte: Elaborada pelo autor.

Esse comportamento representado pela Figura 16 é extremamente benéfico. Isso evidencia um alto desempenho da rede, indicando que ela conseguiu aprender efetivamente os padrões

presentes nos dados. Além disso, a semelhança entre as curvas sugere que o modelo tem uma boa capacidade de generalização, ou seja, pode ser aplicado com sucesso a novos dados. No entanto, apesar dessas características positivas, é crucial validar o modelo com um conjunto de testes independente, para garantir que não esteja super ajustado (*overfitting*) aos dados de treinamento.

A próxima etapa envolveu exatamente isso, a aplicação do modelo proposto ao conjunto de teste independente, que também alcançou uma acurácia de 100%. A matriz de confusão correspondente a essa fase de teste é representada pela Figura 17. Pode-se observar que a arquitetura proposta aprendeu de maneira eficaz e genérica as características que distinguem as imagens de disparidade, não havendo assim um *overfitting* em relação ao conjunto de treinamento.

Figura 17 – Matriz de confusão para a etapa de teste.




Fonte: Elaborada pelo autor.

O fato de não haver falsos positivos ou falsos negativos na matriz de confusão do teste indica que o modelo é altamente confiável, minimizando tanto o risco de aceitar *spoofings* como de rejeitar usuários legítimos. Isso tornaria o modelo particularmente útil para aplicações de segurança e autenticação, onde a precisão e confiabilidade são cruciais.

Contudo, embora esses resultados sejam promissores, é importante lembrar que eles foram obtidos em um ambiente controlado e podem não representar completamente o desempenho do modelo em cenários do mundo real. É crucial observar que a eficácia desta rede neural convolucional foi validada em uma configuração específica, onde a distância entre o rosto e a

câmera variou de $0,5m$ a $1m$. Isso indica que a rede pode funcionar de forma otimizada dentro desses limites de distância.

No entanto, a performance do modelo pode variar para distâncias fora desse intervalo, já que as imagens de disparidade não apresentam variações nítidas à medida que se distancia da câmera. Assim, para uma aplicação mais ampla em ambientes reais, onde a distância entre o rosto e a câmera pode variar significativamente, seria necessário realizar treinamentos e testes adicionais para avaliar e otimizar o desempenho do modelo nessas circunstâncias.

## 5.2 Comparação com a rede Silent-Face-Anti-Spoofing

Com o objetivo de validar a eficácia da rede neural aqui proposta, esta seção dedica-se a uma comparação com a *Silent-Face-Anti-Spoofing* (MINIVISION, 2020), que é uma rede *open-source*. A *Silent-Face-Anti-Spoofing*, por sua vez, trabalha exclusivamente com imagens RGB, ao contrário da abordagem adotada neste trabalho. Portanto, foi implementada a seguinte estratégia: para cada imagem RGB de um rosto, capturou-se a respectiva imagem de disparidade. Desta forma, imagens oriundas do mesmo contexto ou cena foram apresentadas para ambas as redes neurais.

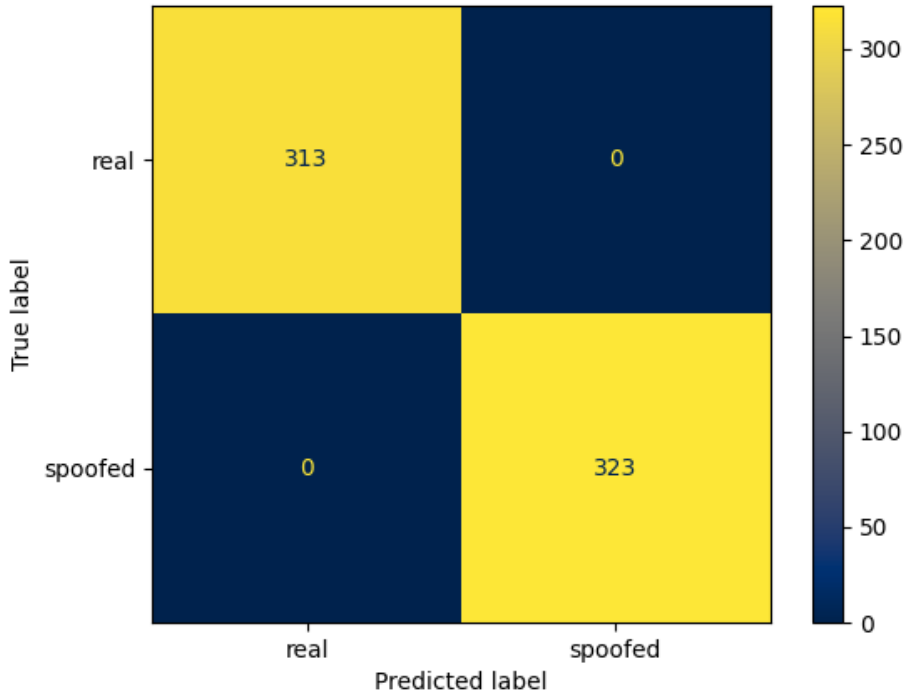
É importante destacar que as imagens de disparidade apresentadas nesta etapa não foram anteriormente expostas à CNN desenvolvida neste trabalho. Portanto, os resultados aqui apresentados não só avaliam a acurácia geral da rede, mas também fornecem evidências adicionais sobre a capacidade de generalização do modelo proposto.

A Figura 18 apresenta a matriz de confusão gerada para a CNN criada neste estudo. Novamente, a acurácia alcançada foi de 100%, confirmando os resultados obtidos na fase de testes. Isso significa que, dentro do ambiente controlado proposto, é viável desenvolver um modelo, baseado em visão estéreo, capaz de distinguir com alta precisão *spoofings* faciais em 2D.

A Figura 19 exibe a matriz de confusão obtida para a rede *Silent-Face-Anti-Spoofing*. Ao contrário do modelo apresentado neste trabalho, a precisão alcançada foi de 80%, com uma quantidade considerável de Falsos Negativos. Isto implica que, embora a *Silent-Face-Anti-Spoofing* tenha sido capaz de detectar a maioria dos ataques de *spoofing* facial, frequentemente classificou rostos reais como *spoofings* de forma equivocada.

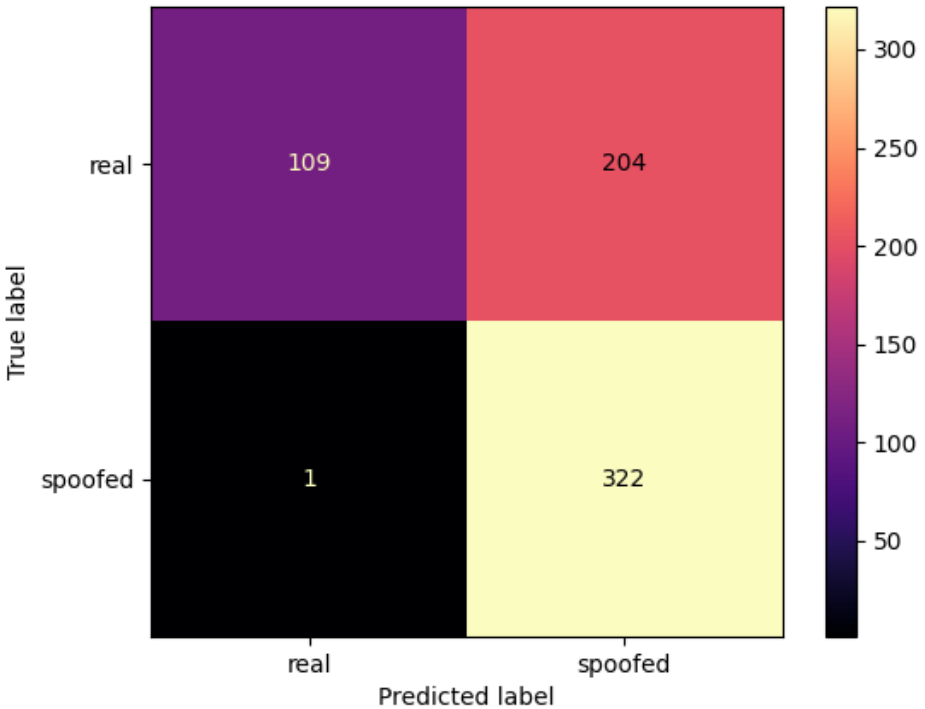
Essa alta incidência de Falsos Negativos, ou seja, casos em que rostos reais foram erroneamente classificados como *spoofing*, tem implicações significativas. Isso pode levar a uma experiência de usuário frustrante em um sistema de reconhecimento facial, por exemplo, onde usuários legítimos poderiam ter frequentemente o acesso negado. Além disso, em um cenário de segurança, esse tipo de erro pode ser prejudicial, pois pessoas autorizadas poderiam ser impedidas de acessar determinadas áreas ou informações.

Figura 18 – Etapa de comparação: matriz de confusão para a CNN proposta.




Fonte: Elaborada pelo autor.

Figura 19 – Etapa de comparação: matriz de confusão para a rede *Silent-Face-Anti-Spoofing*.




Fonte: Elaborada pelo autor.

Este resultado destaca um aspecto bastante usual no campo de *Face Anti-Spoofing*, conforme aponta a pesquisa de Yu et al. (2022). A performance pode decrescer substancialmente quando o modelo é aplicado em conjuntos de dados ou cenários que divergem do contexto

originalmente usado para o seu treinamento.

Por fim, a diferença entre os desempenhos dessas duas redes destaca a eficácia do uso de imagens de disparidade em comparação com imagens RGB tradicionais para a tarefa de detecção de *spoofing* facial. A capacidade da CNN proposta de alcançar 100% de precisão no conjunto de testes e nas imagens inéditas sugere que o uso de imagens de disparidade pode fornecer uma vantagem significativa para essa tarefa em determinados contextos e aplicações.

# 6 CONCLUSÃO

Em suma, este trabalho propôs uma abordagem para a detecção de ataques de *spoofing facial* em 2D, que faz o uso de imagens de disparidade (obtidas através de uma cámera estéreo - OAK-D Lite) em conjunto com uma arquitetura de Rede Neural Convolucional implementada através da técnica de *Transfer Learning*. O desempenho do modelo proposto alcançou 100% de precisão, tanto na fase de treinamento e teste quanto na avaliação com imagens inéditas. Tal resultado atesta a eficácia do uso de imagens de disparidade para esta tarefa.

Ao comparar o desempenho do modelo proposto com a rede *open-source* Silent-Face-Anti-Spoofing, evidenciou-se o quanto redes neurais para FAS sofrem em condições distintas das quais foram treinadas. Embora a rede Silent-Face-Anti-Spoofing tenha tido um desempenho razoável em termos gerais, ela não conseguiu igualar a precisão da CNN proposta, principalmente devido a uma alta taxa de falsos negativos.

No entanto, é importante enfatizar que a rede proposta neste trabalho demonstrou alta eficiência em um ambiente controlado e dentro de uma faixa de distância específica (entre $0,5m$ e $1m$). A performance pode ser potencialmente afetada em situações onde a distância excede esse intervalo. Adicionalmente, para aprimorar a avaliação da rede, foi sugerido testá-la em uma variedade maior de condições, incluindo diferentes cenários de iluminação, tipos de tela, vídeos e impressões em papel físico. Isso proporcionará uma análise mais ampla contra uma maior diversidade de tipos de ataques de spoofing 2D.

Para trabalhos futuros, a elaboração de um conjunto de dados mais abrangente, que incorpore uma ampla gama de ataques de spoofing facial 2D, pode fortalecer ainda mais a robustez desta abordagem. Adicionalmente, sujeitar o modelo a uma maior quantidade de cenários ou ataques desconhecidos poderia resultar em uma avaliação mais genérica da efetividade da rede. Por exemplo, explorar formas de combinar a visão estéreo e as imagens de disparidade com modelos 3D será uma pesquisa conduzida e uma sugestão seria treinar um detector facial que consiga determinar a região de um rosto a partir das imagens de disparidade.

# REFERÊNCIAS


ALBAWI, S.; MOHAMMED, T. A.; AL-ZAWI, S. Understanding of a convolutional neural network. In: IEEE. *2017 international conference on engineering and technology (ICET)*. [S.l.], 2017. p. 1–6.

ALI, A.; DERAVI, F.; HOQUE, S. Liveness detection using gaze collinearity. In: IEEE. *2012 Third International Conference on Emerging Security Technologies*. [S.l.], 2012. p. 62–65.

BOULKENAFET, Z.; KOMULAINEN, J.; HADID, A. Face antispoofing using speeded-up robust features and fisher vector encoding. *IEEE Signal Processing Letters*, IEEE, v. 24, n. 2, p. 141–145, 2016.

BOURKE, D. *Transfer Learning with TensorFlow Part 1: Feature Extraction*. 2021. https://dev.mrdbourke.com/tensorflow-deep-learning/04_transfer_learning_in_tensorflow_part_1_feature_extraction/. Acessado em: 01 de junho de 2023.

BRADSKI, G.; KAEHLER, A. *Learning OpenCV: Computer vision with the OpenCV library*. [S.l.]: "O'Reilly Media, Inc.", 2008.

BROWN, C.; BALLARD, D. Computer vision. *Englewood, NJ, Prentice Hall*, 1982.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255.

GALBALLY, J.; MARCEL, S.; FIERREZ, J. Biometric antispoofing methods: A survey in face recognition. *IEEE Access*, IEEE, v. 2, p. 1530–1552, 2014.

GEORGE, A.; MARCEL, S. Deep pixel-wise binary supervision for face presentation attack detection. In: IEEE. *2019 International Conference on Biometrics (ICB)*. [S.l.], 2019. p. 1–8.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

GU, J. et al. Recent advances in convolutional neural networks. *Pattern recognition*, Elsevier, v. 77, p. 354–377, 2018.

HAMZAH, R. A.; IBRAHIM, H. Literature survey on stereo vision disparity map algorithms. *Journal of Sensors*, Hindawi, v. 2016, 2016.

HARTLEY, R.; ZISSERMAN, A. *Multiple view geometry in computer vision*. [S.l.]: Cambridge university press, 2003.

IANDOLA, F. N. et al. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016.

KOMULAINEN, J.; HADID, A.; PIETIKÄINEN, M. Context based face anti-spoofing. In: IEEE. *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. [S.l.], 2013. p. 1–8.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, AcM New York, NY, USA, v. 60, n. 6, p. 84–90, 2017.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, Ieee, v. 86, n. 11, p. 2278–2324, 1998.

LI, L. et al. An original face anti-spoofing approach using partial convolutional neural network. In: IEEE. *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. [S.l.], 2016. p. 1–6.

LIU, Y. et al. Deep tree learning for zero-shot face anti-spoofing. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 4680–4689.

LUXONIS. *DepthAI API*. 2023. https://github.com/luxonis/depthai-python. Acessado em 01 de junho de 2023.

LUXONIS. *OAK-D Lite*. 2023. https://www.luxonis.com/. Acessado em: 01 de junho de 2023.

MALLICK, S.; KUKIL. *Stereo Vision and Depth Estimation using OpenCV AI Kit*. 2021. https://learnopencv.com/stereo-vision-and-depth-estimation-using-opencv-ai-kit/. Acessado em: 01 de junho de 2023.

MINIVISION. *Silent-Face-Anti-Spoofing*. 2020. https://github.com/infinityglow/Silent-Face-Anti-Spoofing. GitHub repository.

NAQA, I. E.; MURPHY, M. J. *What is machine learning?* [S.l.]: Springer, 2015.

OQUAB, M. et al. Learning and transferring mid-level image representations using convolutional neural networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2014. p. 1717–1724.

PAN, G. et al. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: IEEE. *2007 IEEE 11th international conference on computer vision*. [S.l.], 2007. p. 1–8.

PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, IEEE, v. 22, n. 10, p. 1345–1359, 2010.

PEREIRA, T. de F. et al. Lbp- top based countermeasure against face spoofing attacks. In: SPRINGER. *Computer Vision-ACCV 2012 Workshops: ACCV 2012 International Workshops, Daejeon, Korea, November 5-6, 2012, Revised Selected Papers, Part I 11*. [S.l.], 2013. p. 121–132.

REHMAN, Y. A. U.; PO, L.-M.; LIU, M. Slnet: Stereo face liveness detection via dynamic disparity-maps and convolutional neural network. *Expert Systems with Applications*, Elsevier, v. 142, p. 113002, 2020.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 4510–4520.

SHARMA, A. Introduction to OpenCV AI kit (OAK). In: CHUGH, P. et al. (Ed.). *PyImageSearch*. [S.l.: s.n.], 2022. Https://pyimg.co/dus4w.

SHARMA, A. OAK-D: Understanding and running neural network inference with DepthAI API. In: CHUGH, P. et al. (Ed.). *PyImageSearch*. [S.l.: s.n.], 2022. Https://pyimg.co/8ynbk.

THORAT, S.; NAYAK, S.; DANDALE, J. P. Facial recognition technology: An analysis with scope in india. *arXiv preprint arXiv:1005.4263*, 2010.

WANG, L.; DING, X.; FANG, C. Face live detection method based on physiological motion analysis. *Tsinghua Science & Technology*, Elsevier, v. 14, n. 6, p. 685–690, 2009.

WEISS, K.; KHOSHGOFTAAR, T. M.; WANG, D. A survey of transfer learning. *Journal of Big data*, SpringerOpen, v. 3, n. 1, p. 1–40, 2016.

WU, X. et al. Single-shot face anti-spoofing for dual pixel camera. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 16, p. 1440–1451, 2020.

YANG, J.; LEI, Z.; LI, S. Z. Learn convolutional neural network for face anti-spoofing. *arXiv preprint arXiv:1408.5601*, 2014.

YOSINSKI, J. et al. How transferable are features in deep neural networks? *Advances in neural information processing systems*, v. 27, 2014.

YU, Z. et al. Deep learning for face anti-spoofing: A survey. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 45, n. 5, p. 5609–5631, 2022.

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

ZHANG, Y. et al. Celeba-spoof: Large-scale face anti-spoofing dataset with rich annotations. In: SPRINGER. *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*. [S.l.], 2020. p. 70–85.