Steven Roger dos Santos Soares

Avaliação de Redes Neurais Convolucionais na Classificação da Posição do Olhar em Vídeos do Exame Cover Test

São Luís – MA

2025

Steven Roger dos Santos Soares

Avaliação de Redes Neurais Convolucionais na Classificação da Posição do Olhar em Vídeos do Exame Cover Test

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Universidade Federal do Maranhão

Orientador: Prof. Dr. João Dallyson Sousa De Almeida

São Luís – MA

2025

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Diretoria Integrada de Bibliotecas/UFMA

Santos Soares, Steven Roger dos.
Avaliação de Redes Neurais Convolucionais na Classificação da Posição do Olhar em Vídeos do Exame Cover Test / Steven Roger dos Santos Soares. - 2025.
54 f.

Orientador(a): João Dallyson Sousa de Almeida.
Monografia (Graduação) - Curso de Ciência da Computação, Universidade Federal do Maranhão, São Luís, 2025.

1. Cover Test. 2. Estrabismo. 3. Direção do Olhar. 4. Classificação. 5. Redes Convolucionais. I. Sousa de Almeida, João Dallyson. II. Título.

Steven Roger dos Santos Soares

Avaliação de Redes Neurais Convolucionais na Classificação da Posição do Olhar em Vídeos do Exame Cover Test

Monografia apresentada ao curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 14 de Agosto de 2025, São Luís – MA:

Prof. Dr. João Dallyson Sousa De Almeida
Orientador
Universidade Federal do Maranhão

Profa. Dra. Simara Vieira da Rocha
Examinadora
Universidade Federal do Maranhão

Prof. Dr. Geraldo Braz Junior
Examinador
Universidade Federal do Maranhão

São Luís – MA
2025

Agradecimentos

Quero agradecer primeiramente a Deus por me permitir chegar até esta fase de minha graduação, foram 4 anos e meio de experiências que me edificaram na área que amo.

Agradeço aos meus pais que, se não fosse pela ajuda deles, eu não conseguiria realizar este feito, me apoiando e me permitindo ter o alívio de focar em minha graduação durante estes anos. Não há palavras o suficiente que expressem o carinho e gratidão que tenho a eles.

Agradeço ao professor João Dallyson que me aceitou em seu laboratório e, com sua paciência e conhecimento na área de processamento de imagem e aprendizado de máquina, me orientou durante o desenvolvimento deste trabalho de conclusão de curso.

Agradeço profundamente aos meus amigos de fora quanto de dentro da Universidade, cujo apoio e incentivo foram fundamentais para que eu persistisse em minha jornada até conquistar a graduação que tanto almejei. Com vocês, até os momentos difíceis vieram acompanhados de risos.

Por fim, agradeço aos meus amigos do VipLab pelo apoio, pelos conselhos e pelos momentos únicos de troca de ideias que tornaram as sextas-feiras agregadoras ao longo dessa jornada.

"Combati o bom combate, terminei a corrida, guardei a fé"

2 Timóteo 4:7-8

Resumo

A classificação da posição do olhar é uma tarefa complexa e de grande relevância para aplicações clínicas, como o diagnóstico de estrabismo. Um dos meios seria pelo exame oftalmológico chamado Cover Test, que consiste em testar o nível de desvio de estrabismo em cinco posições diferentes do olhar sendo fundamental para o planejamento cirúrgico. Este estudo tem como objetivo avaliar o desempenho de diferentes arquiteturas de redes neurais convolucionais (CNNs) na predição de cinco posições distintas do olhar em vídeos de Cover Test. Foram selecionadas nove arquiteturas modernas e robustas aplicando de forma sistemática, técnicas como Early Stopping, Fine-Tuning e Descongelamento Progressivo com diferentes taxas de aprendizado, visando otimizar o desempenho e gerar modelos capazes de realizar inferência sobre a base de vídeos do Cover Test, conduzindo os experimentos baseados na ideia de Grid Search, onde dois cenários experimentais foram avaliados: o primeiro empregando o Descongelamento Progressivo das camadas do modelo, e o segundo utilizando o descongelamento de apenas duas camadas de extração de características, ambos os cenários foram submetidos a diferentes taxas de aprendizado. Para o treinamento, validação e teste, utilizaram-se quatro fontes distintas de dados. A avaliação foi conduzida com base em métricas como acurácia, precisão, F1-Score, recall, além da observação do comportamento da função de perda *(loss)* e matriz de confusão. Por meio dos resultados, foram feitos ranqueamentos dos melhores modelos, que alcançaram até 98% em algumas das métricas durante a validação. Ao filtrar os cinco melhores da validação, observou-se o desempenho variando entre 70% e 98% para Precisão e F1-Score. Nos testes com dados de domínio real, a arquitetura AlexNet se destacou, alcançando precisão de 88% e recall de 85%, enquanto as demais apresentaram resultados inferiores abaixo de 80%.

Palavras-chave: Cover Test, Estrabismo, Direção do olhar, Classificação, Redes Convolucionais, Avaliação de arquiteturas.

Abstract

Gaze position classification is a complex task of great relevance for clinical applications, such as strabismus diagnosis. One method is an ophthalmological examination called the Cover Test, which tests the level of strabismus deviation in five different gaze positions and is essential for surgical planning. This study aims to evaluate the performance of different convolutional neural network (CNN) architectures in predicting five distinct gaze positions in Cover Test videos. Nine modern and robust architectures were selected, systematically applying techniques such as Early Stopping, Fine-Tuning, and Progressive Thawing with different learning rates to optimize performance and generate models capable of performing inference on the Cover Test video database. Experiments were conducted based on the concept of Grid Search. One test case used Progressive Thawing and another used only two Thawed Feature Units, both for varying Learning Rates (LRs). Four distinct data sources were used for training, validation, and testing. The evaluation was conducted based on metrics such as accuracy, precision, F1-Score, and recall, as well as observing the behavior of the loss function (loss) and confusion matrix. The results led to rankings of the best models, which achieved up to 98% in some metrics during validation. When filtering the top five, performance ranged from 70% to 98% for Precision and F1-Score. In tests with real-world data, the AlexNet architecture stood out, achieving 88% precision and 85% recall, while the others performed worse, achieving results below 80

Keywords: Cover Test, Strabismus, Gaze Direction, Classification, Convolutional Networks, Architecture Evaluation.

Lista de ilustrações

Lista de Siglas

ACC Acurácia. 35

CNN Convolutional Neural network. 13, 22

DEXTRO Olho virado a direita. 13

F1 F1-Score. 35

GELU *Gaussian Error Linear Unit*. 24

ILSVRC ImageNet Large-Scale Visual Recognition Challenge. 22

INFRA Olho virado para baixo. 13

LEVO Olho virado a esquerda. 13

LR Learning Rate. 34, 36

ML *Machine Learning*. 20

PPO Olho em direção frontal. 13

PRE Precisão. 35

ReLU *Rectified Linear Unit*. 24

SEN Sensibilidade. 35

SUPRA Olho virado para cima. 13

YOLO You Only Look Once. 17

Sumário

1 Introdução

O avanço das técnicas de inteligência artificial, principalmente no campo da visão computacional, tem permitido o desenvolvimento de sistemas inteligentes capazes de analisar imagens com alta precisão. As Redes Neurais Convolucionais (CNNs Convolutional Neural Networks) destacam-se nesse cenário por sua capacidade de extrair automaticamente características relevantes de imagens, sendo amplamente utilizadas em tarefas como detecção facial, reconhecimento de expressões e análise postural (LECUN; BENGIO; HINTON, 2015).

O estrabismo, por exemplo, segundo (YARKHEIR et al., 2025) consiste no desalinhamento ocular durante a fixação de um objeto, é uma deficiência ocular que prejudica a função visual e a visão binocular da pessoa afetada por essa deficiência. O diagnóstico é feito observando o comportamento em paralelo dos olhos, pode haver dois tipos de desvio possíveis horizontal e vertical. Para seu diagnostico há diferentes técnicas. O método mais utilizado segundo (VALENTE, 2017) é o exame de Cover Test, que consiste na oclusão alternada dos olhos sendo feita a observação do comportamento do olho oposto não ocluído. Essa observação permite identificar a presença de desvios oculares, como tropias ou forias. O Cover Test pode ser feito tanto para um olhar fixo centralizado(primário) quanto para posições diferentes do olhar. De acordo com (LEITE et al., 2021) para ser feita a cirurgia de correção de estrabismo é necessário passar por uma fase de planejamento cirúrgico, onde é feita a coleta de algumas informações importantes, dentre elas a medida do desvio para cinco posições do olhar e a medida de versões que consiste analisar os movimentos conjugados dos olhos, ou seja, a capacidade dos músculos de moverem os olhos coordenadamente em diferentes direções do olhar, com estas informações é possível dizer em qual músculo deve ser feita a cirurgia e estimar valores específicos da dimensão exata de cirurgia sobre o músculo - Por assim, ter conhecimento da direção do olhar em vídeos Cover Test é uma etapa importante e necessária.

Modelos baseados em CNNs têm alcançado resultados robustos na classificação de ângulos da cabeça, permitindo identificar com precisão variações nas direções pitch (inclinação), yaw (guinada) e rol(rotação), mesmo sob diferentes condições de iluminação e oclusão parcial (RUIZ; CHONG; REHG, 2018).

Considerando que, em contexto clínico do exame, as imagens podem variar devido a diferenças de iluminação, dispositivos de captura e posicionamento do rosto, este estudo emprega técnicas de domain generalization, utilizando diferentes bases de dados no treinamento, e realiza avaliação cross-dataset, na qual treino e teste ocorrem em bases distintas, de modo a aprender representações robustas e avaliar a capacidade de generalização dos

modelos para dados provenientes de diferentes fontes (ZHOU et al., 2022), aumentando assim sua capacidade de adaptação.

Neste contexto, o presente estudo tem como objetivo avaliar diferentes arquiteturas de redes neurais convolucionais aplicadas à classificação da posição do olhar para as posições Olho virado a esquerda (LEVO), Olho virado a direita (DEXTRO), Olho virado para cima (SUPRA), Olho virado para baixo (INFRA), Olho em direção frontal (PPO), com foco em sua utilidade como etapa auxiliar em sistemas automatizados de análise de estrabismo, uma vez que a avaliação do estrabismo nas cinco posições do olhar é importante para a realização do planejamento cirúrgico do estrabismo (LEITE et al., 2021). Para avaliação de desempenho, serão utilizadas métricas quantitativas.

1.1 Objetivos

Avaliar o desempenho de diferentes arquiteturas de Redes Neurais Convolucionais (CNNs) na classificação da posição do olhar em imagens faciais, com o objetivo de apoiar sistemas automatizados de diagnóstico de estrabismo baseados em vídeos do Cover Test.

1.1.1 Objetivos Específicos

Para alcançar o objetivo geral, destacam-se como objetivos específicos deste trabalho:

- Selecionar e adaptar nove arquiteturas de Convolutional Neural network (CNN) para a tarefa de classificação da posição do olhar para cinco classes;
- Preparar a base de teste para entrada do modelo, visando à inferência nos vídeos do Cover Test;
- Empregar técnicas de domain generalization utilizando dados obtidos de múltiplas fontes e domínios no treinamento.
- Treinar modelos de redes neurais convolucionais utilizando a técnica de aprendizado por transferência;
- Avaliar os modelos, por meio de avaliação cross-dataset, diante de variações de domínio nos dados de teste da base Cover Test.
- Extrair os quadros (*frames*) dos vídeos da base Cover Test para utilização como entrada da rede durante a fase de teste;
- Realizar a inferência individual em video Cover Test;

1.2 Estrutura do Trabalho

O trabalho segue a seguinte estrutura:

- O Capítulo 2 apresentará os principais trabalhos relacionados que deram base teórica e metodológica para o desenvolvimento deste estudo.
- O Capítulo 3 abordará a fundamentação teórica, apresentando os conceitos, arquiteturas das redes neurais e técnicas aplicadas.
- O Capítulo 4 descreverá as etapas da metodologia proposta, como aquisição e características das bases de dados utilizadas e a metologia empregada.
- O Capítulo 5 irá apresentar e discutir os resultados obtidos nos experimentos realizados
- O Capítulo 6 apresentará as conclusões finais, destacando os principais resultados obtidos e propondo sugestões para trabalhos futuros.

2 Trabalhos Relacionados

Neste capítulo, é mencionada uma coleção de documentos dos estudos que inspiraram e influenciaram este trabalho, cada trabalho ajudou na tomada de decisões que possibilitaram um melhoramento em eficiência e resolução do problema.

Os repositórios consultados para a revisão da literatura foram o Google Scholar, ResearchGate e Arxiv com buscas pelas palavras-chave, classificação, direção do olhar, posição da cabeça e CNNs. Foram priorizados os artigos publicados a partir do ano 2015.

No trabalho realizado por Gourier, Hall e Crowley (2004), os autores propuseram uma abordagem para estimativa da orientação da face baseada na detecção automática de estruturas faciais salientes que se destacam no rosto como nariz, olho, boca e queixo. A sua metodologia usa rastreamento por crominância da pele com estimativa Bayesiana, normalização da imagem facial com momentos estatísticos e Filtro de Kalman, isolando o rosto em uma imagem em miniatura em escala de cinza padronizada em posição e tamanho. O isolamento da face é feito com a intenção de focar o processamento apenas na região de interesse, reduzindo o custo computacional e garantindo desempenho em cima das variações de iluminação, pose e identidade. A padronização da entrada para uma imagem em escala de cinza, por meio da soma RGB, aumentou a taxa de detecção de características como os olhos, que passaram de apenas 1,8% para 98,2% após a normalização. Para gerar a essa imagem foi necessário primeiro passar pela etapa de isolamento do rosto, etapa essa que influenciou diretamente no presente projeto, que adota um procedimento similar de recorte facial como etapa de pré-processamento. Os testes que foram feitos com a base Pointing 04 indicaram 97% de taxa média de detecção dos olhos e erro médio de 5 a 15 graus na pose horizontal em ângulos frontais.

No trabalho de Liu et al. (2015), é proposta uma nova estrutura de aprendizado profundo capaz de fazer a predição de atributos faciais em condições da natureza, ou seja, diferentes poses, iluminações e oclusões, que combina duas Redes Neurais Convolucionais (CNNs), chamadas LNet e ANet. A LNet localiza a região facial de forma progressiva e a ANet extrai características para reconhecimento de atributos. A LNet é treinada de forma fraca, ou seja, usa apenas os rótulos de imagens, sendo treinada com 1000 categorias presentes na ImageNet e depois ajustada para rótulos incluídos manualmente por meio de ajuste fino, aproveitando o que já foi pré-treinado para os novos atributos incluídos manualmente. Como segunda etapa, a ANet visa prever atributos faciais com base na região do rosto detectada. Nela é realizado o aumento de dados, sendo treinada em uma gama maior. Ela é pré-treinada para classificar milhares de identidades faciais, ou seja, é capaz de aprender características discriminativas. Por fim, utilizando os vetores de

características (FCs) gerados pela ANet, são usados classificadores SVM para prever os valores dos atributos.

Já Gupta et al. (2017) propuseram um método de reconhecimento da orientação facial através do mapeamento de imagem com extração de característica e dados faciais. Eles utilizam o método Haar para reconhecer a área facial com uma precisão de 90%. Gupta et al. (2017) em seu treinamento, se refere imagens negativas como sendo imagens onde não terão os objetos alvos, e imagens positivas são imagens em que esta presente o objeto alvo com sua localização. Utilizando a base de dados FERET (Facial Recognition Technology) com mais de 10.000 imagens e mais de 1.000 pessoas. Ele utiliza três classificadores separados, um para boca, outro para o olho direito e outro para o esquerdo. Após a detecção do rosto, por meio das regiões isoladas ele inicia a próxima etapa que é a detecção das características individuais. Para estipular a orientação do rosto, ele usa a distância e tamanho entre essas características, como no eixo X, a pose horizontal pode ser estimada pela simetria do rosto. Após todas essas etapas, é gerada uma imagem nova do modelo e forma do rosto, usando uma segunda imagem de referência é calculada a distância entre os pixels, por meio dessas correspondências usando Threshold é feita a estimativa da pose mais próxima da referência. A proposto conseguiu, para 1000 imagens de teste, um resultado positivo de 93% para as 9 posições do olhar. As taxas de sucesso foram Frente reta: 99% , Cima reto: 92% ,Baixo reto: 77%, Frente esquerda: 88% ,Frente direita: 82% ,cima esquerda: 91%, cima direita: 95%, baixo direita: 62%, baixo esquerda: 54%.

Zuhura e Hossain (2024) apresentam uma solução para o reconhecimento da orientação facial aplicado na web para monitoramento em ambientes virtuais. Utilizando da Tecnologia MediaPipe Facemesh, eles detectam e mapeiam as características e pontos de referências faciais, extraindo assim dados geométricos faciais. Por meio da geometria do rosto, usam o cálculo de Euler para calcular o ângulo de rotação do rosto. Os parâmetros utilizados foram: confiança mínima de 0,3 para a detecção facial e 0,1 para os marcos faciais. Para uma captura eficiente do rosto, é recomendável manter uma iluminação adequada e uma distância entre 2 e 4 pés da câmera. Alcançando uma precisão geral de 86,5%, sendo o sistema proposto validado com dados reais. O sistema de videomonitoramento da orientação do olhar para web final proposto apresenta um desempenho confiável para aplicação real, capaz de lidar com variantes como iluminação, a posição da pessoa frente à câmera e tamanhos de janelas.

Em (MUKHERJEE; ROBERTSON, 2015), propõe-se um método de aprendizagem de máquina capaz de estimar a posição do olhar sem restrições ou seja, em todas as posições possíveis em imagens RGB e RGB-D de baixa resolução, explorando o domínio da vigilância visual e da interação humano-computador (HCI). O método visa unificar esses dois domínios, demonstrando que é capaz de capturar sinais sociais em imagens de

baixa resolução e em tempo real, como gestos comunicativos, foco de atenção, detecção de grupos e comportamento de multidões, avaliando a interação pessoa-pessoa e pessoa-cena por meio de métricas. Foram utilizados dois modelos: um classificador para estimar a posição do olhar em imagens RGB e um SVM para realizar a regressão em imagens RGB-D, com o objetivo de estimar a confiança aproximada dessa regressão. Conclui-se que o método alcançou desempenho comparável a técnicas clássicas, como a Random Forest, apresentando erro médio de 12,35°. Em cenários mais desafiadores, observou-se que a qualidade dos dados de profundidade influencia significativamente o desempenho. Ao final, autor sugere que métodos avançados de reconstrução de profundidade podem aprimorar ainda mais os resultados.

O artigo (GEORGE; ROUTRAY, 2016) apresenta um método rápido para estimar a direção do olhar de uma pessoa em uma webcam, sem a necessidade de equipamento caro. Os sistemas encontram o rosto na imagem, após isso identifica onde estão os olhos usando duas maneira: com base na posição do rosto e o outro procurando pontos-chave do rosto. Depois, usa uma rede neural para classificar a direção do olhar, analisando cada olho separadamente e juntando os resultados. Foi testado com um conjunto de imagens, o método acertou quase 90% das vezes quando tinha que escolher entre 7 direções, e mais de 98% em 3 direções (esquerda, centro e direita), melhor que outros métodos. Rápido funcionamento, lida bem com imagens com ruídos ou borradas, sendo ideal para usar em câmeras comuns. Entretanto, possue o problema de ser menos preciso para olhar para cima ou para baixo, mas os autores sugerem que usar mais dados e melhorias pode ajudar nisso.

Em (SHAH et al., 2022) é abordado o tema da necessidade de segurança dos motoristas, existem sistemas de assistência capazes de notificar o motorista de situações de risco. Em propõe, um novo sistema capaz de notificar situações de risco baseado da classificação da direção do olhar, dependendo do valor de atenção obtido o motorista deve ser notificado para situações de risco. Seu método utiliza uma adaptação do algoritmo de detecção de objetos You Only Look Once (YOLO), mais especificamente a versão YOLO-V4, substituindo modificando o modelo para uma CNN Inception-V3, com intuito de deixar a extração mais robusta na detecção do rosto, prosseguindo com o uso do modelo InceptionResNet-V2 substituindo sua camada de classificação por regressão, aplicando ajude fino para fazer a regressão da região recortada da detecção com a saída contendo a posição do rosto e olhos. Os seus resultados atingiram uma precisão média de 91%.

Em uma visão geral, os trabalhos levantados demonstram avanços importantes no reconhecimento da orientação facial, apresentando soluções para diferentes problemas em que a estimativa da direção do olhar é aplicada, explorando desde técnicas clássicas, como Haar e filtros de Kalman, até arquiteturas modernas de CNNs, alcançando resultados promissores. Entretanto, suas abordagens apresentam limitações, uma vez que foram

treinadas em um único domínio, portanto, diante de contextos que nunca viram, podem apresentar instabilidade e imprecisão. O trabalho proposto, por outro lado, busca solucionar um problema distinto dos demais levantados, sendo ele a classificação da direção do olhar em vídeos do Exame Cover Test, adotando técnicas de Domain Generalization no treinamento com bases de dados provenientes de diferentes fontes, sendo a avaliação realizada em um dataset diferente, cross-dataset, de modo a analisar o desempenho das arquiteturas no contexto específico de inferência em imagens do exame Cover Test, não vistas no treinamento.

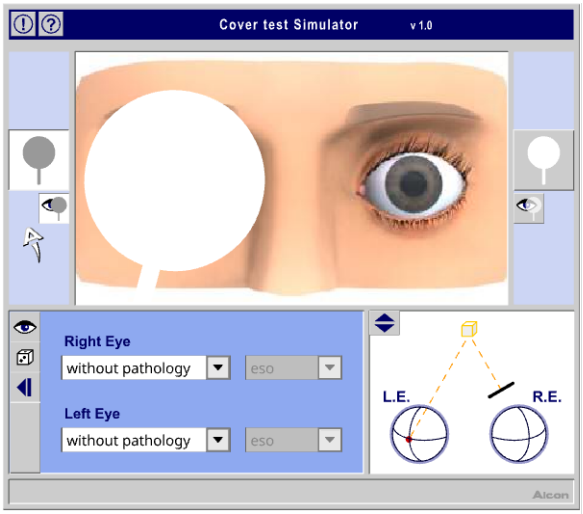
3 Fundamentação Teórica

Este capítulo aborda os conceitos fundamentais e as técnicas computacionais empregadas no desenvolvimento deste estudo.

3.1 Cover Test

O Cover Test é um exame de estrabismo com o uso de oclusor cujo objetivo é identificar comportamentos anormais no olho contralateral ao que está sendo ocluído. O teste é realizado de forma alternada entre os olhos, com o paciente fixando o olhar em um ponto específico durante todo o procedimento. Caso o olho não ocluído não apresente movimento ao alternar a oclusão, considera-se não haver desvio ocular, ou seja, não há tropia presente. Existem diferentes tipos de desvio ocular, nos casos horizontais, ele pode ser convergente (esotropia), quando um dos olhos se desvia para dentro, ou divergente (exotropia), quando o desvio é para fora. Já os desvios verticais são denominados (hipertropia), quando um dos olhos se eleva, e (hipotropia), quando um dos olhos se posiciona abaixo do nível normal (ALMEIDA, 2013), e cada tipo pode estar associado à disfunção de nervos específicos que inervam os músculos extraoculares correspondentes. (VALENTE, 2017)

Figura 1 – Simulação de um exame Cover Test



Fonte: Óptometrial (2025)

No Cover Test, são avaliadas cinco posições oculares: olhar à esquerda (LEVO), à direita (DEXTRO), para cima (SUPRA), para baixo (INFRA) e em posição primária (PPO). A verificação do alinhamento ocular nessas direções fornece informações relevantes

para o diagnóstico e para o planejamento cirúrgico de estrabismo. Esses dados auxiliam o especialista na escolha do tratamento adequado na elaboração do plano cirúrgico (ALMEIDA et al., 2015), quando necessário estimar os valores específicos da dimensão exata de cirurgia sobre o músculo, e para isso é levado em consideração a posição do olhar.

3.2 Pré-processamento de Imagens

Sendo o pré-processamento uma etapa fundamental no desenvolvimento de modelos supervisionados, uma vez que, por meio dele, é possível influenciar a capacidade de generalização, sendo essa influência positiva nos resultados se aplicada corretamente (KOTSIANTIS; KANELLOPOULOS; PINTELAS, 2006). A normalização faz parte do pré-processamento e, segundo (GOODFELLOW; BENGIO; COURVILLE, 2016), é uma tarefa extremamente necessária, sendo a única a ter esse nível de importância em visão computacional. A normalização dos valores da faixa [0,255] para uma escala menor, como [0,1], ajuda manter o valores uniformes e menores. Sem a normalização os gradientes propagados podem atingir valores altos que podem aumentar ainda mais durante a propagação na rede, ocasionando na falha do aprendizado com explosão de gradiente ou seu sumiço, isso ocorre porque o gradiente que dirá se o peso deve diminuir ou aumentar.

Assim como a normalização, a padronização ajuda a prevenir a explosão e desaparecimento de gradiente, consiste em subtrair a média e dividir pelo desvio padrão dos dados. Essa técnica ajusta a distribuição dos valores para média zero e desvio padrão um, estabilizando as ativações, facilita a propagação equilibrada dos gradientes, resultando em um treinamento mais estável e eficiente.(GOODFELLOW; BENGIO; COURVILLE, 2016)

Data Augmentation é uma técnica de geração de dados sintéticos. Dessa maneira, a *Machine Learning* (ML) generaliza melhor para os dados de amostra por ter uma variação maior na base de dados. Eficaz especialmente quando a base de dados é pequena, essa técnica aplica efeitos diversos para tentar aumentar a variância do que o modelo pode ter que predizer. Alguns desses efeitos são a translação, rotação e redimensionamento(GOODFELLOW; BENGIO; COURVILLE, 2016).

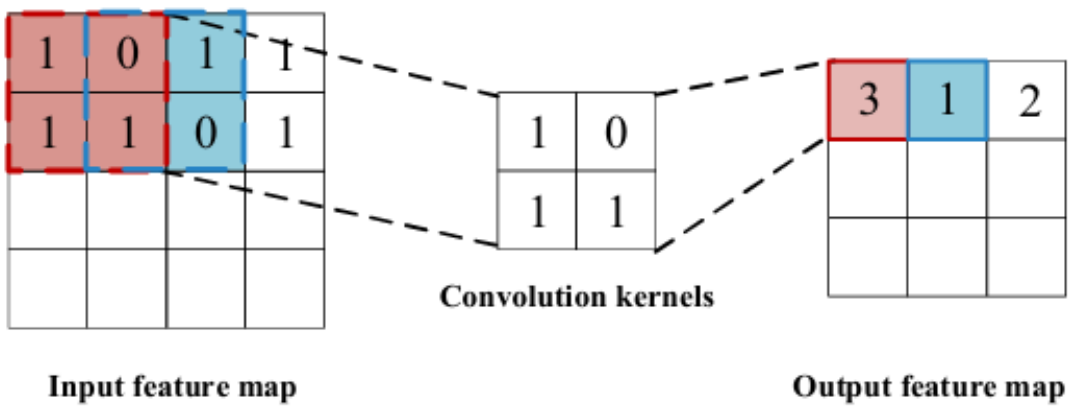
3.3 Redes Neurais Convolucionais

As Redes Neurais Convolucionais, conhecidas como CNNs (Convolutional Neural Networks), são amplamente utilizadas em tarefas de reconhecimento de imagens e séries temporais. Sua principal característica é a utilização da operação de convolução no lugar da multiplicação de matrizes tradicionalmente usada em redes neurais densas. As CNNs são compostas por camadas de convolução, pooling (amostragem) e camadas totalmente

conectadas (fully connected) (YAMASHITA et al., 2018).

A camada de convolução atua aplicando um pequeno filtro chamado kernel sobre a imagem de entrada. Esse kernel, representado como uma matriz que pode ter seu tamanho ajustado, percorre toda a imagem realizando operações de convolução, resultando em um novo mapa bidimensional de características. A cada posição, o kernel aprende a detectar padrões específicos, como bordas ou texturas, os quais são então realçados por meio de funções de ativação, ativando os neurônios conforme a presença de características semelhantes ao padrão aprendido (O'SHEA; NASH, 2015). Na Figura 2 mostra um exemplo de um kernel percorrendo uma entrada:

Figura 2 – Exemplo da operação de convolução



Fonte: Liang et al. (2020)

Os filtros utilizados nas redes convolucionais iniciam a extração de características por meio da identificação de padrões mais simples e concretos, como bordas, contornos e texturas locais. À medida que os dados são processados por camadas sucessivas, essas informações básicas são combinadas de forma hierárquica, permitindo a detecção de formas cada vez mais complexas e abstratas. Esse empilhamento progressivo de filtros possibilita que a rede forme, ao final das iterações, uma representação de alto nível da entrada, adequada para tarefas de reconhecimento de padrões, como classificação ou detecção de objetos.

Na Figura 3, é mostrada a expressão matematica que representa a operação de convolução, onde h_{ij} representa a matriz de saída,o mapa de ativações gerado pela operação na posição (i, j). A função $a[\cdot]$ será a função de ativação que se aplica ao somatório, como a ReLU. O β é o viés adicionado ao resultado da convolução. Já ω_{mn} representa o peso do kernel na posição (m, n), e $x_{i+m-2,\, j+n-2}$ é o valor do pixel da imagem de entrada.

Figura 3 – Representação matemática da operação de convolução

$$h_{ij} = a\left[\beta + \sum_{m=1}^{3}\sum_{n=1}^{3}\omega_{mn}x_{i+m-2,j+n-2}\right]$$

Fonte: Prince (2023)

Existe uma variedade de arquiteturas baseadas em redes convolucionais, entre elas há a AlexNet, que popularizou o uso de CNNs. Na seção seguinte será apresentada essa arquitetura em mais detalhes, demonstrando suas principais características.

3.4 Arquiteturas

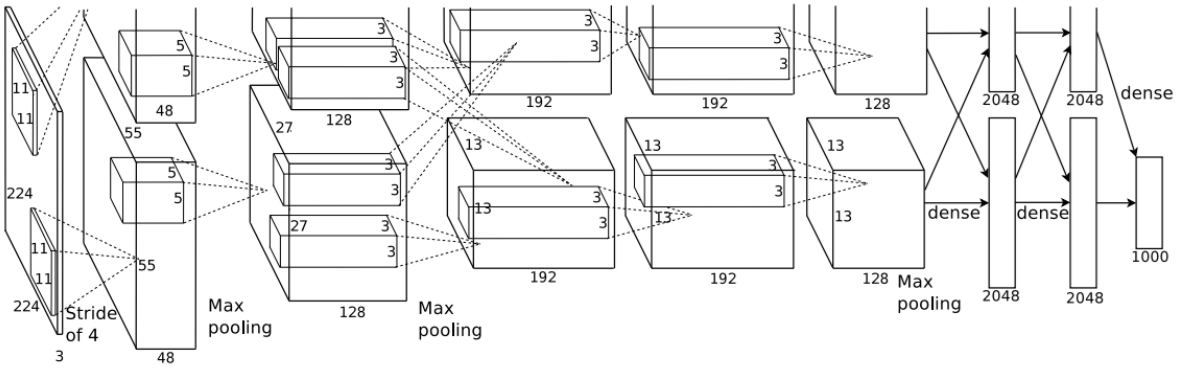
3.4.1 AlexNet

A AlexNet é uma arquitetura de CNN, desenvolvida por Alex Krizhevsky em 2012. Essa rede foi treinada utilizando o conjunto de dados disponibilizado para o ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), uma competição anual voltada ao reconhecimento visual em larga escala.

O conjunto utilizado no ILSVRC corresponde a um subconjunto do ImageNet, uma base de dados que contém mais de 15 milhões de imagens rotuladas, distribuídas em cerca de 22 mil categorias. Para fins da competição, a AlexNet foi treinada especificamente para classificar 1.000 categorias, com aproximadamente 1.000 imagens por categoria.

Ao ser apresentada no ILSVRC de 2012, a AlexNet obteve resultados superiores ao estado da arte da época, alcançando uma taxa de erro top-1 de 37,5% e top-5 de 17,0%, sendo a vencedora da competição.

Figura 4 – Arquitetura AlexNet



Fonte: Krizhevsky, Sutskever e Hinton (2012)

A arquitetura inicia com a convolução de imagens RGB com dimensão $224 \times 224 \times 3$, utilizando 96 filtros de tamanho $11 \times 11 \times 3$ e *stride* 4 na primeira camada. A segunda camada aplica 256 filtros de $5 \times 5 \times 48$. As camadas convolucionais 3, 4 e 5 utilizam filtros 3×3, com 384, 384 e 256 filtros, respectivamente, e operam de forma sequencial, sem camadas intermediárias de *pooling* ou normalização. A saída da última camada totalmente conectada da rede é passada por uma função *softmax* com 1.000 saídas, gerando uma distribuição de probabilidade sobre as 1.000 classes possíveis.

A rede possui 60 milhões de parâmetros e, segundo (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), uma das estratégias para lidar com o *overfitting*, causado pela alta capacidade da rede em relação à quantidade de informação útil extraida da entrada, é o uso de *data augmentation*, como citado na 3.2, além do *dropout*, já implementado nas camadas densas da rede. *Dropout* consiste no desligamento das saídas dos neurônios com probabilidade de 50%, com o objetivo de estimular a rede a extrair características mais complexas ao forçar novos caminhos entre neurônios ativados aleatoriamente. Krizhevsky, Sutskever e Hinton (2012) reforça que, sem o uso de *dropout*, a rede sofre de *overfitting*.

Há diferentes maneiras de se utilizar a convolução, seja somente utilizando arquiteturas completamente convolucionais ou combinando-a com técnicas baseadas em Transformers, obtendo-se modelos híbridos. Na seção seguinte serão apresentadas outras oito arquiteturas utilizadas neste estudo.

3.4.2 MaxVit

A MaxVit é uma abordagem hibrida, apresentada em (TU et al., 2022), que combina dois domínios Convolução e Transformers. Ela possui uma estrutura simples baseada na repetição de blocos que integram convolução e atenção local e global, no que diz respeito às camadas convolucionais ela utiliza blocos MBConv (Mobile Inverted Bottleneck Convolution) e módulos Squeeze-and-Excitation, são blocos que atuam nos canais capaz de recalibrar recurso, ele faz um ajuste inteligente reforçando canais importantes e enfraquece os menos uteis baseado na informação global, tornando a rede mais eficiente e com um poder de representação maior (HU; SHEN; SUN, 2018).

Capaz de capturar informações globais na primeira camada, ao contrário do que ocorre comumente em arquiteturas convolucionais puras, graças ao mecanismo de Atenção global dilatada (grid attention). Além disso, possui uma eficiência com baixo custo computacional, uma vez que utiliza o mecanismo de atenção local que se restringe a janelas pequenas, para capturar as informações mais relevantes, semelhante ao que ocorre com o Swin Transformers. Para a tarefa de classificação do olhar pode ser útil, ao ser uma arquitetura robusta, mas que não abdica da eficiência e baixo custo computacional, a abordagem de captura global nas primeiras camadas podem oferecer um desempenho positivo.

3.4.3 ConvNext-T

ConvNeXt Tiny tem como base a arquitetura ConvNet puramente convolucional, usando uma reinterpretação das arquiteturas Transformer, com sua estruturação hierárquica, Swin Transformer, mostrando que as arquiteturas convolucionais ainda têm muito a oferecer em contraste com as novas Transformers. Sendo uma modernização da ResNet, é construída especialmente com módulos padrão de redes convolucionais. Foi redesenhada para substituir a função de ativação *Rectified Linear Unit* (ReLU) por *Gaussian Error Linear Unit* (GELU), além de reduzir a quantidade de funções de ativação o que caracteriza os Transformers , utilizando somente uma única ativação GELU em cada camada, e substituindo a normalização em batch (Batch Normalization) por normalização de camada (Layer Normalization). As camadas de subamostragem (Downsampling) separadas foram adaptadas para reproduzir o método de usar camadas convolucionais 2×2 com passo(Stride) 2 para subamostragem espacial, com uma camada de normalização (LN) antes de cada subamostragem. (LIU et al., 2022).

Assim como a MaxViT, proposta por Tu et al. (2022), oferece uma abordagem atípica das demais, sendo uma arquitetura que buscou explorar ao máximo o potencial das redes convolucionais puras com novas técnicas derivadas dos Transformers, segundo Liu et al. (2022) a arquitetura alcançou 87,8% de precisão top-1 no ImageNet.

3.4.4 Inception-v3

Em (SZEGEDY et al., 2015), a Inception-v3 é a versão mais avançada da arquitetura Inception, resultante de uma série de melhorias. Sua base é a Inception-v2, que já incorpora a fatoração da convolução 7×7 em três convoluções 3×3. As otimizações que definem a Inception-v3 incluem o uso do otimizador RMSProp, a regularização por suavização de rótulos (*label smoothing*) para reduzir a superconfiança do modelo, a fatoração de convoluções grandes em operações menores, que entrega mais eficiência, e a aplicação de Batch Normalization aos classificadores auxiliares.

A Inception-v3 oferece uma abordagem diferenciada, especialmente no uso de label smoothing e na fatoração de convoluções, características que a distinguem das demais arquiteturas utilizadas.

3.4.5 EfficientNetV2-S

De acordo com (TAN; LE, 2021), as EfficienteNet tem a proposta de oferecer um treinamento mais eficiente com menos parâmetros e mais velocidade de treinamento somente usando combinações de blocos MBConv (Mobile Inverted Bottleneck Convolution) e Fused-MBConv, usam kernels menores, 1×1 e 3×3 respectivamente. Tan e Le (2021) afirma que manter a mesma regularização para imagens cada vez menores entrega

resultados insatisfatórios na precisão, então propõe uma abordagem diferente capaz de acelerar o treinamento que consiste em usar uma transição de regularização fraca e imagens pequenas para regularização forte com aumento gradativo das imagens, entregando um treinamento mais acelerado sem perder a precisão, diferente do método anterior.

3.4.6 DenseNet

Em (HUANG et al., 2018) é apresentada a DenseNet, uma arquitetura de rede convolucional que alimenta cada camada com os mapas de características de todas as camadas anteriores, de forma sucessiva, seguindo o fluxo tradicional de redes feed-forward. Resultando em $\frac{L(L+1)}{2}$ conexões em uma rede com L camadas. Em vez de somar os mapas de características como nas ResNets, a DenseNet os concatena, permitindo que cada camada reutilize diretamente as saídas de todas as anteriores. A arquitetura é organizada em blocos densos intercalados com camadas de transição que realizam redução de resolução (downsampling), e utiliza camadas de gargalo, ou seja camadas que primeiro aplicam uma convolução menor de 1×1, em seguida uma maior de 3×3 por exemplo, com um custo computacional menor. Essa arquitetura busca explorar a ideia de que encurtar as conexões próximas as entradas pode oferecer uma melhora na eficiência do treinamento.

3.4.7 GoogLeNet

De acordo com (SZEGEDY et al., 2014), a GoogLeNet é uma arquitetura Inception, pensada em praticidade e eficiência computacional, a rede foi projetada para rodar até mesmo em dispositivos com recursos limitados, especialmente com baixo consumo de memória. Ela aplica reduções com filtros 1×1 que antecedem as convoluções 3×3 e 5×5, seguidas por camadas de pooling e ativação ReLU. Ela tem sua camada totalmente conectada substituída por *pooling* médio, faz uso de *dropout* de 70% semelhante a AlexNet discutida na seção 3.4.1 que aplica 50%.

3.4.8 EfficientNet B0

Apresentada em (TAN; LE, 2020) a EfficienteNet B0 é uma arquitetura que foi pensada na possibilidade de balancear profundidade, largura e resolução. A solução encontrada foi um método que escala uniformemente os três, por meio de coeficientes de escala fixo, aumentando a profundidade da rede em N, a largura em N e o tamanho em N, N sendo um coeficiente fixo, se tem um uso 2N de recursos computacionais, por meio desse método a rede pode ser escalada de forma composta. Sua arquitetura consiste em blocos MBConv (Mobile Inverted Bottleneck), com otimização squeeze-and-excitation, bloco que foi explicado na seção 3.4.2.

3.4.9 MNASNET_A1

MNASNET_A1, apresentada em (TAN et al., 2019) como resultado do experimento, é uma arquitetura com a proposta de ser leve o suficiente para ser usada em dispositivos com recursos limitados. Foi criada a partir da MNAS (Mobile Neural Architecture Search), um método de busca automática, ou seja, um algoritmo de busca que explora diferentes arquiteturas de redes neurais, levando em conta latência real e acurácia, para encontrar modelos ideais para dispositivos móveis. Ela utiliza tanto convoluções 3×3 quanto 5×5 em diferentes partes da rede, sendo essa uma de suas características que a diferenciam de outros modelos, os quais usam apenas 3×3.

3.5 Aprendizado por Transferência

De acordo com (CAO et al., 2010) Transfer Learning é uma técnica que visa usar o conhecimento aprendido de um modelo pre-treinado para melhorar o desempenho em uma nova tarefa específica, tarefa essa que é preciso ter similaridade com a tarefa original para qual foi treinada a Rede para não ocasionar numa transferência negativa prejudicando o desempenho da aprendizagem para a nova tarefa. Uma das estratégias do Aprendizado por transferência é o Ajuste Fino (Fine Turning).

O fine-tuning é uma estratégia amplamente utilizada, especialmente em situações em que se dispõe de uma base de dados pequena. Utilizando um modelo pré-treinado e ajustando apenas algumas camadas, sendo esse ajuste o descongelamento ou congelamento, obtém-se uma rede neural capaz de generalizar melhor para a tarefa-alvo para qual se planeja treinar, com destaque para aplicações na área de imagem medicas. (VRBANČIČ; PODGORELEC, 2020). Os dois principais tipos de ajuste fino são o superficial e o profundo.(SETHI; ARORA; SUSAN, 2020)

O Ajuste Fino Superficial *Shallow Fine-turning* é quando se descongela somente a camada de classificação chamadas Top layers, composta por camadas totalmente conectadas *(fully connected/dense layers)*que contribui paras as características de alto nível combinando as representações extraídas das camadas de features, nesse sentindo, para (SETHI; ARORA; SUSAN, 2020) é semelhante com o procedimento de extração de características, que utiliza o conhecimento da rede pré-treinada e substitui a camada de classificação por uma SVM para treinamento nesse novo classificador. De acordo com Vrbančič e Podgorelec (2020) não recomendado quando a tarefa-alvo tem pouca similidade com o que o modelo foi treinado.

A escolha de qual camada descongelar não é padronizada, é descoberto de forma empírica qual traz um bom resultado para o treinamento, quando a tarefa-alvo tem pouca similaridade com o que o modelo foi treinado o ideal é utilizar o Ajuste fino profundo (Deep Fine-Tuning) o qual é o descongelamento de 2 ou mais camadas, incluindo as

intermediárias ou iniciais.

O descongelamento progressivo,"scheduled unfreezing"ou "Progressive Unfreezing.", é uma técnica de ajuste fino progressivo. Consiste em iniciar o treinamento com todas as camadas congeladas, e ir descongelando-as gradativamente ao longo do treinamento por meio de um agendamento que usa o hiperparâmetro de Época como condicional (LI et al., 2024).A técnica evita que os pesos pré-treinados sejam alterados bruscamente no início do treinamento, incentivando a rede a adaptar-se de forma incremental para a nova tarefa.

3.6 Técnicas de regularização

Durante o treinamento de um modelo, pode ser adotada algumas técnicas capazes de otimizar o desempenho e trazer resultados melhores como por exemplo o Early Stopping.

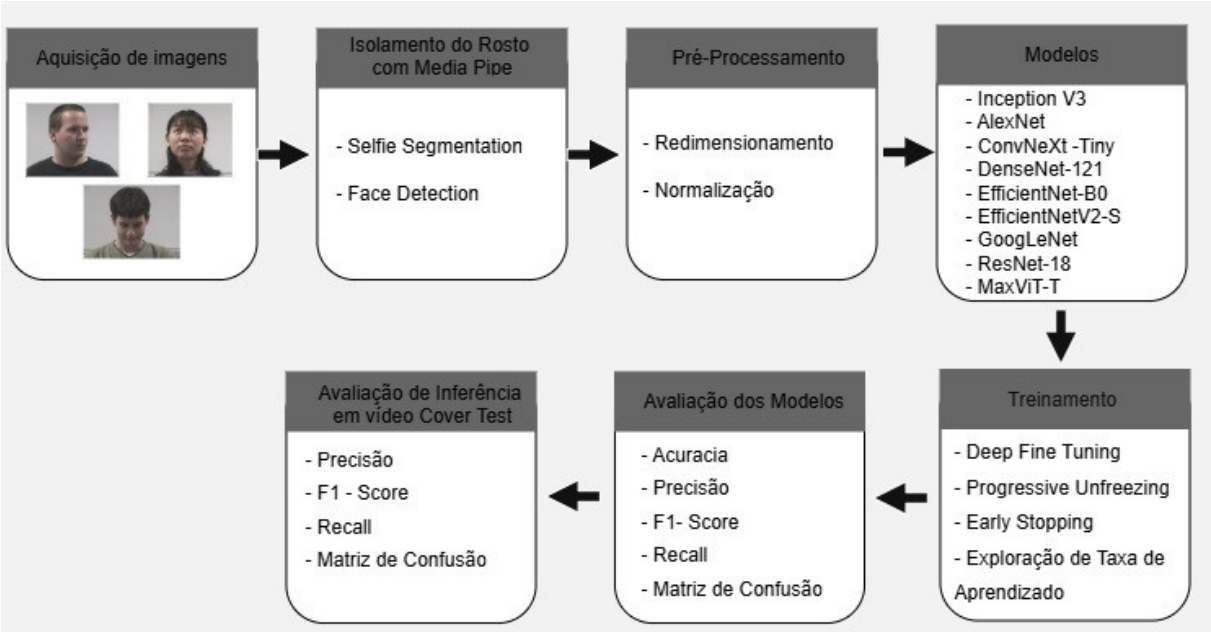
Early Stopping é uma técnica de parada antecipada comumente utilizada para minimizar problemas de overfitting. A ideia central é interromper o treinamento quando a rede já aprendeu as características e padrões gerais dos dados de treinamento e começa a memorizar padrões específicos demais. Nesse ponto, há risco de a rede perder sua capacidade de generalização e se ajustar excessivamente aos dados de treino.

A técnica atua interrompendo o treinamento com base em um critério definido sobre o desempenho em um conjunto de validação, geralmente usando uma métrica como a função *loss*. Quando essa métrica deixa de melhorar, ou começa a piorar, por um número consecutivo de épocas definido previamente por um parâmetro de tolerância (*patience*), o treinamento é interrompido automaticamente. Assim, busca-se preservar a capacidade preditiva máxima do modelo. Em outras palavras, manter o treinamento após esse ponto poderia degradar o desempenho que o modelo teria alcançado. Bartlett et al. (2023).

4 Metodologia

Neste capítulo, é descrita a metodologia usada na pesquisa, desde a aquisição da base de dados ao pré-processamento, descrição das arquiteturas selecionadas, experimentos conduzidos e a avaliação. Na Figura 5 está representado o fluxo metodológico do estudo.

Figura 5 – Etapas da metodologia.



Fonte: acervo do autor

4.1 Aquisição de Imagens

O conjunto de imagens utilizado na pesquisa é oriundo de 4 fontes distintas. Elas podem apresentar qualidade e angulação diferentes devido a divergência do protocolo de aquisição definido para cada um dos datasets. As 4 bases de dados são a Head Pose Image DataBase, Biwi Kinect Head Pose Database, Voluntários VipLab e base Cover Test. Cada uma dessas fontes serão descritas em detalhes abaixo.

4.1.1 Head Pose Image Database

Foram utilizadas as imagens da base de dados "Head Pose Image Database", obtidas do trabalho de Gourier, Hall e Crowley (2004). Composto por 2.790 imagens de rosto monocular, obtidas a partir de 15 indivíduos sob variações de inclinação (tilt) e rotação lateral (pan) entre $-90°$ e $+90°$. Para cada indivíduo, estão disponíveis duas

séries contendo 93 imagens com diferentes poses, permitindo a avaliação de algoritmos tanto em faces conhecidas quanto desconhecidas. A base apresenta diversidade em termos de uso de óculos e tonalidade de pele, enquanto o fundo das imagens é neutro e livre de elementos visuais, visando preservar o foco nas operações faciais.

Figura 6 – Exemplos de imagens da Base Head Pose Image Database

(a) DEXTRO **(b)** SUPRA **(c)** INFRA

Fonte: Gourier, Hall e Crowley (2004)

4.1.2 Biwi Kinect Head Pose Database

O segundo conjunto de dados utilizado neste trabalho foi obtido do trabalho de Fanelli et al. (2013) sendo composto por mais de 15.000 imagens de 20 indivíduos (14 do sexo masculino e 6 do sexo feminino), sendo que 4 participantes foram registrados em dois ambientes diferentes. As imagens foram adquiridas com um sensor Kinect posicionado a aproximadamente um metro de distância, enquanto os indivíduos realizavam movimentos livres de cabeça, cobrindo uma ampla faixa de ângulos de rotação (yaw) e inclinação (pitch), variando aproximadamente de $\pm 75°$ em *yaw* e $\pm 60°$ em *pitch*. Cada quadro da sequência é composto por uma imagem RGB e uma imagem de profundidade, ambas com resolução de 640×480 pixels, além de anotações automáticas obtidas por meio do sistema da plataforma FaceShift, uma tecnologia de captura de movimento, descontinuada em 2015 e adquirida pela Apple. As anotações incluem a posição tridimensional do centro da cabeça e os ângulos de rotação correspondentes, fornecendo o ground truth necessário para avaliação de sistemas de estimação de pose em tempo real. Embora o método de aquisição se baseie em estimação por quadro, e não em rastreamento contínuo.

Figura 7 – Exemplos de imagens da Base Biwi Kinect.

(a) PPO **(b)** LEVO **(c)** DEXTRO

Fonte: Fanelli et al. (2013)

4.1.3 Voluntários VipLab

O terceiro conjunto de dados, esse sendo do acervo do autor, foi obtido usando a câmera do celular Xiaomi Poco M5S, na horizontal com resolução de 4624 x 2080, com o voluntário sentado e o celular em uma altura de 1,10 metros. Foram 8 voluntários sob variações de inclinação (*tilt*) e rotação lateral (*pan*) entre $-90°$ e $+90°$ em fundo branco, gerando assim 80 imagens.

Figura 8 – Exemplos de imagens da Base VipLab

(a) DEXTRO **(b)** INFRA **(c)** LEVO

Fonte: Acervo do autor

4.1.4 Base Cover Test

A base utilizada é composta por vídeos de 9 pessoas voluntárias com 35 imagens de cada pose, porém é possível extrair variações de cada pose para graus menores de translação ou com elementos. Para cada pose, estipulam-se 7 variações de poses extraídos frame a frame, logo temos 5 poses $\times$ 7 variações $\times$ 9 pessoas resultando em 315 amostras de frames. A aquisição foi realizada em um consultório por uma profissional, com a devida autorização dos voluntários para uso acadêmico. O procedimento consistiu no posicionamento frontal fixo da câmera, na horizontal, com o voluntário centralizado e

sentado, olhando fixamente para a câmera durante as translações da cabeça entre as cinco posições. Em cada uma dessas posições, foi realizado o procedimento de oclusão, alternando entre os olhos.

Figura 9 – Exemplos de imagens da Base Cover Test

(a) INFRA **(b)** DEXTRO **(c)** LEVO

Fonte: Acervo do autor

4.2 Detecção da Face

Com o objetivo de melhorar o foco da rede, foi utilizado, antes do pré-processamento, o framework MediaPipe (LUGARESI et al., 2019), que oferece um conjunto de soluções, bibliotecas e ferramentas para detectar face. Inicialmente, utilizou-se o módulo Selfie Segmentation disponível no MediaPipe para remover o fundo das imagens por meio da segmentação corporal, utilizando sua máscara booleana resultante, se obtêm uma segmentação composta por valores que se > 0.5 será True (indica que há pessoa), e que se for < 0.5 será igual a False (indica que não há pessoa), por meio da imagem original é criado um fundo preto com suas mesmas dimensões, nele é feito um mapeamento da imagem original usando os valores Booleanos da segmentação, assim, fazendo uma cópia somente da figura humana para a imagem preta. Em seguida, empregou-se o módulo Face Detection para recortar a região da face. Esse procedimento visa eliminar elementos de distração no fundo das imagens, uma vez que as amostras apresentam variações significativas desses elementos circundado o objeto alvo que é o rosto, o que poderia comprometer o processo de aprendizagem da rede.

O módulo de Selfie Segmentation do MediaPipe utiliza uma rede neural convolucional (CNN) leve, baseada na arquitetura MobileNetV3, projetada para aplicações em tempo real. O modelo foi treinado para classificar cada pixel da imagem em duas categorias: plano de fundo (índice 0) e pessoa (índice 1). Ele opera com entradas de 256x256 pixels ou 144x256 pixels, dependendo do modo de uso selecionado, o índice [0] é ideal para rostos próximos (selfies) e o índice [1] para capturar planos mais amplos, como o corpo inteiro. A imagem de entrada deve ser previamente convertida de BGR para RGB, pois o modelo espera a representação em RGB. Após o pré-processamento, o modelo re-

aliza a segmentação da imagem, gerando uma máscara probabilística, na qual cada valor representa a probabilidade de o pixel pertencer a uma pessoa. (GOOGLE, 2023).

O módulo Face Detection é uma ferramenta baseada no BlazeFace, proposto em (BAZAREVSKY et al., 2019) como um detector de rostos leve que apresenta bom desempenho em aplicações em tempo real, sendo útil em contextos de realidade aumentada, como na segmentação de regiões faciais. A saída do modelo consiste em uma caixa delimitadora (bounding box) que envolve o rosto, além de seis pontos-chave: os dois olhos, ponta do nariz, centro da boca e orelhas, todos normalizados em relação às dimensões da imagem. Durante a inferência, adota-se um limiar de confiança de 0,5 para uma região ser considerada como contendo a presença de um rosto humano, é possível a ajustar a confiança durante a inferência. (GOOGLE, 2023)

4.3 Pré-processamento

A etapa de pré-processamento é essencial para garantir o desempenho satisfatório das arquiteturas que serão avaliadas descritas na Seção 4.2. Nessa fase, realiza-se a padronização das imagens, especialmente por serem obtidas por procedimentos distintos. Para a maioria das redes, as imagens das quatro bases de dados são redimensionadas para a escala de 224×224 pixels, com exceção da InceptionV3, que requer imagens com dimensões de 229×229 pixels como entrada.

- Padronização (*Standardization*): Conversão das imagens para tensores, etapa necessária para que os dados possam ser processados pela rede neural;
- Normalização (*Normalize*): Normalização com média e desvio padrão calculados anteriormente, assegurando que os dados tenham distribuição próxima de média zero e variância unitária.

4.4 Classificação da posição do Olhar

A tarefa de classificação do olhar é um problema de classificação multiclasse, com cinco categorias correspondentes às posições presentes no exame Cover Test: DEXTRO, LEVO, SUPRA, INFRA e PPO, devido a isto, adotou-se a função CrossEntropyLoss para estimar a probabilidade entre as classes. Para otimização utilizou-se o Adam (*Adaptive Moment Estimation*) devido a sua eficiência e seu grande poder de convergência, uma vez que é capaz de fazer adaptações na taxa de learning rate de maneira individual por meio de dois momentos, observando a direção e intensidade do gradiente (KINGMA; BA, 2017).

Foram selecionadas diferentes tipos de Arquiteturas pré-treinadas para a avaliação, desde arquiteturas robustas a mais simples. As saídas dessas redes foram ajustadas para o problema proposto de classificação em cinco classes. Na Tabela 1 é listado do maior número de parâmetros ao menor para evidenciar a complexidade de cada uma.

Tabela 1 – Quantidade de parâmetros por modelo.

Modelo	**Total de Parâmetros**
AlexNet	57,024,325
MaxViT	30,410,189
ConvNeXt Tiny	27,823,973
Inception V3	25,122,509
EfficientNet V2 S	20,183,893
DenseNet	6,958,981
GoogLeNet	5,605,029
EfficientNet B0	4,013,953
MnasNet	3,108,717

Para o acompanhamento das etapas de treinamento, validação e teste foram utilizadas as métricas Acurácia, Precisão, Recall e F1-Score e Matriz de Confusão disponibilizadas pela biblioteca scikit-learn. Na seção seguinte serão explorados a demais técnicas de preparação.

4.5 Preparação dos Experimentos

As quatro bases de dados foram organizadas e atribuídas a etapas distintas do experimento, de modo a evitar vazamento de amostras entre as fases. As bases Head Position Estimation, 4/15 do total da Biwi Kinect e VipLab foram utilizadas para treinamento. As demais amostras restantes da base Biwi Kinect foram atribuídas à etapa de validação, enquanto a base Cover Test foi reservada para a avaliação cross-dataset final dos modelos treinados.

A quantidade de amostras por classe alcançada foi: 25 amostras para treinamento, 16 amostras para validação e, para teste, incrementando a base com intervalos das poses, conforme explicado na Seção 4.1.4, 47 amostras por classe.

Para a preparação da base Cover Test que será usada na fase de Teste, foram extraídos frames chaves dos vídeos, em torno de 7 variações para cada pose. Totalizando como descrito na Seção 3.1, resultando em 315 amostras. Os frames foram separados por classes, mantendo os voluntários agrupados dentro de cada classe.

As amostras são processadas antes de serem utilizadas como entrada para as redes; por conseguinte, empregamos a biblioteca Media Pipe mencionada na Seção 4.2, inicialmente recorrendo ao módulo Selfie Segmentation para eliminar o fundo da imagem.

Dessa forma, facilitamos o isolamento do rosto, utilizando o módulo Face Detection. As amostras tratadas foram salvas e organizadas em classes, conforme a disposição original.

Antes do início do treinamento, todos os modelos tiveram suas camadas congeladas, exceto a camada final de classificação, substituída por uma nova camada totalmente conectada adaptada para a tarefa de classificação multiclasse com cinco classes. Em seguida, foi adotada uma estratégia de descongelamento progressivo das camadas, buscando ajustar gradualmente os pesos das camadas convolucionais conforme a complexidade de cada arquitetura. Procurou-se manter uma proporção justa e equilibrada na quantidade de camadas descongeladas para cada modelo, respeitando suas particularidades estruturais e profundidade. Essa abordagem visa favorecer a capacidade de generalização dos modelos, reduzindo o risco de overfitting, especialmente diante de um conjunto de dados limitado.

Foram usadas quatro Learning Rate (LR) diferentes na avaliação das arquiteturas para avaliar com qual LR ela se sairia melhor na faixa de 1e-2 a 1e-5, semelhante ao método Grid Search por meio de exaustão, o que significa que é uma maneira de encontrar os melhores valores de configuração, escolhem-se diferentes valores de hiperparâmetros e ele testará deferentes combinações para cada um deles, avaliando o desempenho, em nosso caso é somente um. Para evitar overfitting também foi utilizada a técnica de (*Early stopping*) com o limite de paciência (*patience*) em 50 Epochs, finalizando quando a Loss apresentar estagnação no treinamento ou a perda tiver um aumento > 0.7 que possa prejudicando o progresso anterior. Foi adotado um tamanho de Lote (batch) igual a 1 para evitar o Overffiting, ou seja, teremos 1 imagem por iteração.

Visando diversificar a base de treino e melhorar a capacidade de generalização dos modelos avaliados, foram aplicadas as seguintes técnicas de Data Augmentation ao conjunto de treinamento:

- Rotação Aleatória (*RandomRotation*): rotação aleatória limitada a 3 graus, promovendo leve variação angular das imagens
- Corte Redimensionado Aleatório (*RandomResizedCrop*): redimensionamento e recorte aleatório, com variação de escala entre 90% e 110% do tamanho original, simulando um leve zoom in e zoom out;
- Contraste (*ColorJitter*): ajustes aleatórios de brilho e contraste, através da técnica de color jitter, visando adaptar o modelo a diferentes condições de iluminação;

4.6 Avaliação

A avaliação das arquiteturas foi realizada por meio de treinamentos conduzidos com diferentes hiperparâmetros previamente estabelecidos, gerando um modelo final para testar o desempenho na tarefa-alvo, que consiste na classificação da base de dados Cover Test.

O monitoramento do desempenho durante a validação foi feito utilizando métricas como acurácia, precisão, F1-Score, recall e loss, além da matriz de confusão, que permitiu analisar quais classes estavam sendo confundidas entre si ou se o modelo apresentava uma tendência a predizer excessivamente uma determinada classe.

Recall ou sensibilidade (Equação 4.1) informa a proporcionalidade de quantas posições reais do rosto foram corretamente previstas. Acurácia (Equação 4.3) foca em medir a proporção de previsões corretas da pose, tanto positivo quanto negativo, em relação ao número total de amostras. Precisão (Equação 4.2) mede a proporção de exemplos realmente positivos entre todas as imagens nas quais o modelo classificou como positivos. Já o F1-Score (Equação 4.4) representa a média harmônica entre a precisão e o recall, sendo uma métrica que busca equilibrar ambos os valores e penaliza quando um dos dois é muito baixo.

Todas essas métricas são calculadas a partir da contagem de True Positive(TP), False Positive(FP), True Negative(TN), False Negative(FN).

$$\text{Sensibilidade (SEN)} = \frac{TP}{TP + FN} \tag{4.1}$$

$$\text{Precisão (PRE)} = \frac{TP}{TP + FP} \tag{4.2}$$

$$\text{Acurácia (ACC)} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.3}$$

$$\text{F1-Score (F1)} = \frac{2TP}{2TP + FP + FN} \tag{4.4}$$

Para selecionar o melhor desempenho da rede, foi utilizado um esquema de checkpoint, salvando o modelo correspondente à época com a melhor Loss durante o treinamento. A avaliação final do modelo na base de teste foi julgada com base nas mesmas métricas empregadas na fase de validação.

5 Resultados

Nesta seção, são apresentados os resultados obtidos a partir dos experimentos de treinamento e teste realizados com nove diferentes modelos. Os modelos foram avaliados e ranqueados, destacando-se os cinco com melhor desempenho, sendo evidenciado aquele que obteve os melhores resultados na predição, aplicado à base de teste referente ao Cover Test, descrita na Seção 4.1.4.

5.1 Experimentos com Quatro Configurações de Taxa de Aprendizado

Nesta condição de avaliação, foram realizados quatro experimentos com taxas de aprendizado (LR) diferentes, no intervalo de 1×10^{-2} a 1×10^{-5} sem descongelamento progressivo e somente com o descongelamento das duas últimas camadas da rede, além das camadas de Classificação.

Para assegurar a equidade no treinamento, realizaram-se os descongelamentos de duas camadas com base em um critério de descongelamento proporcional à complexidade da arquitetura. Ou seja, arquiteturas mais simples onde se tem somente camadas sequencias de convolução foram descongeladas somente duas camadas, mas em casos onde a arquitetura é composta por blocos, por meio do critério, deve-se descongelar dois blocos, uma vez que o bloco pode ter camadas convolucionais, camadas de normalização e ativação. A menor unidade de uma arquitetura composta por blocos é o próprio bloco. Ou seja, arquiteturas mais simples nas quais se tem apenas camadas sequencias de convolução foram descongeladas somente duas camadas, mas em casos onde a arquitetura é composta por blocos, por meio do critério, deve-se descongelar dois blocos, uma vez que o bloco pode ter camadas convolucionais, normalização e ativação. Assim, considerou-se que a menor unidade de uma arquitetura composta por blocos é o próprio bloco.

Para a análise comparativa do desempenho de cada modelo, foram extraídos e avaliados gráficos correspondentes às métricas de Precisão, Acurácia, F1-Score, Recall e à função de perda (*loss*).

Na Tabela 2 são comparados nove modelos, abrangendo desde arquiteturas mais simples até arquiteturas mais complexas e robustas. A principal métrica utilizada para o ranqueamento dos melhores modelos foi a função de perda (Loss), seguida pelas métricas de F1-Score, Recall. Observa-se que da Googlenet à Convnet-Tiny, a primeira no ranque, houve uma melhora de 35% na loss, nota-se também que houve um desempenho semelhante com uma pequena diferença entre os três primeiros, uma média de crescimento de

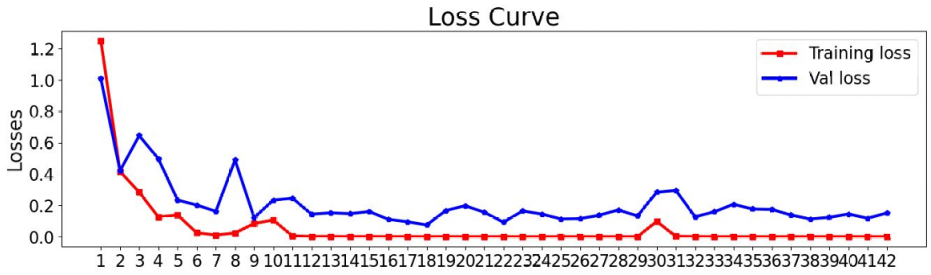
somente 1.16% entre si para as demais métricas.

Tabela 2 – Desempenho dos modelos nas métricas de validação

Modelos	**Val Accuracy**	**Val Precision**	**Val Recall**	**Val F1**	**Val Loss**
Convnet_tiny	0.9885	0.9891	0.9885	0.9885	0.0291
Alexnet	0.9770	0.9783	0.9770	0.9770	0.1442
MaxViT_T	0.9655	0.9675	0.9655	0.9654	0.2013
EfficientNet_V2S	0.8391	0.8560	0.8391	0.8391	0.8522
GoogLeNet	0.7011	0.7737	0.7011	0.7056	1.0565
MnasNet1_0	0.6322	0.7408	0.6322	0.6152	1.0130
Resnet	0.6437	0.7482	0.6437	0.6407	1.0623
EfficientNet_B0	0.6782	0.7826	0.6782	0.6128	1.4276
Densenet	0.2299	0.3631	0.2299	0.1907	1.6296

O gráfico de Loss na Figura 10 mostra que a rede obteve um bom aprendizado para as características de treinamento, estabilizando-se perto de 0. A Loss de validação decresce com o treinamento, possivelmente devido a dados nunca antes utilizados no treinamento. Vale lembrar que foram utilizadas bases de dados distintas para cada fase.

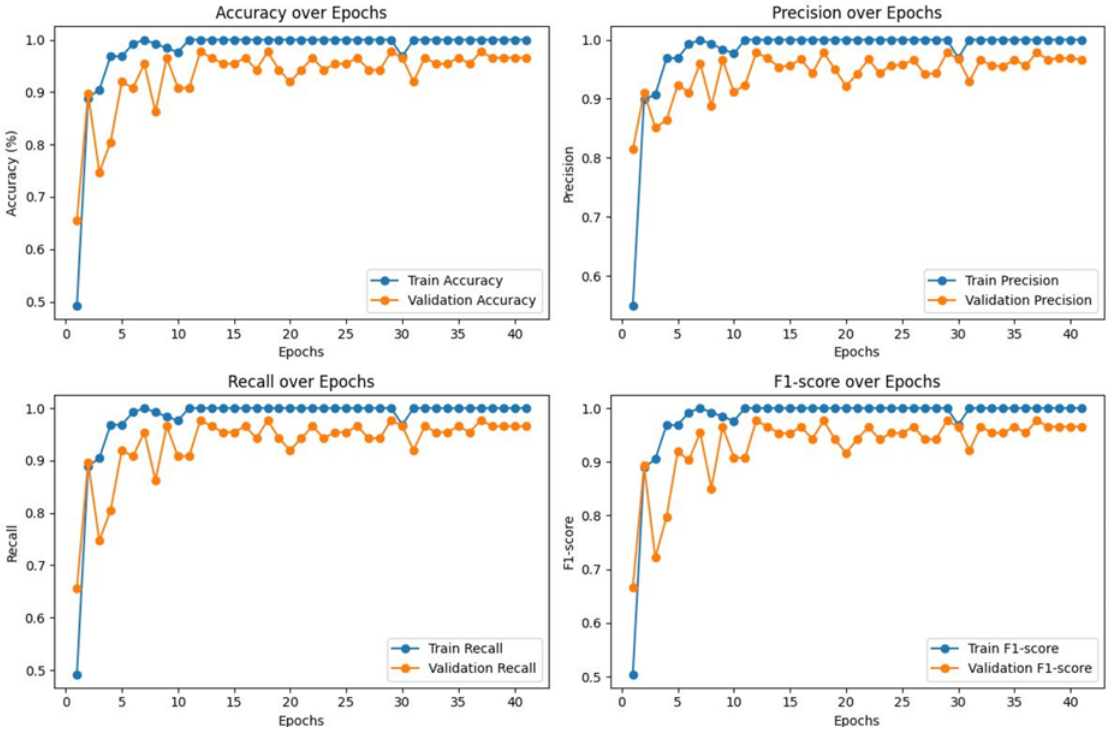
Figura 10 – Gráfico de Loss Alexnet - LR Fixa



Fonte: acervo do autor

Ao examinar as métricas apresentadas na Figura 11, observa-se que a rede exibe um desempenho notável, com leves oscilações na validação, alcançando a marca de 90% até a décima época, momento em que se estabiliza. Em geral, as curvas são estáveis sem overffiting.

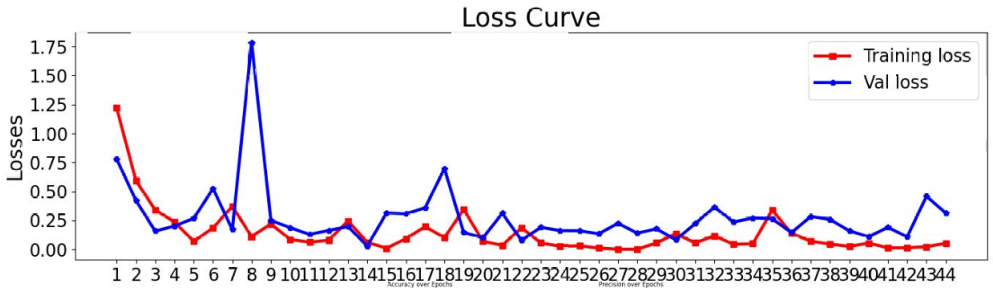
Figura 11 – Gráfico do desempenho da rede Alexnet - LR Fixa



Fonte: acervo do autor

Na Figura 12 podemos observar a Loss da arquitetura ConvNext-Tiny. Ela exibe uma diminuição gradual da perda de treinamento, com algumas oscilações, enquanto a validação se mantém em níveis reduzidos, embora com alguns picos que podem ser atribuídos à variação das amostras ou à sensibilidade da rede, mas permanecem dentro de um patamar aceitável.

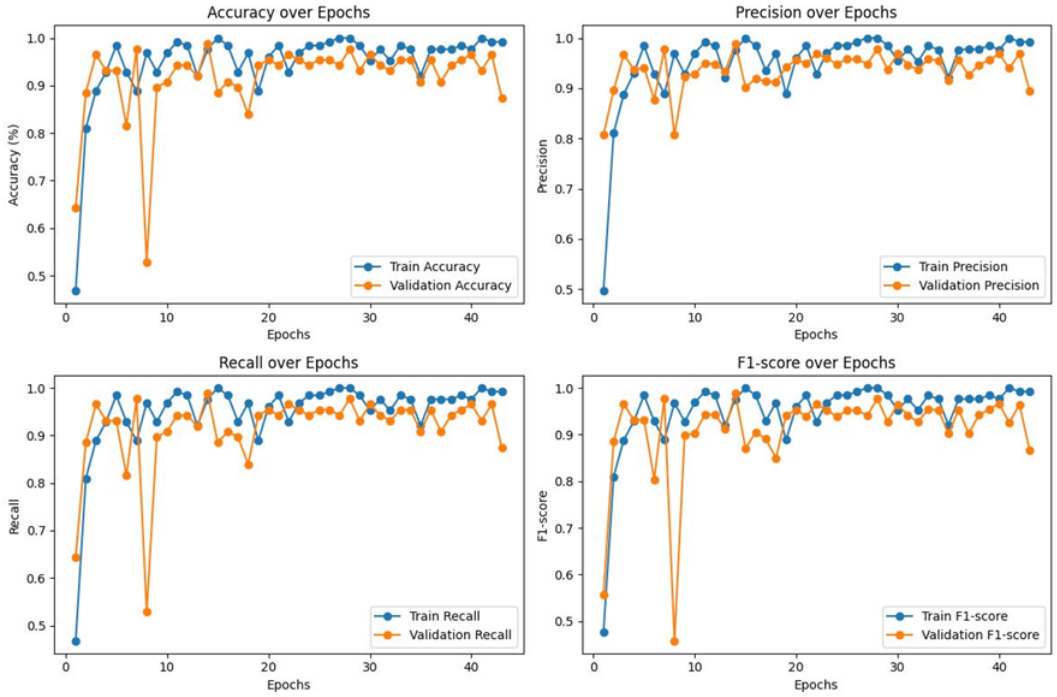
Figura 12 – Gráfico de Loss ConvNext-Tiny - LR Fixa



Fonte: acervo do autor

Nos gráficos de métricas exibidos na Figura 13, observa-se um crescimento gradual até aproximadamente 10 épocas, seguido por uma instabilidade na rede, a qual começa a se estabilizar na vigésima época

Figura 13 – Gráfico do desempenho da ConvNext-Tiny - LR Fixa



Fonte: acervo do autor

5.1.1 Avaliação da variação do Learning Rate com Descongelamento Progressivo

Neste experimento, foram usados os quatro valores citados antes na seção 5.1,que são 1×10^{-2}, 1×10^{-3},1×10^{-4} 1×10^{-5}. Iniciou-se com um Fine turning raso e depois um Fine Turning mais profundo com todas as camadas descongeladas. Esta abordagem resultou em desempenho inferior.

A Tabela 3, além de apresentar similaridade em relação à interpretação da anterior, revela que os resultados para a grande maioria dos modelos foram inferiores aos obtidos quando se descongelam apenas duas camadas. Apenas dois modelos apresentaram resultados próximos aos do teste anterior: Alexnet e ConvNext-Tiny. Apesar de ainda serem considerados os melhores modelos em comparação ao teste anterior, registraram uma Loss maior.

Ranqueando os modelos apresentados na Tabela 3, observa-se que os modelos apresentaram desempenho consideravelmente inferior, caracterizado por métricas significativamente baixas e valores elevados de loss. Esses resultados indicam que essas redes estão cometendo muitos erros de classificação e realizando predições com baixa confiança.

Por outro lado, os dois primeiros modelos demonstraram desempenho robusto, com métricas elevadas de acurácia, precisão, recall e F1-score. Embora os valores de loss obtidos por esses modelos sejam maiores que os observados em experimentos anteriores,

ainda se mantêm em um patamar considerado satisfatório, principalmente por se tratar de uma tarefa de classificação multiclasse.

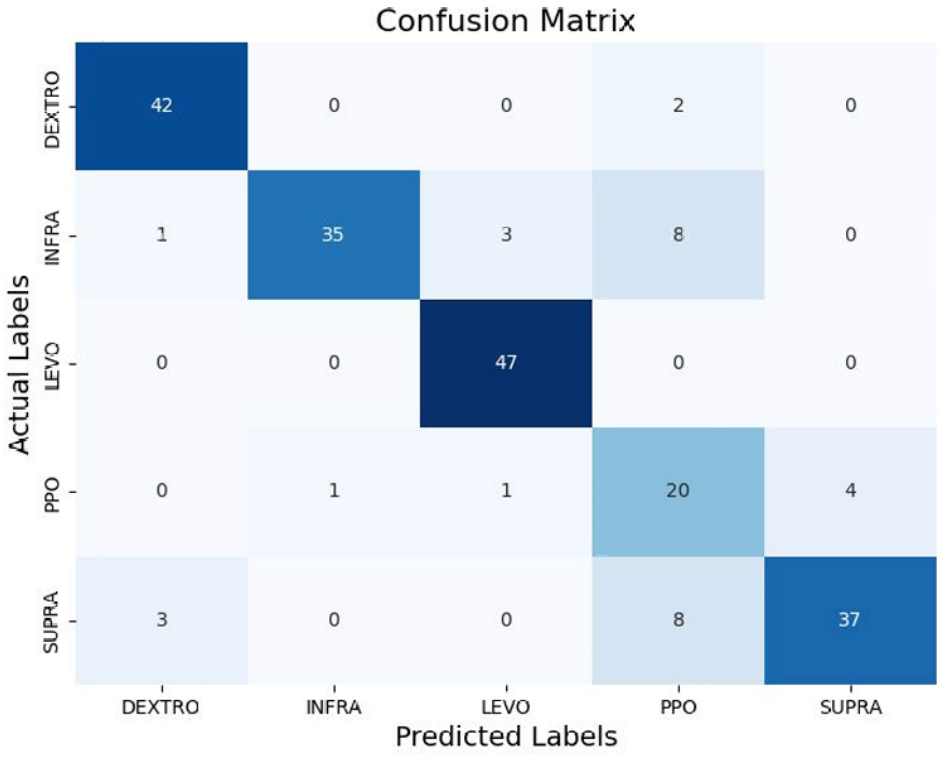
Tabela 3 – Desempenho dos modelos nas métricas de validação

Modelos	**Val Accuracy**	**Val Precision**	**Val Recall**	**Val F1**	**Val Loss**
Alexnet	0.9540	0.9600	0.9540	0.9538	0.4712
Convnext_tiny	0.8736	0.8998	0.8736	0.8719	0.4419
MaxViT_T	0.6092	0.6517	0.6092	0.6169	1.4765
EfficientNet_B0	0.2989	0.3664	0.2989	0.2490	1.5914
EfficientNet_V2S	0.3218	0.5520	0.3218	0.2643	1.6175
Resnet18	0.3678	0.3363	0.3678	0.2884	1.6210
Densenet	0.2989	0.6087	0.2989	0.2202	1.6479
GoogLeNet	0.3333	0.3600	0.3333	0.3042	2.5531
MnasNet1_0	0.3103	0.4369	0.3103	0.2475	5.0228

5.1.2 Teste dos melhores modelos com a base Cover Test

A Figura 14 ilustra a matriz de confusão. É perceptível que o modelo treinado da rede Alexnet, com apenas duas camadas descongeladas, obteve os seguintes acertos: DEXTRO com 95,45%, INFRA com 74,47%, LEVO com 100%, PPO com 76,92% e SUPRA com 75,51%. Ao se analisar os erros evidenciados na matriz de confusão, deve-se considerar que as posições do olhar nos vídeos do Cover Test podem apresentar semelhanças visuais sutis, dificultando, até mesmo, a discriminação a olho nu. Portanto, determinados equívocos perpetrados pelo modelo podem estar relacionados às restrições intrínsecas dos próprios dados, e não à deficiência do modelo em realizar predições.

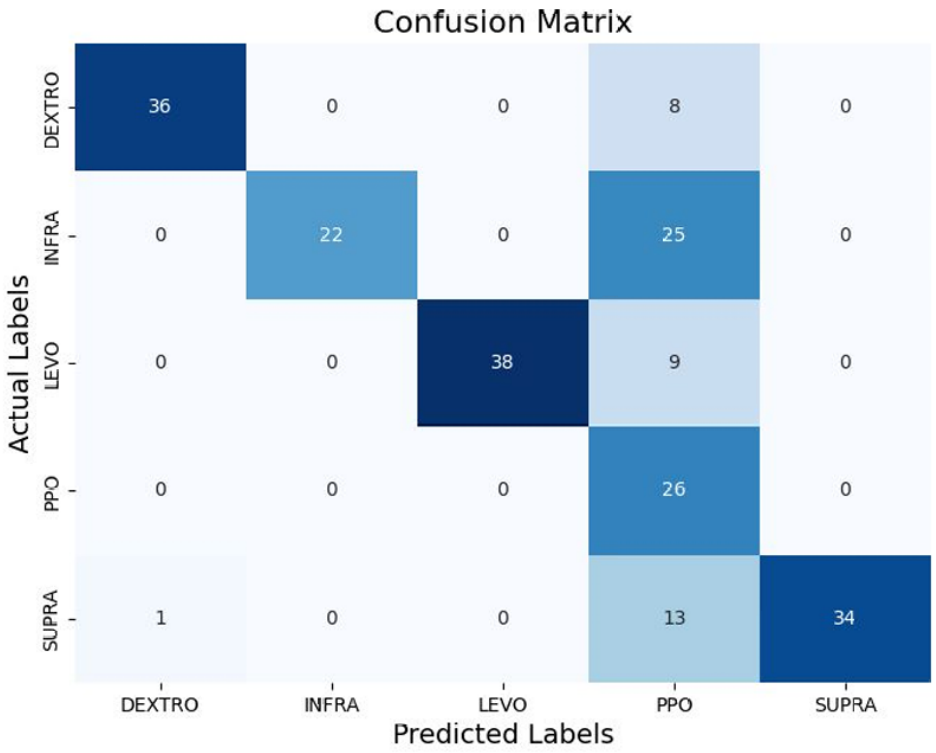
Figura 14 – Teste Alexnet LR FIXA



Fonte: acervo do autor

Já na matriz de confusão resultante da aplicação da rede ConvNext-Tiny apresentada na Figura 15, observa-se uma diferença nos resultados. Embora ocupe a primeira posição entre os modelos testados, a rede apresentou dificuldades na generalização dos padrões, sendo significantemente confundida com a posição PPO, evidenciando que não conseguiu assimilar adequadamente os padrões discriminativos. Considera-se que a classe PPO é a mais simples de reconhecer, pois as posições geralmente são consistentes e facilmente distinguíveis, embora outras possam ser confundíveis com esta. É evidente que ela se adaptou muito bem para o PPO, conseguindo 100% de acertos. Por outro lado, a Alexnet obteve 76,92%. Isso mostra que a rede aprendeu os padrões corretamente e não apenas se ajustou, mas também apresentou uma distribuição de acertos mais equilibrada entre as diferentes classes.

Figura 15 – Teste ConvNext-Tiny LR Fixa

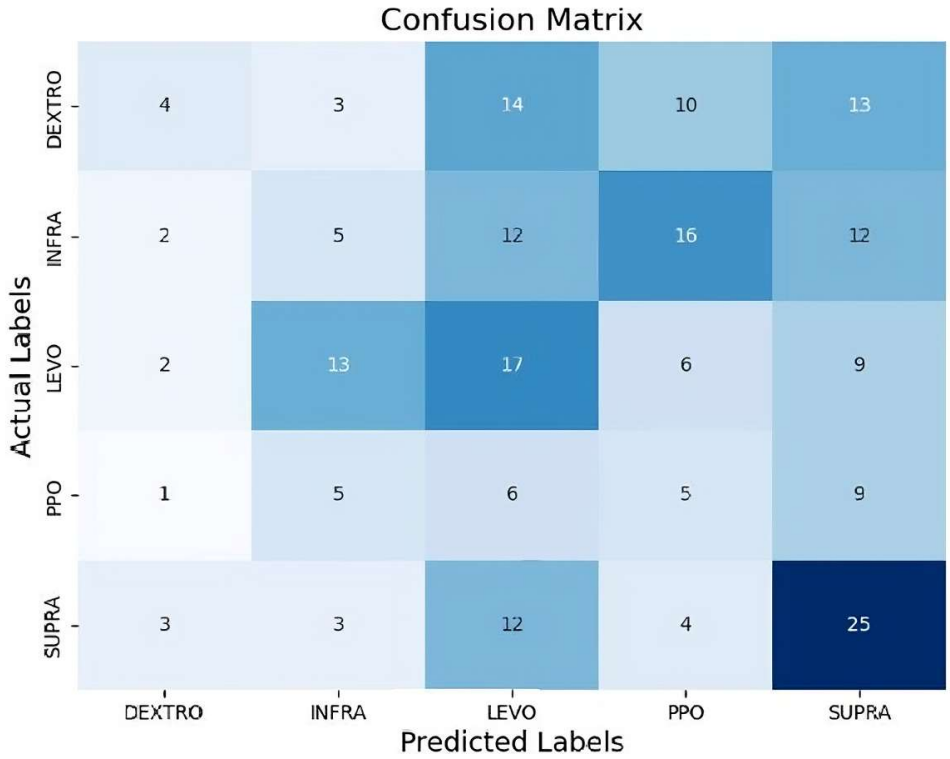


Fonte: acervo do autor

Nas Figuras 16 e 17 são apresentadas as matrizes de confusão dos modelos AlexNet e Convnext-Tiny resultantes do teste com os modelos treinados com descongelamento progressivo.

A AlexNet, mostrada na Figura 16, teve um desempenho equilibrado entre as classes. No entanto, vários erros de classificação.. Nota-se, por exemplo, uma considerável dificuldade em distinguir entre as classes DEXTRO, LEVO e SUPRA. Isso mostra que o modelo enfrentou dificuldades para perceber diferenças entre padrões que podem visualmente serem semelhantes entre si. No entanto, a rede não apresentou sinais evidentes de sobreajuste a nenhuma classe particular, o que pode ser interpretado como uma aprendizagem mais generalizável, embora com menor precisão em algumas classes. Isso significa que a rede é capaz de generalizar bem, porém não aprendeu adequadamente a descriminalizar características para as classes específicas. A classe DEXTRO foi frequentemente confundida com LEVO, SUPRA e PPO; A classe LEVO foi predita corretamente 17 vezes, mas confundida com DEXTRO e SUPRA ; A classe SUPRA foi confundida com LEVO e PPO.

Figura 16 – Teste Alexnet LR Descongelamento Progressivo

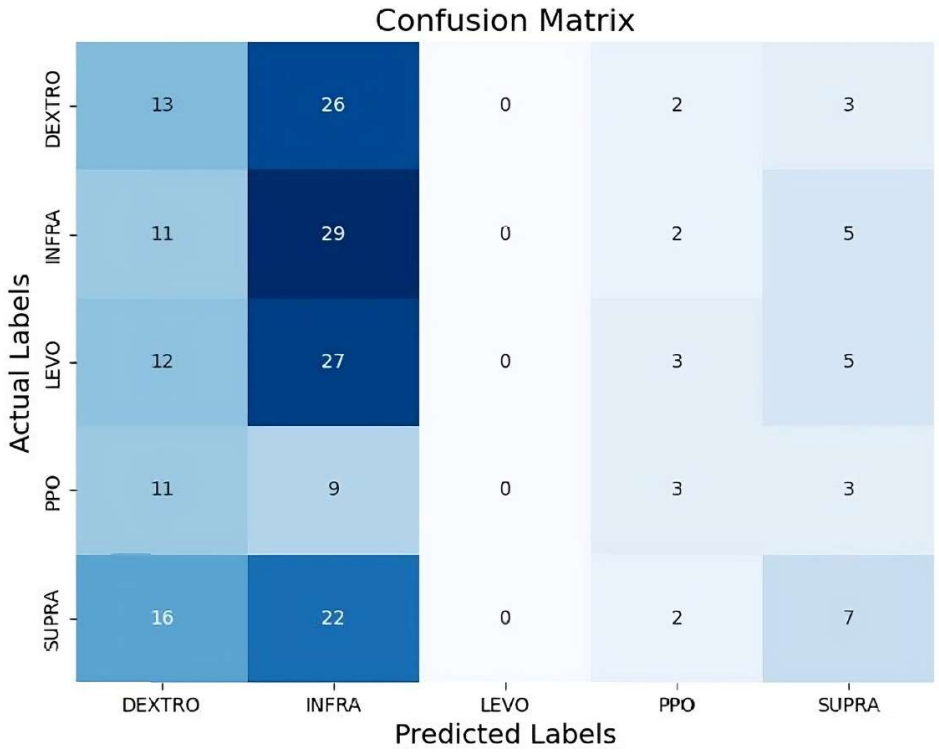


Fonte: acervo do autor

Já na Figura 17, a rede ConvNext-Tiny, mesmo sendo considerada uma arquitetura mais moderna e ocupando uma posição de liderança no ranking, demonstrou um comportamento de ajuste excessivo às classes DEXTRO e INFRA utilizando descongelamento progressivo.

A rede ConvNext-Tiny ignorou praticamente as demais classes, LEVO, PPO e SUPRA, que foram confundidas com DEXTRO e INFRA. Esse comportamento sugere que a rede não foi capaz de generalizar bem para estas outras classes, mesmo para poses com padrões simples como PPO. Isso pode indicar sobreajuste a características específicas do conjunto de treinamento, reduzindo sua capacidade de generalização.

Figura 17 – Teste ConvNext-Tiny LR Descongelamento Progressivo



Fonte: acervo do autor

5.1.3 Estudo de caso em vídeos do Cover Test

Para o início da fase de inferência em vídeo do estudo, foi necessário primeiramente escolher o modelo mais adequado. Por meio dos resultados obtidos na seção 5.1.2, onde são apresentados os testes dos melhores modelos na base Cover Test, que contém todas as poses dos voluntários, extraídas e separadas por classes para a avaliação em teste, observou-se que, dentre esses cinco modelos, o modelo da AlexNet demonstrou um desempenho satisfatório em comparação aos demais. Dessa forma, ele foi o modelo selecionado para teste de inferência de vídeo.

Dentre os vídeos disponíveis, foram selecionados os pacientes X e Y, por conterem todos os elementos esperados em uma situação do domínio real, como a presença das mãos da profissional durante o exame, além de variações de iluminação e movimento.

Foram utilizadas as bibliotecas OpenCV (cv2) e MediaPipe, responsáveis, respectivamente, pelo carregamento do vídeo e pelo tratamento dos frames. O processamento dos frames foi realizado da mesma forma descrita na etapa de Preparação dos Experimentos, na Seção 4.5. Os frames tratados foram então utilizados como entrada para o modelo na fase de inferência. Por fim, foram geradas as métricas de desempenho: F1-score, precisão, recall e matriz de confusão, para avaliar os resultados obtidos.

Para uma avaliação precisa, os frames das poses extraídos foram rotulados utili-

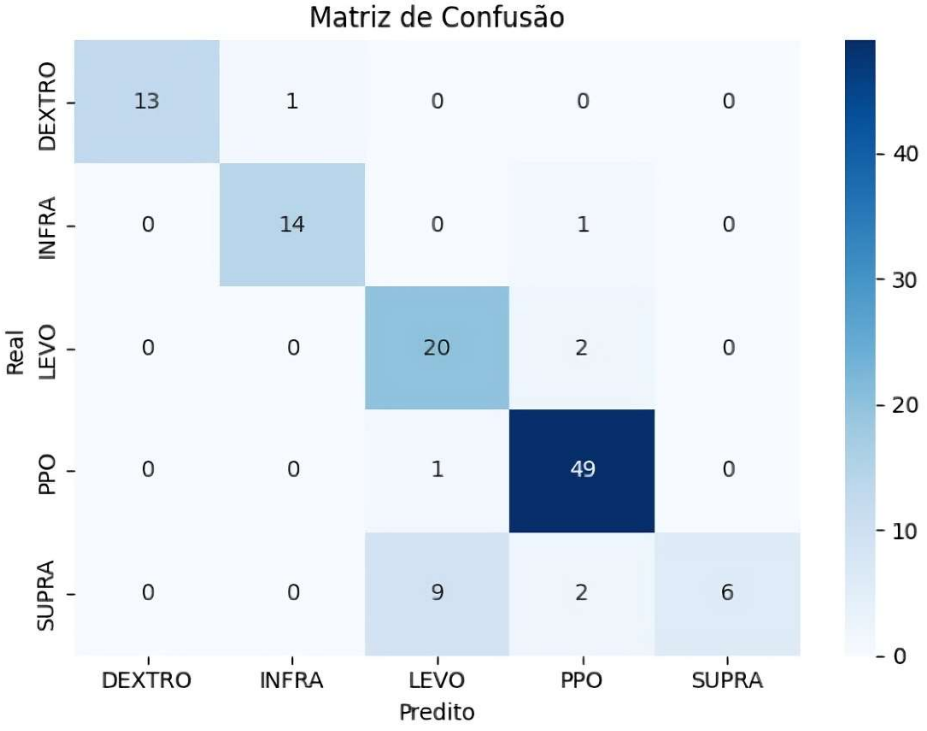
zando, localmente, a ferramenta Label Studio (HEARTEX, 2023) . O processo gerou um arquivo CSV contendo os campos ID, label e nome, que será utilizado na inferência em vídeo.

O resultado obtido para a paciente X é apresentado na Figura 18, na qual se observa que o modelo conseguiu distinguir satisfatoriamente todas as classes, com ressalva para a posição SUPRA uma vez que em determinados momentos, devido à similaridade entre algumas posições especificas, ocorreram alguns erros de classificação, assim como para os pequenos erros cometidos nas outras classes. De modo geral, conforme a Tabela 5, observa-se que o modelo apresentou um desempenho promissor para o vídeo da paciente X, alcançando uma precisão de 90%.

F1-score	**Recall**	**Precisão**
0.825	0.820	0.901

Tabela 4 – Métricas de desempenho do modelo

Figura 18 – Inferência em video da paciente X



Fonte: acervo do autor

Na Figura 19 são apresentados alguns frames que demonstram que, para as classes PPO, DEXTRO e INFRA, a rede mantém um desempenho satisfatório de classificação, sem sinais de viés para outras classes. Entretanto, observa-se um viés em direção à classe LEVO quando SUPRA está na condição de oclusão do olho esquerdo, ocasionando uma

queda na precisão da classe SUPRA para 56%, o que justifica os falsos positivos em LEVO na Figura 18.

Figura 19 – Frames de Inferência em video da paciente X

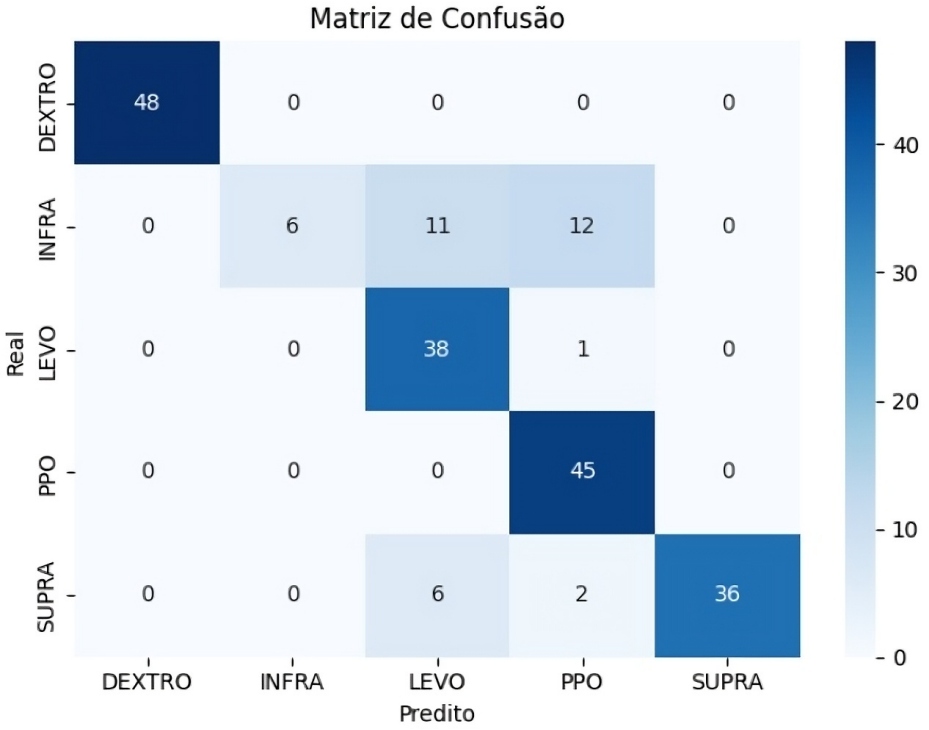
Fonte: acervo do autor

O resultado da inferência em video da paciente Y apresentado na Figura 20, demonstra que o modelo cometeu alguns Falsos Positivos para a classe INFRA em LEVO e DEXTRO. Alcançando métricas inferiores a paciente X, com F1-Score de 78% equilibrado com as demais métricas.

F1-score	**Recall**	**Precisão**
0.781	0.799	0.888

Tabela 5 – Métricas de desempenho do modelo

Figura 20 – Inferência em video da paciente Y



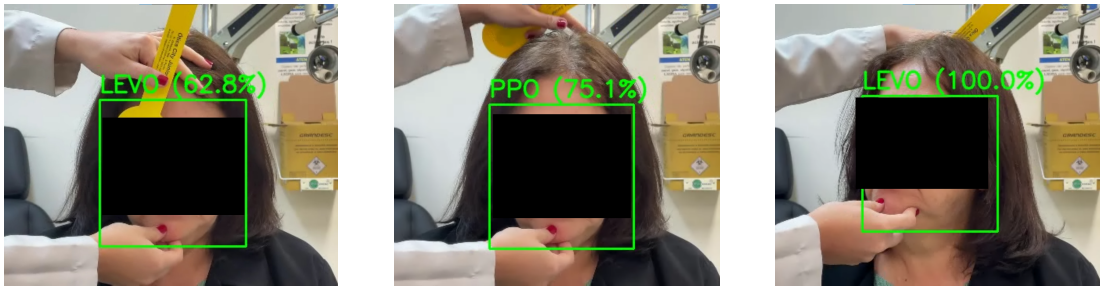
Fonte: acervo do autor

Na Figura 21 são apresentados alguns frames extraídos da inferência em vídeo da Paciente Y. O frame (a) evidencia que, quando o olho esquerdo é ocluído, a rede pode estar perdendo informações relevantes, o que a torna propensa a predizer a classe LEVO, sendo na realidade INFRA, de forma semelhante ao observado na Figura 19, com a classe SUPRA deixando-a com uma baixa precisão de 56%.

No frame (b), é possível observar a similaridade entre a classe INFRA E PPO, cujas diferenças são sutis. Os falsos negativos cometidos sugerem que a rede pode não ter visto amostras suficientes da classe INFRA, o que limitou a aprendizagem de padrões mais discerníveis e resultou em imprecisão diante de um novo contexto, em que as posições são facilmente confundíveis, dependendo do procedimento de posicionamento do olhar, uma vez que para a paciente X na Figura 19, a classe INFRA chega a 86% de precisão.

Por outro lado, no frame (c), nota-se que a posição LEVO apresenta uma leve inclinação vertical para cima, aproximando-se de um movimento SUPRA, sendo uma variação pouco presente, ou até mesmo inexistente, na base de treinamento. Ainda assim, a rede alcança 100% de precisão, o que pode indicar uma tendência de classificação para a classe LEVO.

Figura 21 – Frames extraídos da inferência em video da paciente Y



(a) INFRA predito LEVO **(b)** INFRA predito PPO **(c)** LEVO com inclinação

Fonte: Acervo do Autor

6 Conclusão

Este estudo teve como objetivo avaliar o desempenho de diferentes arquiteturas de redes neurais convolucionais nas classificações para 5 do olhar: LEVO, DEXTRO, SUPRA, INFRA E PPO em vídeos Cover Test, visando auxiliar no diagnóstico do estrabismo em vídeos do cover test. O estudo fez uso da base de dados Cover Test para teste de uso real dos melhores modelos, por meio de uma avaliação de treino completa em uma variação de hiperparâmetros e técnica como Fine-Turning.

A avaliação do desempenho das arquiteturas foi realizada usando medidas como Acurácia, Precisão, F1-Score e Recall. Também foi observada a função de perda. As melhores medidas foram coletadas para comparar as arquiteturas. Além disso, foi usada a Matriz de confusão para análise dos resultados por classe.

Por meio dos resultados de teste final na base de dados Cover Test, observa-se que os melhores desempenhos nos testes foram alcançados pelas três arquiteturas com maior número de parâmetros totais: AlexNet, MaxViT e ConvNeXt-Tiny. No entanto, somente um número grande de parâmetros não garanta um bom desempenho, podendo ter efeito contrário. Esse resultado pode estar associado à complexidade dessas redes, que empregam estratégias estruturais distintas e conferem uma capacidade representacional superior em relação às demais AlexNet sendo uma Arquitetura mais simples apresentou melhor ajuste para a tarefa do que MaxVit e ConvNext-Tiny, que replicam técnicas avançadas de Transformers.

Foi mostrado um desempenho superior das arquiteturas em que apenas as duas últimas camadas foram mantidas treináveis, em comparação com aquelas que passaram pela técnica de descongelamento progressivo de todas as camadas. Esse bom desempenho pode acontecer porque os modelos usados já passaram por um treinamento antes em grandes conjuntos de dados, como o ImageNet. Eles conseguem identificar bem as características importantes nas primeiras camadas da rede, onde estão os detalhes mais relevantes. Limitando o treinamento às camadas finais, é possível evitar o overfitting na validação, principalmente pela pequena quantidade de amostras totais usadas. O descongelamento total pode ter causado a perda de características úteis aprendidas previamente, dificultando a generalização e aumentando a perda de desempenho como os resultados sugerem.

Por meio da avaliação dos resultados deste estudo, conclui-se que o objetivo da tarefa foi alcançado, uma vez que foi possível avaliar comparativamente o desempenho de diversas arquiteturas CNN's na tarefa de classificação multiclasse da posição do olhar para cinco posições diferentes em vídeos do Cover Test. A análise do resultado leva ao

entendimento que essas 3 Arquiteturas robustas por meio do uso adequado de Fine-Tuning Profundo e a escolha certa da LR pode entregar resultados promissores para a tarefa-alvo em comparação com as outras arquiteturas, se sobressaindo entre as Três a Alexnet com uma precisão de 88% e Recall de 85% nos teste de modelo empregado no domínio real e no teste em vídeo alcançando 90% de precisão.

Apesar dos resultados promissores, este estudo apresenta algumas limitações. Primeiramente, a quantidade de amostras disponíveis foi relativamente pequena, o que pode ter limitado a generalização dos modelos treinados. Isso se refletiu no fato de que a rede conseguiu distinguir satisfatoriamente as classes para as pacientes X e Y, mas apresentaram dificuldade nas mesmas duas classes. INFRA, devido à sua similaridade com PPO, em que as distinções são sutis, dependendo do posicionamento do olhar da paciente, e SUPRA, que apresenta uma pequena inclinação vertical durante a posição LEVO, observada na paciente Y, gerando uma variação pouco vista ou até mesmo ausente no treinamento. A falta de mais amostras pode ter impedido que o modelo estivesse totalmente preparado para variações das poses, com destaque para as classes INFRA e SUPRA. Com um maior número de amostras de treino e variações dessas poses, o modelo poderia capturar mais padrões relevantes e melhorar a distinção entre essas classes e lidar melhor com o ruido ocasionado pelo oclusor.

Para pesquisas futuras, é possível aprofundar-se na aplicação de técnicas de regularização avançadas como Suavização de rótulos *(Label Smoothing)*, para colocar uma pequena incerteza nas labels, ajuda a evitar que o modelo se torne excessivamente confiante (CHEN et al., 2020), visando aprimorar a generalização tanto das redes que tiveram êxito como das redes menos complexas. Ademais, poderão ser realizados novos testes com amostras mais alinhadas àquelas adquiridas em consultórios, contando com a presença de um oclusor para as cinco posições e variações da pose que cause dificuldade de distinção, com o intuito de aumentar a confiabilidade das predições.

Referências

ALMEIDA, J. D. S. d. *Metodologia computacional para detecção e diagnóstico automáticos e planejamento cirúrgico do estrabismo*. Tese (Tese (Doutorado em Engenharia de Eletricidade)) — Universidade Federal do Maranhão, São Luís, 2013. Disponível em: <http://tedebc.ufma.br:8080/jspui/handle/tede/1823>. Citado na página 19.

ALMEIDA, J. D. S. de et al. Computer-aided methodology for syndromic strabismus diagnosis. *Journal of Digital Imaging*, v. 28, n. 4, p. 462–473, 2015. Citado na página 20.

BARTLETT, C. et al. Invasive or more direct measurements can provide an objective early-stopping ceiling for training deep neural networks on non-invasive or less-direct biomedical data. *SN Computer Science*, v. 4, 01 2023. Citado na página 27.

BAZAREVSKY, V. et al. *BlazeFace: Sub-millisecond Neural Face Detection on Mobile GPUs*. 2019. Disponível em: <https://arxiv.org/abs/1907.05047>. Citado na página 32.

CAO, B. et al. Adaptive transfer learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI Press, v. 24, n. 1, p. 407–412, 2010. Citado na página 26.

CHEN, B. et al. An investigation of how label smoothing affects generalization. *CoRR*, abs/2010.12648, 2020. Disponível em: <https://arxiv.org/abs/2010.12648>. Citado na página 50.

FANELLI, G. et al. Random forests for real time 3d face analysis. *Int. J. Comput. Vision*, v. 101, n. 3, p. 437–458, February 2013. Citado 2 vezes nas páginas 29 e 30.

GEORGE, A.; ROUTRAY, A. Real-time eye gaze direction classification using convolutional neural network. In: *2016 International Conference on Signal Processing and Communications (SPCOM)*. [S.l.: s.n.], 2016. p. 1–5. Citado na página 17.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 20.

GOOGLE. *MediaPipe*. 2023. Available at: <https://mediapipe.dev>. Accessed on: July 23, 2025. Citado na página 32.

GOURIER, N.; HALL, D.; CROWLEY, J. L. Estimating face orientation from robust detection of salient facial structures. 2004. Disponível em: <http://www-prima.inrialpes.fr/perso/Gourier/Faces/pose.html>. Citado 3 vezes nas páginas 15, 28 e 29.

GUPTA, K. et al. A robust approach of facial orientation recognition from facial features. *BRAIN. Broad Research in Artificial Intelligence and Neuroscience*, v. 8, n. 3, p. 5–12, 2017. ISSN 2067-3957. Disponível em: <https://www.edusoft.ro/brain/index.php/brain/article/view/705>. Citado na página 16.

HEARTEX. *Label Studio*. 2023. <https://labelstud.io/>. Accessed: 2025-08-01. Citado na página 45.

HU, J.; SHEN, L.; SUN, G. Squeeze-and-excitation networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2018. Citado na página 23.

HUANG, G. et al. *Densely Connected Convolutional Networks*. 2018. Disponível em: <https://arxiv.org/abs/1608.06993>. Citado na página 25.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. 2017. Disponível em: <https://arxiv.org/abs/1412.6980>. Citado na página 32.

KOTSIANTIS, S. B.; KANELLOPOULOS, D.; PINTELAS, P. E. Data preprocessing for supervised learning. *International journal of computer science*, v. 1, n. 2, p. 111–117, 2006. Citado na página 20.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. Curran Associates, Inc., v. 25, 2012. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>. Citado 2 vezes nas páginas 22 e 23.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015. Citado na página 12.

LEITE, F. H. F. et al. Surgical planning of horizontal strabismus using multiple output regression tree. *Computers in Biology and Medicine*, v. 134, p. 104493, 2021. ISSN 0010-4825. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0010482521002870>. Citado 2 vezes nas páginas 12 e 13.

LI, C. et al. Efficient training of large vision models via advanced automated progressive learning. 2024. Disponível em: <https://arxiv.org/abs/2410.00350>. Citado na página 27.

LIANG, J. et al. Two - terminal fault location method of distribution network based on adaptive convolution neural network. *IEEE Access*, PP, p. 1–1, 03 2020. Citado na página 21.

LIU, Z. et al. Deep learning face attributes in the wild. 2015. Disponível em: <https://arxiv.org/abs/1411.7766>. Citado na página 15.

LIU, Z. et al. *A ConvNet for the 2020s*. 2022. Disponível em: <https://arxiv.org/abs/2201.03545>. Citado na página 24.

LUGARESI, C. et al. *MediaPipe: A Framework for Building Perception Pipelines*. 2019. Disponível em: <https://arxiv.org/abs/1906.08172>. Citado na página 31.

MUKHERJEE, S.; ROBERTSON, N. Deep head pose: Gaze-direction estimation in multimodal video. *IEEE Transactions on Multimedia*, v. 17, p. 1–1, 11 2015. Citado na página 16.

O'SHEA, K.; NASH, R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015. Citado na página 21.

PRINCE, S. J. *Understanding Deep Learning*. The MIT Press, 2023. Disponível em: <http://udlbook.com>. Citado na página 22.

RUIZ, N.; CHONG, E.; REHG, J. M. Fine-grained head pose estimation without keypoints. p. 2155–215509, 2018. Citado na página 12.

SETHI, D.; ARORA, K.; SUSAN, S. Transfer learning by deep tuning of pre-trained networks for pulmonary nodule detection. p. 168–173, 11 2020. Citado na página 26.

SHAH, S. M. et al. A driver gaze estimation method based on deep learning. *Sensors*, v. 22, n. 10, 2022. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/22/10/3959>. Citado na página 17.

SZEGEDY, C. et al. *Going Deeper with Convolutions*. 2014. Disponível em: <https://arxiv.org/abs/1409.4842>. Citado na página 25.

SZEGEDY, C. et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. Disponível em: <https://arxiv.org/abs/1512.00567>. Citado na página 24.

TAN, M. et al. *MnasNet: Platform-Aware Neural Architecture Search for Mobile*. 2019. Disponível em: <https://arxiv.org/abs/1807.11626>. Citado na página 26.

TAN, M.; LE, Q. V. *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. 2020. Disponível em: <https://arxiv.org/abs/1905.11946>. Citado na página 25.

TAN, M.; LE, Q. V. *EfficientNetV2: Smaller Models and Faster Training*. 2021. Disponível em: <https://arxiv.org/abs/2104.00298>. Citado na página 24.

TU, Z. et al. *MaxViT: Multi-Axis Vision Transformer*. 2022. Disponível em: <https://arxiv.org/abs/2204.01697>. Citado 2 vezes nas páginas 23 e 24.

VALENTE, T. L. A. Metodologia computacional para detecção e diagnóstico automáticos de estrabismo em vídeos digitais utilizando o cover test. São Luís, 2017. Citado 2 vezes nas páginas 12 e 19.

VRBANČIČ, G.; PODGORELEC, V. Transfer learning with adaptive fine-tuning. *IEEE Access*, IEEE, v. 8, p. 196197–196211, 2020. Citado na página 26.

YAMASHITA, R. et al. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, Springer, v. 9, n. 4, p. 611–629, 2018. Citado na página 21.

YARKHEIR, M. et al. Automated strabismus detection and classification using deep learning analysis of facial images. *Scientific Reports*, Springer Nature, v. 15, p. 3910, 2025. Citado na página 12.

ZHOU, K. et al. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Institute of Electrical and Electronics Engineers (IEEE), p. 120, 2022. ISSN 1939-3539. Disponível em: <http://dx.doi.org/10.1109/TPAMI.2022.3195549>. Citado na página 13.

ZUHURA, F.; HOSSAIN, S. A framework for real-time orientation detection. *AIP Conference Proceedings*, v. 3245, 08 2024. Citado na página 16.

Óptometrial. *Simulador do teste de cobertura [imagem]*. 2025. Acesso em: 26 jul. 2025. Disponível em: <https://www.optometrial.com/cover-test-simulator/>. Citado na página 19.