UNIVERSIDADE ESTADUAL DE CAMPINAS
INSTITUTO DE MATEMÁTICA, ESTATÍSTICA E COMPUTAÇÃO CIENTÍFICA
DEPARTAMENTO DE MATEMÁTICA APLICADA

ANDERSON FERREIRA SOUZA DE OLIVEIRA

Redes neurais convolucionais aplicadas à topografia corneana para apoio ao diagnóstico de ceratocone

Campinas
2026

ANDERSON FERREIRA SOUZA DE OLIVEIRA

Redes neurais convolucionais aplicadas à topografia corneana para apoio ao diagnóstico de ceratocone

Monografia apresentada ao Instituto de Matemática, Estatística e Computação Científica da Universidade Estadual de Campinas como parte dos requisitos para obtenção de créditos na disciplina Projeto Supervisionado, sob a orientação do(a) Prof. Dr. João Batista Florindo.

Resumo

O ceratocone é uma ectasia corneana progressiva que afeta aproximadamente 265 por 100 000 indivíduos e, nos casos avançados, é uma das principais indicações de transplante de córnea. Este trabalho compara três redes neurais convolucionais profundas (ResNet-18, ResNet-50 e EfficientNet-B0) para a detecção automática de ceratocone a partir do conjunto de dados público CornOrb, formado por 1 454 olhos de 744 pacientes, com quatro mapas Orbscan por olho. A estratégia de fusão precoce (*early fusion*) concatena os quatro mapas (curvatura axial, elevação anterior, elevação posterior e paquimetria) em um tensor de quatro canais, processado por redes pré-treinadas no ImageNet e adaptadas para quatro canais de entrada. A divisão treino/validação/teste (70%/15%/15%) é feita por paciente, eliminando o vazamento de dados. A ResNet-18 obteve o melhor desempenho: $F_1 = 0{,}980$, AUC-ROC $= 0{,}997$ e revocação $= 0{,}974$, com apenas 2 falsos negativos em 77 casos reais de ceratocone, resultado competitivo com o estado da arte.

Palavras-chave: ceratocone; topografia corneana; redes neurais convolucionais; aprendizado profundo; aprendizado por transferência.

Abstract

Keratoconus is a progressive corneal ectasia affecting approximately 265 per 100,000 individuals and is a leading indication for corneal transplantation in advanced cases. This work compares three deep convolutional neural networks (ResNet-18, ResNet-50, and EfficientNet-B0) for automatic keratoconus detection from the public CornOrb dataset, comprising 1,454 eyes from 744 patients with four Orbscan maps per eye. The early fusion strategy concatenates the four maps (axial curvature, anterior elevation, posterior elevation, and pachymetry) into a four-channel input tensor processed by ImageNet-pretrained networks adapted for four input channels. The train/validation/test split (70%/15%/15%) is performed per patient, eliminating data leakage. ResNet-18 achieved the best performance: $F_1 = 0.980$, AUC-ROC = 0.997, and recall = 0.974, with only 2 false negatives out of 77 real keratoconus cases, a result competitive with the state of the art.

Keywords: keratoconus; corneal topography; convolutional neural networks; deep learning; transfer learning.

Conteúdo

1 Introdução

O ceratocone é a ectasia primária corneana mais comum, com prevalência estimada de 265 por 100 000 indivíduos [Godefrooij et al., 2017]. A doença manifesta-se tipicamente na segunda década de vida e progride com afinamento e protrusão da córnea, astigmatismo irregular e perda gradual de acuidade visual [Singh et al., 2024]. Em casos avançados, é uma das principais indicações de transplante de córnea nos países ocidentais [Gomes et al., 2026]. O diagnóstico precoce é determinante para o prognóstico, pois permite intervenções como o *cross-linking* corneano, capaz de estabilizar a progressão. Contudo, o ceratocone subclínico frequentemente mimetiza um astigmatismo comum, o que torna o diagnóstico diferencial desafiador e motiva o uso de métodos automatizados de apoio à decisão.

O diagnóstico baseia-se na topografia corneal computadorizada. O sistema Orbscan (Bausch & Lomb) mapeia simultaneamente as superfícies anterior e posterior da córnea e a sua espessura [Cairns and McGhee, 2005], sendo capaz de revelar alterações sutis antes que as manifestações clínicas sejam evidentes. Apesar da importância clínica desses exames, conjuntos de dados públicos de topografia corneana permaneceram escassos por muito tempo. Nesse cenário, o conjunto de dados CornOrb [Lazouni et al., 2026] representa um recurso público inédito baseado no Orbscan 3: são 1 454 olhos com quatro mapas corneais padronizados e anotações clínicas estruturadas, coletados em um centro oftalmológico argelino entre 2017 e 2023.

Formulação do problema. O diagnóstico assistido por computador é formulado aqui como uma tarefa de classificação binária. Para cada olho i, dispõe-se de um tensor multimodal $\mathbf{x}_i \in \mathbb{R}^{4\times224\times224}$ (os quatro mapas Orbscan empilhados ao longo do eixo de canais) e de um rótulo verdadeiro $y_i \in \{0, 1\}$ atribuído por oftalmologistas, em que $y_i = 0$ indica olho normal e $y_i = 1$ indica olho com ceratocone. O objetivo é aprender, a partir do conjunto de treino, parâmetros $\hat{\boldsymbol{\phi}}$ de uma rede convolucional $f[\,\cdot\,, \boldsymbol{\phi}]$ tais que a probabilidade predita $\lambda_i = \mathrm{sig}[f[\mathbf{x}_i, \hat{\boldsymbol{\phi}}]]$ seja próxima de y_i. A decisão $\hat{y}_i \in \{0, 1\}$ é obtida limiarizando λ_i em 0,5. No conjunto de teste, as predições $\hat{y}_i$ são comparadas aos rótulos verdadeiros por meio das métricas F_1, AUC-ROC, AUC-PR, acurácia, precisão, revocação e especificidade.

Objetivos e contribuições. Este trabalho tem três objetivos: (1) implementar e comparar ResNet-18, ResNet-50 e EfficientNet-B0 para a detecção de ceratocone via fusão precoce dos quatro mapas Orbscan; (2) descrever matematicamente cada etapa do modelo, da regressão linear aos blocos residuais, de forma autocontida e acessível; e (3) dividir os dados por paciente, evitando o vazamento de informação biológica que infla artificialmente as estimativas de desempenho.

2 Contexto Clínico

O ceratocone é um transtorno ectásico corneal progressivo e não-inflamatório [Singh et al., 2024], caracterizado pelo afinamento localizado e pela protrusão cônica da córnea. O consenso global mais recente [Gomes et al., 2026] define como parâmetros diagnósticos principais a ceratometria máxima ($K_{\max}$, em dioptrias), a espessura corneana mínima e os índices de elevação das superfícies anterior e posterior. A condição exibe considerável heterogeneidade na apresentação, e uma meta-análise de 11 529 olhos identificou que pacientes mais jovens e com $K_{\max}$ mais elevado no início apresentam risco significativamente maior de progressão [Ferdi et al., 2019].

O exame de imagem que sustenta o diagnóstico é a topografia corneana. O sistema Orbscan combina varredura por fenda luminosa com topografia de disco de Plácido [Cairns and McGhee, 2005] e fornece quatro mapas complementares. O mapa de curvatura axial descreve o poder refrativo da superfície anterior; o de elevação anterior mede a altura da superfície relativamente à esfera de melhor ajuste; o de elevação posterior faz o mesmo para a face posterior, onde as alterações precoces costumam surgir primeiro; e o de paquimetria registra a espessura corneana, em micrômetros. A combinação desses quatro mapas fornece uma descrição tridimensional da córnea, o que justifica tratá-los conjuntamente como entrada de um único modelo.

O conjunto de dados utilizado neste trabalho, o CornOrb [Lazouni et al., 2026], reúne 1 454 olhos de 744 pacientes (889 normais e 565 com ceratocone), cada um com os quatro mapas em formato PNG (560×560 pixels) e anotações clínicas estruturadas. O rótulo é binário: $y_i = 0$ (normal) ou $y_i = 1$ (ceratocone). A Tabela 1 resume as estatísticas clínicas das duas populações. Todas as diferenças entre os grupos são estatisticamente

significativas, e os valores são coerentes com a fisiopatologia da doença: olhos com ceratocone apresentam córneas mais finas, $K_{\max}$ mais elevado e astigmatismo mais acentuado do que olhos normais.

Tabela 1: Estatísticas do CornOrb [Lazouni et al., 2026] (média ± desvio padrão; $p_{\mathrm{BH}} < 0{,}001$ em todos os parâmetros após correção de Benjamini–Hochberg).

Parâmetro	Normal ($n = 889$)	Ceratocone ($n = 565$)
Idade (anos)	$33{,}5 \pm 7{,}4$	$27{,}9 \pm 7{,}1$
Paquimetria central (µm)	530 ± 43	465 ± 54
Paquimetria mínima (µm)	521 ± 44	427 ± 59
$K_{\max}$ (D)	$43{,}7 \pm 2{,}9$	$53{,}1 \pm 6{,}9$
Astigmatismo (D)	$-1{,}35 \pm 0{,}91$	$-5{,}32 \pm 3{,}32$

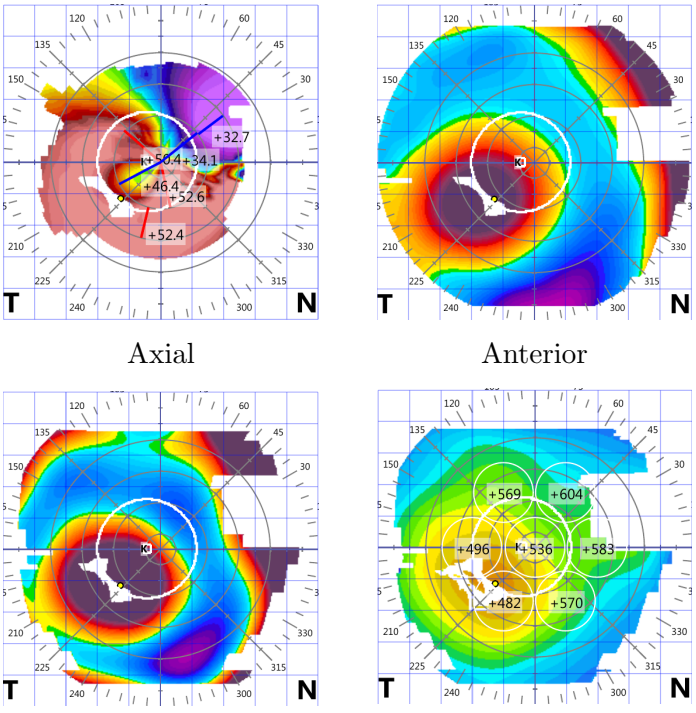

Figura 1: Exemplo de um exame de topografia corneana do conjunto CornOrb (`1AAX3_OD`), com os quatro mapas Orbscan fornecidos para cada olho. Fonte: CornOrb [Lazouni et al., 2026].

3 Fundamentos Matemáticos

Esta seção constrói, passo a passo, a matemática por trás das redes usadas no trabalho. A ideia é começar do modelo mais simples possível e ir acrescentando elementos

só quando o anterior se mostra insuficiente, de modo que cada conceito apareça motivado pelo problema que ele resolve. A notação segue Prince [2023] e está reunida na Tabela 2; daqui em diante, salvo quando um resultado específico precisa ser creditado, escrevo diretamente, sem repetir a fonte a cada equação. O caminho parte da regressão linear, passa pela rede rasa, chega à rede profunda e então define a função de custo, a otimização e a retropropagação.

Tabela 2: Notação utilizada ao longo do texto, seguindo Prince [2023].

Símbolo	Definição
$\mathbf{x} \in \mathbb{R}^{D_i}$	vetor de entrada
$f[\mathbf{x}, \boldsymbol{\phi}]$	saída da rede (modelo)
$\mathbf{h}_k$	ativações da camada k (após $a[\cdot]$)
$\mathbf{f}_k$	pré-ativações da camada k (antes de $a[\cdot]$)
$\boldsymbol{\Omega}_k$	matriz de pesos da camada k
$\boldsymbol{\beta}_k$	vetor de vieses da camada k
$\boldsymbol{\phi} = \{\boldsymbol{\beta}_k, \boldsymbol{\Omega}_k\}_{k=0}^{K}$	todos os parâmetros
$a[\cdot]$	função de ativação (elemento a elemento)
ℓ_i	custo da amostra i
$L[\boldsymbol{\phi}] = \sum_i \ell_i$	custo total
$\alpha > 0$	taxa de aprendizado
$\text{sig}[z] = 1/(1+\exp[-z])$	sigmoide logística
$\lambda_i = \text{sig}[f[\mathbf{x}_i, \boldsymbol{\phi}]]$	probabilidade predita de ceratocone
$\hat{y}_i$	predição pontual (1 se $\lambda_i > 0{,}5$)
$\mathbb{I}[z > 0]$	função indicadora (derivada da ReLU)
$\odot$	multiplicação elemento a elemento
$\text{argmin}_x\, g(x)$	valor de x que minimiza $g(x)$
$\text{argmax}_y\, g(y)$	valor de y que maximiza $g(y)$

3.1 Do modelo linear à rede profunda

O ponto de partida natural é o modelo linear, que mapeia uma entrada escalar $x \in \mathbb{R}$ em uma saída $y \in \mathbb{R}$ por meio de

$$f[x, \boldsymbol{\phi}] = \beta_0 + \beta_1 x, \qquad \boldsymbol{\phi} = \{\beta_0, \beta_1\}, \tag{1}$$

em que β_0 é o intercepto e β_1 é o peso aplicado à entrada. A limitação é evidente: a Equação (1) descreve apenas uma reta e, por isso, só captura relações lineares. A relação

entre os mapas Orbscan e a presença de ceratocone é fortemente não-linear, o que torna o modelo linear insuficiente e motiva a introdução de redes neurais.

A rede rasa supera essa limitação ao compor a entrada com funções de ativação não-lineares. No exemplo mínimo, com uma entrada, três unidades ocultas e uma saída, a rede calcula primeiro as três unidades ocultas

$$h_d = a[\theta_{d0} + \theta_{d1}\, x], \qquad d = 1, 2, 3, \tag{2}$$

e depois combina linearmente essas ativações para produzir a saída

$$f[x, \boldsymbol{\phi}] = \phi_0 + \phi_1 h_1 + \phi_2 h_2 + \phi_3 h_3. \tag{3}$$

A função de ativação $a[\cdot]$ é a ReLU (*rectified linear unit*), definida por

$$a[z] = \max(0, z) = \begin{cases} 0 & z < 0 \quad \text{(unidade inativa)}, \\ z & z \geq 0 \quad \text{(unidade ativa)}, \end{cases} \tag{4}$$

ilustrada na Figura 2. É justamente o ponto de junção em $z = 0$, que separa a região ativa da inativa, que confere à rede a sua não-linearidade: a combinação de várias unidades com diferentes pontos de junção permite aproximar funções com curvaturas arbitrárias.

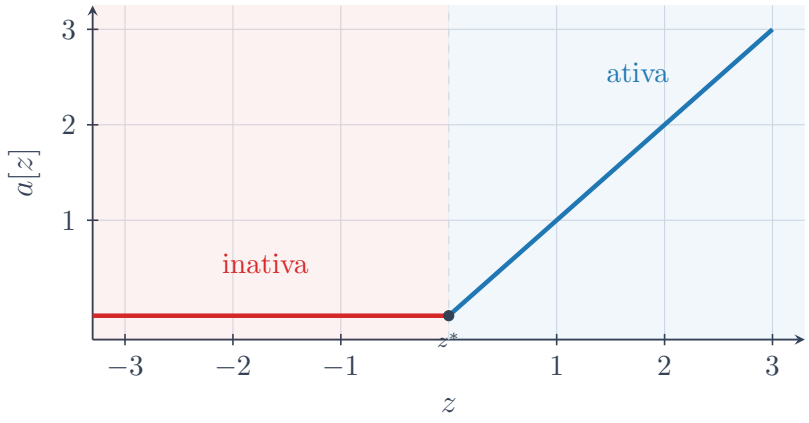

Figura 2: Função de ativação ReLU, $a[z] = \max(0, z)$. O ponto de junção $z^* = 0$ separa a região inativa (vermelho) da região ativa (azul). Fonte: adaptado de Prince [2023].

O caso geral substitui a entrada escalar por um vetor $\mathbf{x} \in \mathbb{R}^{D_i}$ e admite D

unidades ocultas. Cada unidade agrega todas as entradas por meio de

$$h_d = a\left[\theta_{d0} + \sum_{i=1}^{D_i} \theta_{di}\, x_i\right], \quad d = 1, \ldots, D, \qquad f[\mathbf{x}, \boldsymbol{\phi}] = \phi_0 + \sum_{d=1}^{D} \phi_d h_d. \tag{5}$$

No problema deste trabalho, a entrada tem dimensão $D_i = 4 \times 224 \times 224 = 200\,704$ pixels, e a saída é o escalar $f[\mathbf{x}_i, \boldsymbol{\phi}] \in \mathbb{R}$ que, após a sigmoide, expressa a probabilidade de ceratocone. Em forma matricial, definindo $\boldsymbol{\Omega}_0 \in \mathbb{R}^{D \times D_i}$ e $\boldsymbol{\beta}_0 \in \mathbb{R}^D$, a rede rasa escreve-se de modo compacto como

$$\mathbf{h}_1 = a[\boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}], \qquad f[\mathbf{x}, \boldsymbol{\phi}] = \beta_1 + \boldsymbol{\Omega}_1 \mathbf{h}_1. \tag{6}$$

Há uma garantia teórica de que redes rasas são expressivas. O teorema da aproximação universal afirma que, para qualquer função contínua $g : \mathbb{R}^{D_i} \to \mathbb{R}$ definida em um subconjunto compacto e qualquer $\varepsilon > 0$, existe uma rede com ReLU tal que $\sup_{\mathbf{x}} |f[\mathbf{x}, \boldsymbol{\phi}] - g(\mathbf{x})| < \varepsilon$. O teorema garante a existência de uma rede capaz de aproximar g com precisão arbitrária, mas não diz quantas unidades são necessárias nem que o treinamento de fato a encontre. Com $D_i = 200\,704$, uma rede rasa precisaria de um número impraticável de unidades, e é essa observação que motiva as redes profundas.

A rede profunda é obtida pela composição de várias camadas. Generalizando a notação para K camadas, a primeira camada processa a entrada e cada camada seguinte processa a anterior:

$$\mathbf{h}_1 = a[\boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0 \mathbf{x}], \tag{7}$$

$$\mathbf{h}_k = a[\boldsymbol{\beta}_{k-1} + \boldsymbol{\Omega}_{k-1} \mathbf{h}_{k-1}], \quad k = 2, \ldots, K, \tag{8}$$

$$f[\mathbf{x}, \boldsymbol{\phi}] = \beta_K + \boldsymbol{\Omega}_K \mathbf{h}_K, \tag{9}$$

com $\boldsymbol{\phi} = \{\boldsymbol{\beta}_k, \boldsymbol{\Omega}_k\}_{k=0}^{K}$. Empilhar camadas é muito mais eficiente do que alargar uma única camada: cada nova camada pode recombinar as características já extraídas pela anterior, de modo que a quantidade de regiões lineares que a rede consegue representar cresce de forma muito mais rápida com a profundidade do que com a largura. É essa eficiência que torna as redes profundas a ferramenta adequada para imagens de alta dimensão como os

mapas corneais.

3.2 Função de custo para classificação binária

Para treinar a rede, precisamos de uma medida de quão boas são as suas predições. Como o rótulo $y_i \in \{0, 1\}$ é binário, é natural modelar a saída pela distribuição de Bernoulli com parâmetro $\lambda \in [0, 1]$:

$$\Pr(y \mid \lambda) = (1 - \lambda)^{1-y}\, \lambda^{y}. \tag{10}$$

A Figura 3 mostra como a massa de probabilidade migra de $y = 0$ (normal) para $y = 1$ (ceratocone) à medida que λ cresce. O parâmetro λ deve, portanto, representar a probabilidade de o olho ter ceratocone.

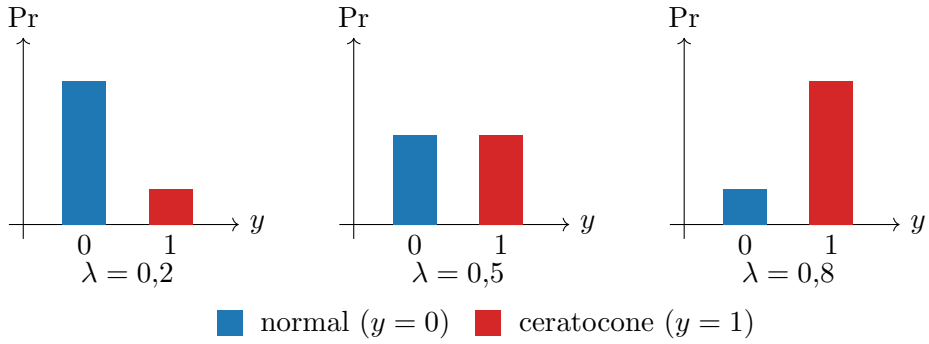

Figura 3: Distribuição de Bernoulli para três valores de λ. À esquerda, a maior parte da massa está em $y = 0$ (normal, azul); à direita, em $y = 1$ (ceratocone, vermelho). Fonte: adaptado de Prince [2023].

A rede, porém, produz um valor real $f[\mathbf{x}, \boldsymbol{\phi}] \in \mathbb{R}$ sem restrição de sinal ou de escala, enquanto λ precisa estar em $[0, 1]$. A sigmoide logística faz exatamente essa conversão:

$$\text{sig}[z] = \frac{1}{1 + \exp[-z]}, \qquad \lambda_i = \text{sig}[f[\mathbf{x}_i, \boldsymbol{\phi}]]. \tag{11}$$

A sigmoide não é uma escolha arbitrária: ela é a consequência direta de exigir que λ_i seja uma probabilidade válida da distribuição de Bernoulli. Como mostra a Figura 4, ela mapeia o logit $z = f[\mathbf{x}, \boldsymbol{\phi}]$ em $(0, 1)$, e o limiar de decisão $\lambda = 0{,}5$ separa as predições normal e ceratocone.

O custo é obtido pelo negativo da log-verossimilhança da Bernoulli. Para uma

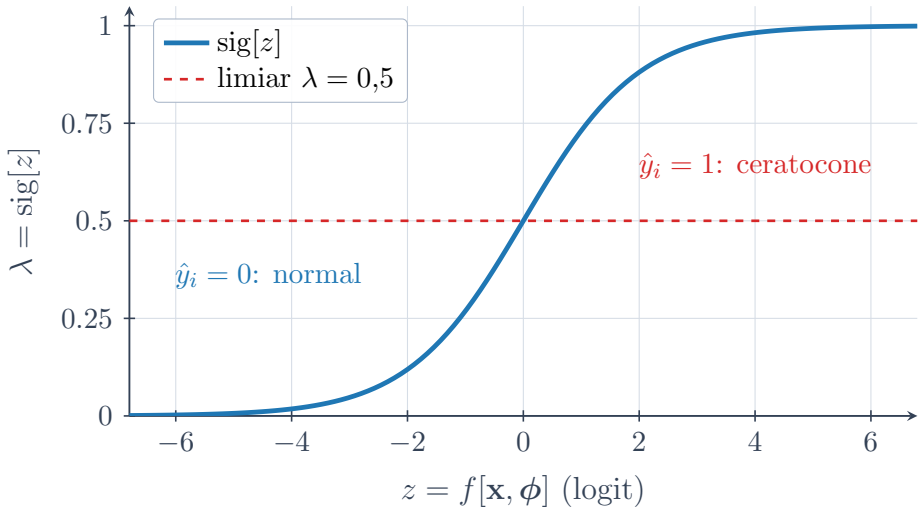

Figura 4: Sigmoide logística $\text{sig}[z] = 1/(1+\exp[-z])$. Ela transforma o logit $z \in \mathbb{R}$ em uma probabilidade $\lambda \in (0,1)$. A reta tracejada em vermelho marca o limiar de decisão $\lambda = 0{,}5$: logits positivos levam à predição de ceratocone e negativos, à de olho normal. Fonte: adaptado de Prince [2023].

amostra, isso resulta na entropia cruzada binária, e o custo total é a soma sobre o conjunto de treino:

$$\ell_i = -(1-y_i)\log\big(1-\text{sig}[f[\mathbf{x}_i,\boldsymbol{\phi}]]\big) - y_i \log\big(\text{sig}[f[\mathbf{x}_i,\boldsymbol{\phi}]]\big), \qquad L[\boldsymbol{\phi}] = \sum_{i=1}^{I} \ell_i. \tag{12}$$

Cada termo penaliza um tipo de erro: o primeiro atua quando $y_i = 0$ e a rede atribui probabilidade alta a ceratocone, e o segundo atua quando $y_i = 1$ e a rede atribui probabilidade baixa. Treinar a rede é, então, encontrar os parâmetros que minimizam esse custo, ao passo que a inferência escolhe a classe mais provável:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\text{argmin}}\, L[\boldsymbol{\phi}], \qquad \hat{y}_i = \underset{y\in\{0,1\}}{\text{argmax}}\, \Pr(y \mid \lambda_i) = \begin{cases} 1 & \lambda_i > 0{,}5, \\ 0 & \text{caso contrário.} \end{cases} \tag{13}$$

3.3 Otimização: gradiente descendente e Adam

A minimização de $L[\boldsymbol{\phi}]$ é feita por gradiente descendente. A atualização básica desloca os parâmetros na direção oposta ao gradiente, com passo controlado pela taxa de

aprendizado α:

$$\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \alpha \frac{\partial L}{\partial \boldsymbol{\phi}}. \tag{14}$$

Como o conjunto de treino é grande, avalia-se o gradiente em *mini-batches* $\mathcal{B}_t$ a cada passo, o que dá origem ao gradiente descendente estocástico:

$$\boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \alpha \sum_{i \in \mathcal{B}_t} \frac{\partial \ell_i[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}. \tag{15}$$

Um passo fixo α é, porém, pouco robusto: direções com gradientes de magnitudes muito diferentes exigem passos diferentes. O otimizador Adam, usado neste trabalho, resolve esse problema normalizando cada coordenada pela sua magnitude histórica. Ele mantém estimativas do primeiro e do segundo momentos do gradiente, atualizadas como médias móveis exponenciais com fatores $\beta = 0{,}9$ e $\gamma = 0{,}999$:

$$\mathbf{m}_{t+1} \leftarrow \beta\, \mathbf{m}_t + (1-\beta) \frac{\partial L[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}}, \tag{16}$$

$$\mathbf{v}_{t+1} \leftarrow \gamma\, \mathbf{v}_t + (1-\gamma) \left(\frac{\partial L[\boldsymbol{\phi}_t]}{\partial \boldsymbol{\phi}} \right)^2. \tag{17}$$

Como $\mathbf{m}$ e $\mathbf{v}$ são inicializados em zero, eles ficam enviesados para baixo nas primeiras iterações; a correção de viés compensa esse efeito antes da atualização final:

$$\tilde{\mathbf{m}}_{t+1} = \frac{\mathbf{m}_{t+1}}{1-\beta^{t+1}}, \quad \tilde{\mathbf{v}}_{t+1} = \frac{\mathbf{v}_{t+1}}{1-\gamma^{t+1}}, \quad \boldsymbol{\phi}_{t+1} \leftarrow \boldsymbol{\phi}_t - \frac{\alpha}{\sqrt{\tilde{\mathbf{v}}_{t+1}} + \epsilon} \odot \tilde{\mathbf{m}}_{t+1}, \tag{18}$$

em que $\epsilon = 10^{-8}$ previne divisão por zero. O efeito prático é uma taxa de aprendizado adaptativa por coordenada, que torna o treinamento mais estável e menos sensível à escolha de α.

3.4 Retropropagação e regularização

O gradiente descendente (Equação (15)) requer as derivadas $\partial \ell_i / \partial \boldsymbol{\beta}_k$ e $\partial \ell_i / \partial \boldsymbol{\Omega}_k$ para todas as camadas. Calculá-las separadamente seria proibitivo; a retropropagação as obtém eficientemente em duas passagens. Na passagem direta (*forward*), a entrada

percorre a rede e todas as pré-ativações $\mathbf{f}_k$ e ativações $\mathbf{h}_k$ são armazenadas:

$$\mathbf{f}_0 = \boldsymbol{\beta}_0 + \boldsymbol{\Omega}_0\mathbf{x}_i, \quad \mathbf{h}_k = a[\mathbf{f}_{k-1}], \quad \mathbf{f}_k = \boldsymbol{\beta}_k + \boldsymbol{\Omega}_k\mathbf{h}_k, \quad k \in \{1, \ldots, K\}. \tag{19}$$

Na passagem reversa (*backward*), parte-se da derivada do custo em relação à saída e propaga-se o gradiente de trás para frente. Para cada camada,

$$\frac{\partial \ell_i}{\partial \boldsymbol{\beta}_k} = \frac{\partial \ell_i}{\partial \mathbf{f}_k}, \quad \frac{\partial \ell_i}{\partial \boldsymbol{\Omega}_k} = \frac{\partial \ell_i}{\partial \mathbf{f}_k}\mathbf{h}_k^\top, \quad \frac{\partial \ell_i}{\partial \mathbf{f}_{k-1}} = \mathbb{I}[\mathbf{f}_{k-1} > 0] \odot \boldsymbol{\Omega}_k^\top \frac{\partial \ell_i}{\partial \mathbf{f}_k}, \tag{20}$$

em que o fator $\mathbb{I}[\mathbf{f}_{k-1} > 0]$ é a derivada da ReLU, que deixa passar o gradiente apenas pelas unidades ativas. O ponto de partida dessa recursão tem forma particularmente simples para a entropia cruzada binária com sigmoide:

$$\frac{\partial \ell_i}{\partial \mathbf{f}_K} = \lambda_i - y_i = \text{sig}[f[\mathbf{x}_i, \boldsymbol{\phi}]] - y_i, \tag{21}$$

ou seja, o próprio erro de predição: positivo quando a rede superestima a probabilidade de ceratocone e negativo quando a subestima. A Figura 5 resume o fluxo das duas passagens.

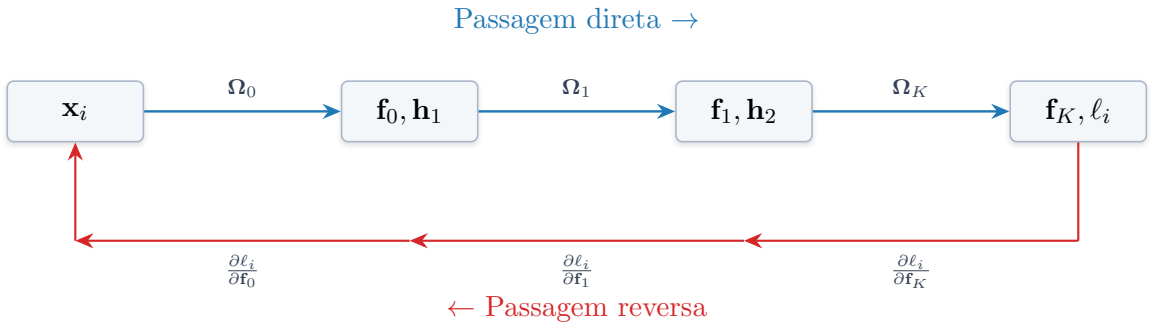

Figura 5: Passagem direta (azul, acima): propaga $\mathbf{x}_i$ até ℓ_i aplicando os pesos $\boldsymbol{\Omega}_k$ e a ReLU e armazenando ativações. Passagem reversa (vermelho, abaixo): propaga $\partial \ell_i / \partial \mathbf{f}_k$ de ℓ_i até $\mathbf{x}_i$. Fonte: baseado nas Figuras 7.3 e 7.4 de Prince [2023].

Para evitar o sobreajuste, empregamos o *dropout*, que durante o treinamento zera aleatoriamente uma fração das ativações de uma camada e reescala as remanescentes para preservar a magnitude esperada:

$$\tilde{\mathbf{h}}_k = \frac{1}{1 - p_d}(\mathbf{h}_k \odot \mathbf{m}), \qquad m_j \sim \text{Bernoulli}(1 - p_d), \tag{22}$$

em que $p_d = 0{,}4$ é a taxa de descarte. Ao impedir que a rede dependa de unidades específicas, o dropout funciona como um regularizador e melhora a generalização, sendo particularmente útil em um conjunto de tamanho moderado como o CornOrb.

4 ImageNet e Aprendizado por Transferência

Treinar uma rede profunda do zero exige um volume de dados que o CornOrb, com pouco mais de mil olhos, não oferece. A solução adotada é o aprendizado por transferência: parte-se de uma rede já treinada em uma tarefa-fonte rica em dados e adaptam-se os seus pesos à tarefa-alvo. A tarefa-fonte é a classificação no ImageNet [Deng et al., 2009], um banco hierárquico com mais de um milhão de imagens em mil categorias; o desafio associado, o ILSVRC [Russakovsky et al., 2015], impulsionou as arquiteturas modernas de CNNs e é o benchmark padrão de classificação de imagens.

Os pesos pré-treinados $\hat{\boldsymbol{\phi}}_{\text{ImageNet}} = \operatorname{argmin}_{\boldsymbol{\phi}} L_{\text{ImageNet}}[\boldsymbol{\phi}]$ inicializam as camadas do *backbone* em uma região informativa do espaço de parâmetros. As camadas iniciais aprendem detectores de bordas e texturas, características gerais e transferíveis entre domínios; as camadas profundas capturam padrões específicos do domínio, que são reajustados durante o *fine-tuning*. Essa divisão explica por que a transferência funciona mesmo quando os domínios diferem: Tajbakhsh et al. [2016] demonstraram que, em imagens médicas, o fine-tuning a partir de pesos do ImageNet supera o treinamento do zero, e Raghu et al. [2019] mostraram que as representações de baixo nível aprendidas no ImageNet são efetivamente reutilizáveis em imagens clínicas. Esses resultados fundamentam a estratégia de transferência empregada aqui.

5 Arquiteturas

Esta seção descreve as três arquiteturas comparadas. Todas operam sobre imagens por meio da convolução 2D e diferem na forma como organizam e escalonam as camadas. A convolução estende a multiplicação matricial das camadas densas para o caso de imagens, em que cada saída depende apenas de uma vizinhança local da entrada e o

mesmo filtro é aplicado em todas as posições:

$$h_{ij} = a\left[\beta + \sum_{m=1}^{F_h}\sum_{n=1}^{F_w} \omega_{mn}\, x_{i+m-2,\, j+n-2}\right], \tag{23}$$

em que ω_{mn} são os pesos do filtro $F_h \times F_w$. O compartilhamento de pesos reduz drasticamente o número de parâmetros: com C_i canais de entrada, C_o canais de saída e filtros $K \times K$, são apenas $C_i \times C_o \times K^2$ pesos, em vez de uma conexão por par de pixels.

A forma como o filtro percorre a imagem é governada por dois hiperparâmetros [Prince, 2023]. O *stride* (passo) é o deslocamento do filtro entre posições consecutivas: com *stride* 1 avalia-se toda posição, ao passo que com *stride* 2 gera-se aproximadamente metade das saídas em cada dimensão, reduzindo a resolução espacial. O *padding* (preenchimento) trata o que ocorre nas bordas, onde o filtro ultrapassa a entrada; há duas opções usuais. No *zero-padding*, assume-se que a entrada vale zero fora do seu intervalo válido, o que permite preservar o tamanho espacial; na convolução *"valid"*, descartam-se as posições em que o filtro não cabe inteiramente na entrada, o que não introduz informação artificial nas bordas, mas faz a saída encolher.

Esses dois hiperparâmetros determinam o tamanho da saída:

$$\left\lfloor \frac{W - K + 2p}{s} \right\rfloor + 1, \tag{24}$$

em que W é o lado da entrada, K o lado do filtro, p o *padding*, s o *stride* e $\lfloor\cdot\rfloor$ a função piso. É por isso que, na Tabela 3, a convolução inicial 7×7 com *stride* 2 (e $p = 3$) reduz a entrada de 224 para

$$\left\lfloor \frac{224 - 7 + 2 \cdot 3}{2} \right\rfloor + 1 = 112.$$

5.1 ResNet: conexões residuais e *batch normalization*

Redes muito profundas, quando empilhadas de forma puramente sequencial, sofrem do problema dos gradientes fragmentados. Numa cadeia de camadas, o gradiente da saída em relação a uma camada inicial é o produto de muitos termos,

$$\frac{\partial y}{\partial \mathbf{f}_1} = \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1} \cdot \frac{\partial \mathbf{f}_3}{\partial \mathbf{f}_2} \cdots, \tag{25}$$

e esse produto torna-se numericamente instável, dificultando o treinamento. A ResNet [He et al., 2016] resolve o problema com a conexão residual, que soma a entrada do bloco à sua transformação:

$$\mathbf{h}_k = \mathbf{h}_{k-1} + \mathbf{f}_k[\mathbf{h}_{k-1}, \boldsymbol{\phi}_k]. \tag{26}$$

Com isso, o gradiente passa a conter um termo identidade,

$$\frac{\partial y}{\partial \mathbf{f}_1} = \mathbf{I} + \frac{\partial \mathbf{f}_2}{\partial \mathbf{f}_1} + \cdots, \tag{27}$$

que garante um caminho direto para a sua propagação, mesmo em redes com dezenas de camadas. A ideia é que, se a transformação ótima de um bloco for próxima da identidade, basta que ele aprenda um resíduo pequeno, tarefa mais fácil do que aprender a identidade a partir do zero.

A *batch normalization* complementa as conexões residuais ao estabilizar a distribuição das ativações. Para cada *batch* $\mathcal{B}$, ela padroniza as ativações usando média e desvio do próprio batch e, em seguida, reescala e desloca o resultado por parâmetros aprendíveis γ e δ:

$$m_h = \frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}} h_i, \quad s_h = \sqrt{\frac{1}{|\mathcal{B}|}\sum_{i\in\mathcal{B}}(h_i - m_h)^2}, \quad h_i \leftarrow \gamma\,\frac{h_i - m_h}{s_h + \epsilon} + \delta. \tag{28}$$

Isso reduz a sensibilidade à inicialização e permite taxas de aprendizado maiores. A Figura 6 compara os dois tipos de bloco usados nas ResNets deste trabalho: o *BasicBlock* da ResNet-18, com dois convolucionais 3×3, e o *Bottleneck* da ResNet-50, que usa convoluções 1×1 para reduzir e restaurar o número de canais em torno de um 3×3 central, economizando computação.

A Tabela 3 detalha como as duas ResNets transformam uma entrada de $4 \times 224 \times 224$ ao longo dos estágios, da convolução inicial (*stem*) ao *pooling* médio global, evidenciando a diferença de profundidade e de número de parâmetros entre elas.

5.2 EfficientNet-B0

A EfficientNet [Tan and Le, 2019] parte da observação de que escalonar uma rede ajustando isoladamente profundidade, largura ou resolução é subótimo. A proposta

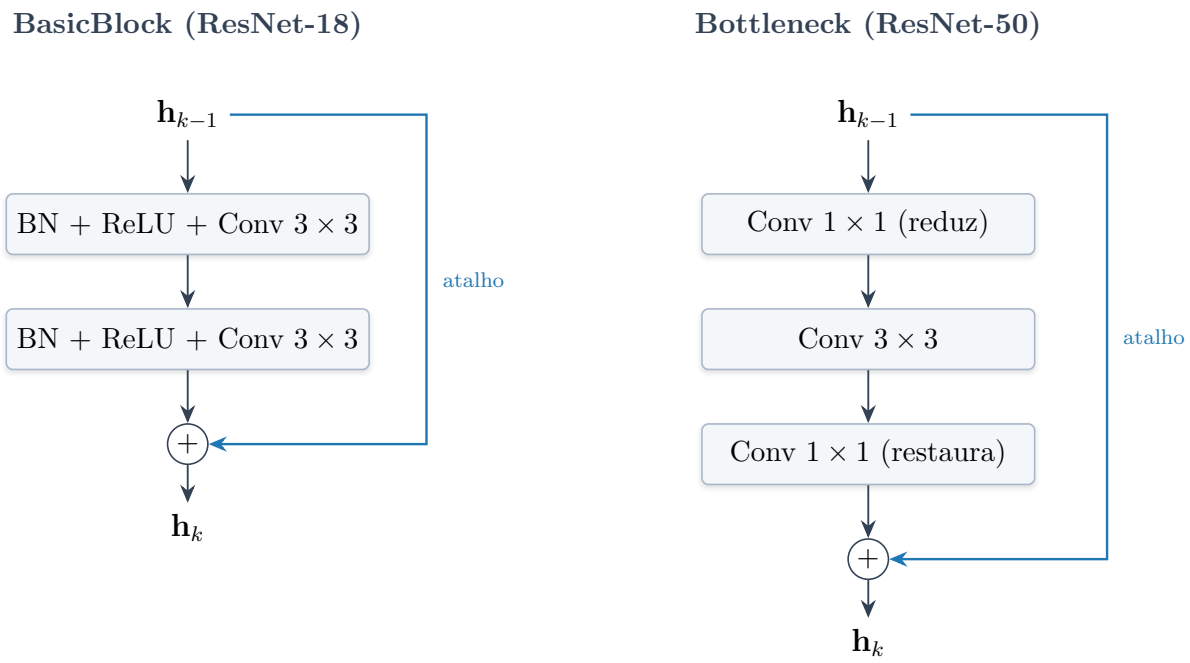

Figura 6: Blocos residuais. A conexão de atalho (azul) garante o fluxo direto do gradiente. O Bottleneck usa convoluções 1×1 para reduzir e restaurar canais em torno do 3×3, baixando o custo computacional. Fonte: adaptado de He et al. [2016].

é o escalonamento composto, que cresce as três dimensões de forma equilibrada por um único coeficiente ϕ:

$$d = \alpha^{\phi}, \quad w = \beta^{\phi}, \quad r = \gamma^{\phi}, \qquad \text{com } \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2. \tag{29}$$

O modelo base, EfficientNet-B0 ($\phi = 0$), é construído por busca de arquitetura neural e tem como peça central o bloco MBConv (*mobile inverted bottleneck convolution*). Diferentemente do Bottleneck da ResNet, que comprime os canais no miolo do bloco, o MBConv faz o oposto: ele primeiro expande o número de canais, aplica uma convolução separável em profundidade (bem mais econômica do que uma convolução comum) e só então comprime de volta. A Figura 7 ilustra essa estrutura. Embutido no bloco está o módulo de *squeeze-and-excitation*, que funciona como um mecanismo de atenção sobre os canais: ele resume cada canal a um único número via pooling global, aprende a importância relativa de cada um e reescala as ativações de acordo. Em termos formais,

$$\mathbf{s} = \text{sig}[\mathbf{\Omega}_2\, a[\mathbf{\Omega}_1\, \text{GAP}(\mathbf{h})]] \in [0,1]^{C_e}, \qquad \mathbf{h}' = \mathbf{s} \odot \mathbf{h}, \tag{30}$$

Tabela 3: Comparação ResNet-18 vs. ResNet-50 para entrada $4 \times 224 \times 224$.

Etapa	Operação	ResNet-18	ResNet-50
Stem	Conv 7×7, s=2	64×112^2	64×112^2
Pool	MaxPool 3×3, s=2	64×56^2	64×56^2
Layer 1	2×Basic / 3×Bottleneck	64×56^2	256×56^2
Layer 2	2×Basic / 4×Bottleneck	128×28^2	512×28^2
Layer 3	2×Basic / 6×Bottleneck	256×14^2	1024×14^2
Layer 4	2×Basic / 3×Bottleneck	512×7^2	2048×7^2
GAP	pooling médio global	$\mathbf{v} \in \mathbb{R}^{512}$	$\mathbf{v} \in \mathbb{R}^{2048}$
$\lvert\boldsymbol{\phi}\rvert$	–	11,3 M	24,6 M

em que $\mathbf{s}$ é o vetor de pesos por canal e $\mathbf{h}'$ é a saída recalibrada. Na prática, isso permite que a rede dê mais peso aos canais mais informativos para a tarefa, algo potencialmente útil quando os quatro mapas Orbscan contribuem de forma desigual para o diagnóstico.

Com apenas 4,3 M de parâmetros (menos de metade da ResNet-18), a EfficientNet-B0 é a arquitetura mais compacta entre as três avaliadas. Essa economia vem justamente das convoluções separáveis e do escalonamento equilibrado, e faz dela um ponto de comparação interessante: permite verificar se uma rede pequena, porém bem projetada, rivaliza com modelos maiores neste problema.

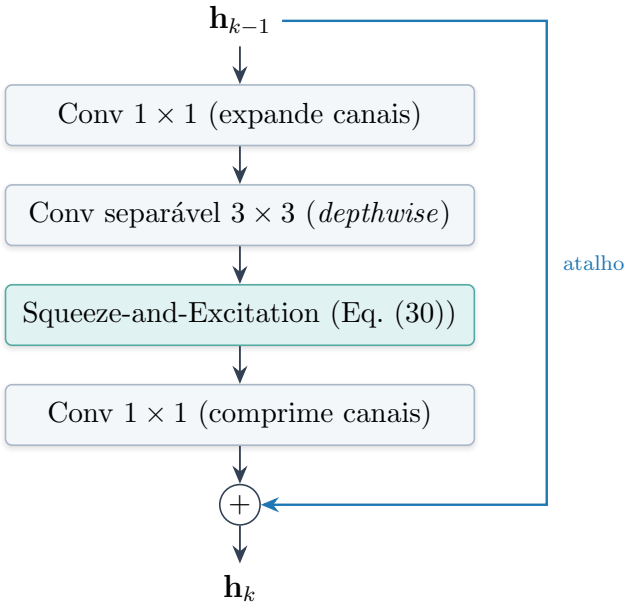

Figura 7: Bloco MBConv da EfficientNet-B0. Os canais são expandidos por uma convolução 1×1, processados por uma convolução separável em profundidade (mais barata que a convolução comum), recalibrados pelo módulo de squeeze-and-excitation (em destaque) e comprimidos de volta. A conexão de atalho (azul) preserva o fluxo do gradiente, como nas ResNets. Fonte: adaptado de Tan and Le [2019].

6 Metodologia

6.1 Formulação e visão geral da arquitetura

Seja $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ o conjunto CornOrb [Lazouni et al., 2026], com $N = 1\,454$ olhos. Cada olho fornece uma entrada $\mathbf{x}_i \in \mathbb{R}^{4\times224\times224}$, obtida pela fusão precoce [Baltrušaitis et al., 2019] dos quatro mapas Orbscan, e um rótulo $y_i \in \{0, 1\}$ atribuído por oftalmologistas. A predição é $\lambda_i = \text{sig}[f[\mathbf{x}_i, \boldsymbol{\phi}]]$ e $\hat{y}_i = \mathbb{I}[\lambda_i > 0{,}5]$. O treinamento busca os parâmetros que minimizam a entropia cruzada binária ponderada sobre o conjunto de treino:

$$\hat{\boldsymbol{\phi}} = \underset{\boldsymbol{\phi}}{\text{argmin}} \sum_{i \in \mathcal{D}_{\text{treino}}} \ell_i(\boldsymbol{\phi}). \tag{31}$$

A Figura 8 apresenta o pipeline completo, da entrada (quatro mapas brutos) à decisão clínica $\hat{y}_i \in \{0, 1\}$, organizado em três blocos. O primeiro bloco redimensiona, normaliza e empilha os mapas, formando o tensor de quatro canais. O segundo é o *backbone* convolucional pré-treinado, cuja primeira camada é adaptada para aceitar quatro canais e que produz um vetor de características $\mathbf{v} \in \mathbb{R}^D$. O terceiro converte o logit

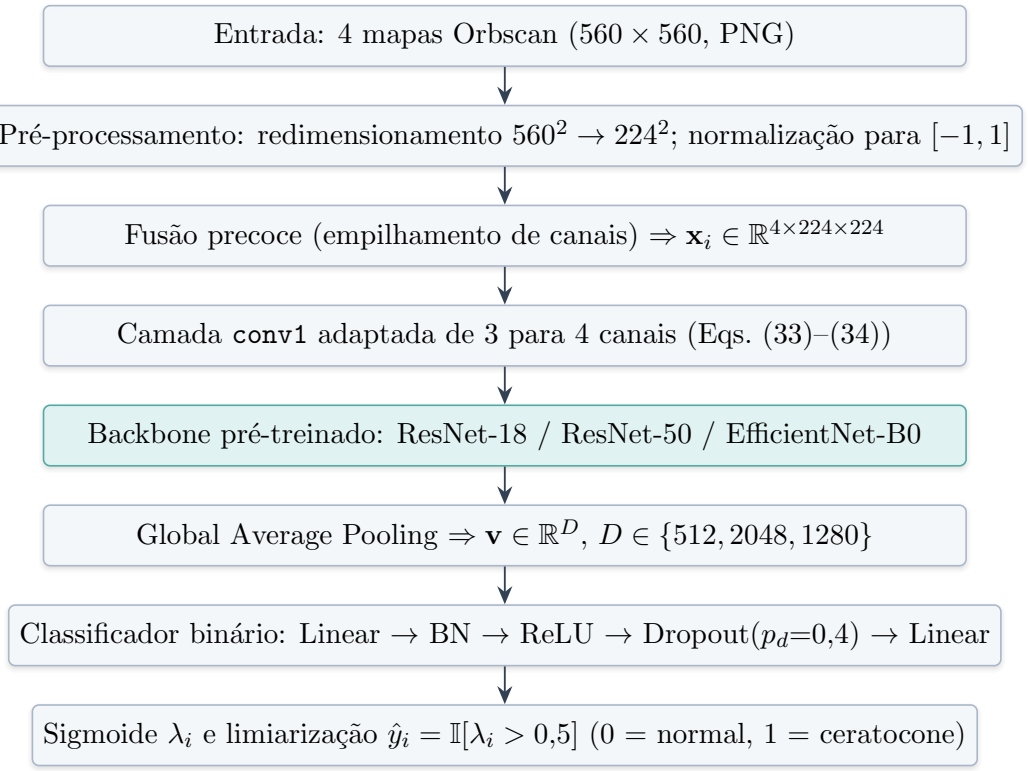

Figura 8: Pipeline da arquitetura proposta, da entrada (quatro mapas Orbscan brutos) à decisão clínica. Três backbones pré-treinados no ImageNet são comparados sob protocolo idêntico; a `conv1` é adaptada de 3 para 4 canais, preservando os pesos pré-treinados. Fonte: elaborado pelo autor.

escalar em probabilidade pela sigmoide e em decisão pelo limiar 0,5. Os três backbones são comparados sob protocolo idêntico.

6.2 Pré-processamento, fusão precoce e divisão por paciente

O pré-processamento de cada mapa segue três passos: redimensionamento bilinear de 560×560 para 224×224; normalização para o intervalo $[-1, 1]$ por $\tilde{X}_{c,i,j} = (I_{c,i,j}/255 - 0{,}5)/0{,}5$; e fusão precoce, que empilha os quatro mapas ao longo do eixo de canais,

$$\mathbf{h}_0 = \left[\tilde{X}_{\text{axial}};\ \tilde{X}_{\text{ant}};\ \tilde{X}_{\text{post}};\ \tilde{X}_{\text{pachy}}\right] \in \mathbb{R}^{4\times224\times224}. \tag{32}$$

Reunir as modalidades já na entrada permite que a rede aprenda correlações entre os quatro mapas desde as primeiras camadas, sem necessidade de arquiteturas de fusão intermediária ou tardia.

A divisão dos dados merece atenção especial. Dois olhos de um mesmo pa-

ciente são biologicamente correlacionados (o ceratocone é, por definição, uma doença bilateral, ainda que tipicamente assimétrica [Gomes et al., 2026]), de modo que distribuir olhos do mesmo paciente entre treino e teste vazaria informação e inflaria o desempenho. Para evitar isso, a divisão é feita por paciente com `GroupShuffleSplit` (semente = 42), garantindo que todos os olhos de um paciente fiquem na mesma partição [Kapoor and Narayanan, 2023]. A Tabela 4 mostra a divisão resultante, que preserva aproximadamente a proporção entre as classes em cada partição.

Tabela 4: Divisão do conjunto de dados por paciente (`GroupShuffleSplit`, semente = 42).

Partição	Olhos	Pacientes	Normal	Ceratocone
Treino	1061	537	639	422
Validação	182	95	116	66
Teste	211	112	134	77
Total	1454	744	889	565

6.3 Adaptação para quatro canais, custo ponderado e protocolo

As redes foram pré-treinadas em imagens RGB de três canais, mas a nossa entrada tem quatro. Sejam $\mathbf{\Omega}_0^{\text{ImageNet}} \in \mathbb{R}^{N\times 3\times F\times F}$ os pesos pré-treinados da primeira convolução. Construímos os novos pesos $\mathbf{\Omega}_0^{\text{novo}} \in \mathbb{R}^{N\times 4\times F\times F}$ preservando os três primeiros canais e inicializando o quarto pela média deles:

$$\mathbf{\Omega}_0^{\text{novo}}[:, 0{:}3, :, :] = \mathbf{\Omega}_0^{\text{ImageNet}}, \tag{33}$$

$$\mathbf{\Omega}_0^{\text{novo}}[:, 3, :, :] = \frac{1}{3}\sum_{c=0}^{2} \mathbf{\Omega}_0^{\text{ImageNet}}[:, c, :, :]. \tag{34}$$

A Equação (33) mantém intactos os filtros aprendidos para os três primeiros canais, e a Equação (34) fornece ao quarto canal uma inicialização informada, bem mais próxima de um detector de bordas válido do que ruído gaussiano seria. A camada de classificação original (mil classes) é substituída por um classificador binário com a estrutura $\text{Linear}(D \to D/2) \to \text{BN} \to \text{ReLU} \to \text{Dropout}(p_d) \to \text{Linear}(D/2 \to 1)$.

Como o conjunto tem mais olhos normais do que com ceratocone, a função

de custo é ponderada para penalizar mais os erros na classe minoritária. Com peso $w_+ = N_-/N_+ = 639/422 = 1{,}5142$, o custo de cada amostra torna-se

$$\ell_i = -(1 - y_i)\log\big(1 - \mathrm{sig}[f[\mathbf{x}_i, \boldsymbol{\phi}]]\big) - w_+\, y_i \log\big(\mathrm{sig}[f[\mathbf{x}_i, \boldsymbol{\phi}]]\big). \tag{35}$$

O treinamento segue um protocolo de duas fases. Na primeira fase (épocas 1 a 5), o backbone é mantido congelado e apenas a camada de classificação é treinada, com taxa de aprendizado $\alpha = 10^{-3}$. Na segunda fase (a partir da época 6), todos os parâmetros são liberados, com taxas distintas para o backbone ($\alpha_{\mathrm{bb}} = 10^{-5}$) e para a camada de classificação ($\alpha_{\mathrm{head}} = 10^{-4}$); a taxa é reduzida pela metade após sete épocas sem melhora (`ReduceLROnPlateau`) e o treinamento é interrompido por parada precoce após doze épocas sem melhora. O otimizador é o Adam, e o treinamento foi executado em GPU NVIDIA Tesla T4. A Tabela 5 resume os três modelos sob esse protocolo comum.

Tabela 5: Comparação dos três modelos sob protocolo idêntico.

Componente	ResNet-18	ResNet-50	EfficientNet-B0
Bloco base	BasicBlock	Bottleneck	MBConv + SE
Dim. do vetor latente D	512	2048	1280
Parâmetros $\lvert\boldsymbol{\phi}\rvert$	11,3 M	24,6 M	4,3 M

7 Resultados

Esta seção apresenta os resultados experimentais. Primeiro descrevemos o comportamento do treinamento, depois o desempenho no conjunto de teste e as matrizes de confusão; a interpretação aprofundada e a comparação com a literatura ficam reservadas para a Seção 7.4.

7.1 Treinamento

A Tabela 6 resume o curso do treinamento de cada modelo. As colunas indicam o número de épocas até a parada precoce, o tempo total na GPU Tesla T4, o melhor F_1 obtido no conjunto de validação e a época em que esse melhor valor ocorreu. Os três modelos atingem F_1 de validação acima de 0,96, mas diferem na rapidez de convergência:

a EfficientNet-B0 alcança o seu melhor valor logo na terceira época, a ResNet-18 na sexta, enquanto a ResNet-50 precisa de mais do que o triplo de épocas e de quase o dobro do tempo de treino para estabilizar. A Figura 9 ilustra essa dinâmica para a ResNet-18, evidenciando o descolamento entre treino e validação que motiva a parada precoce; a Figura 10 compara a perda de validação dos três modelos ao longo das épocas.

Tabela 6: Resumo do treinamento. A coluna "Melhor época" indica a época em que ocorreu o melhor F_1 de validação.

Modelo	# de épocas	Tempo (min)	F_1 val.	Melhor época
ResNet-18	18	6,8	0,9692	6
ResNet-50	35	12,8	0,9618	23
EfficientNet-B0	15	6,0	0,9697	3

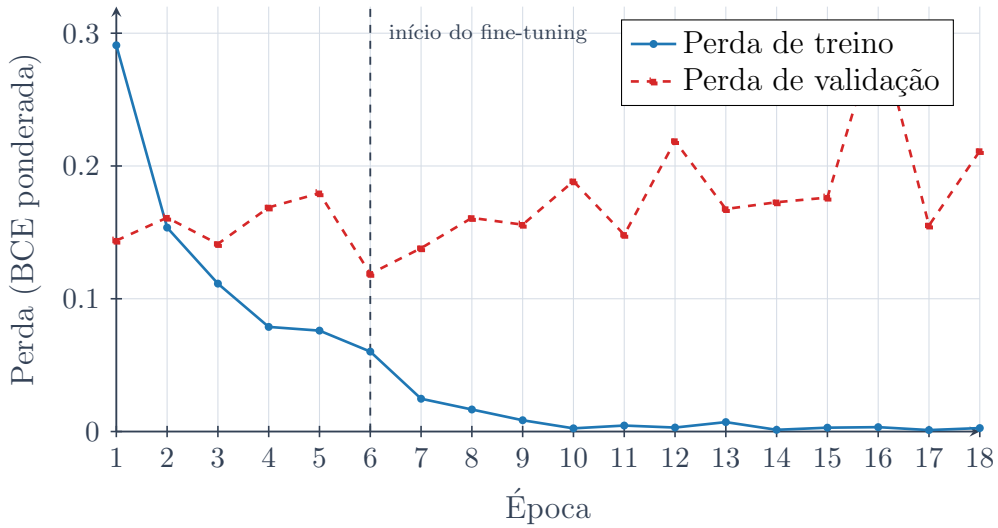

Figura 9: Curvas de perda por época da ResNet-18. A perda de treino decresce de forma contínua até valores próximos de zero, enquanto a perda de validação atinge o seu mínimo na época 6 (início da segunda fase do fine-tuning, linha tracejada) e passa a oscilar a partir daí. Esse descolamento entre as duas curvas indica o início do sobreajuste e justifica a parada precoce. Fonte: elaborado pelo autor.

7.2 Desempenho no conjunto de teste

A Tabela 7 reúne as métricas de classificação avaliadas nos 211 olhos do conjunto de teste, com o melhor valor de cada linha destacado em negrito. Além das métricas usuais (acurácia, precisão, revocação, especificidade e F_1), reportamos a AUC-ROC e a

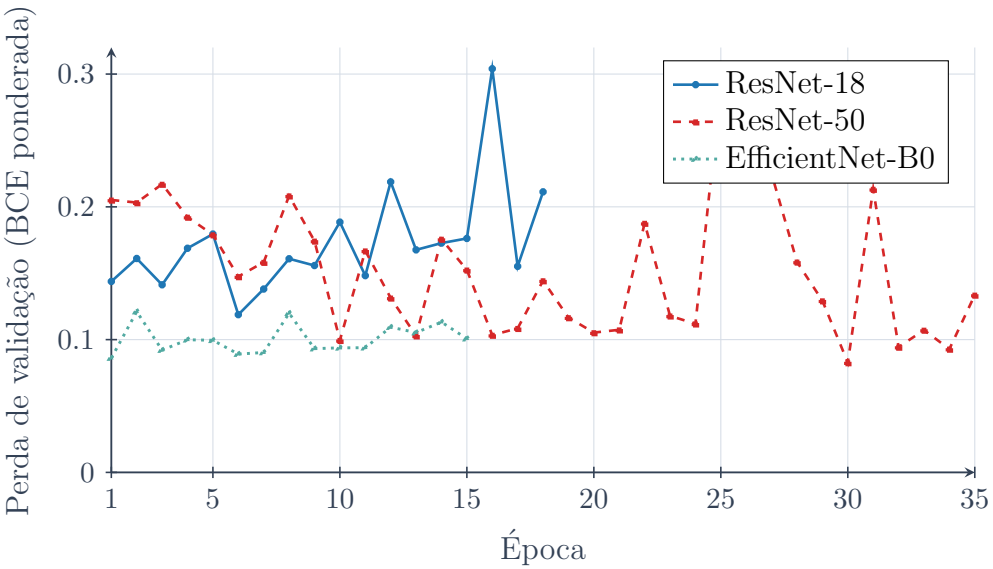

Figura 10: Perda de validação por época dos três modelos (mesmos treinamentos das Tabelas 6 e 7). A EfficientNet-B0 parte de valores baixos e estáveis e converge cedo; a ResNet-18 atinge bom desempenho já nas primeiras épocas, porém com mais oscilação; a ResNet-50 leva muito mais tempo para estabilizar, refletindo a maior dificuldade de ajustar um modelo de maior capacidade neste regime de dados moderado. Fonte: elaborado pelo autor.

AUC-PR; esta última é particularmente informativa em conjuntos desbalanceados [Saito and Rehmsmeier, 2015]. As duas últimas linhas repetem o número de parâmetros e o tempo de treino, para situar o desempenho frente ao custo. A ResNet-18 lidera em quase todas as métricas, com a única exceção da AUC-ROC e da AUC-PR, em que a ResNet-50 a supera por margem mínima.

7.3 Matrizes de confusão

A Tabela 8 apresenta, para cada modelo, a contagem de verdadeiros negativos (TN), falsos positivos (FP), falsos negativos (FN) e verdadeiros positivos (TP). A gravidade relativa de cada erro depende do tipo de problema. No apoio ao diagnóstico de ceratocone (doença progressiva cuja fase subclínica frequentemente mimetiza astigmatismo comum), deixar de detectar um caso tende a ter maior custo clínico do que um alarme falso: um falso negativo corresponde a um diagnóstico perdido, que adia a possibilidade de intervenção precoce por *cross-linking*, ao passo que um falso positivo apenas encaminha o paciente a exames confirmatórios adicionais. Por isso, na avaliação dos mo-

Tabela 7: Desempenho no conjunto de teste (211 olhos). Melhor valor de cada métrica em negrito.

Métrica	ResNet-18	ResNet-50	Eff-B0
Acurácia	**0,9858**	0,9716	0,9668
Precisão	**0,9868**	0,9733	0,9605
Revocação	**0,9740**	0,9481	0,9481
Especificidade	**0,9925**	0,9851	0,9776
F_1	**0,9804**	0,9605	0,9542
AUC-ROC	0,9965	**0,9973**	0,9924
AUC-PR	0,9952	**0,9957**	0,9899
$\lvert\boldsymbol{\phi}\rvert$	11,3 M	24,6 M	4,3 M
Tempo de treino	6,8 min	12,8 min	6,0 min

delos damos peso especial à revocação e à contagem de FN. A ResNet-18 comete apenas 2 falsos negativos, contra 4 dos outros dois modelos, o que reforça a sua adequação a esse uso clínico.

Tabela 8: Matrizes de confusão no conjunto de teste (TN: verdadeiro negativo; FP: falso positivo; FN: falso negativo; TP: verdadeiro positivo).

Modelo	TN	FP	FN	TP
ResNet-18	133	1	2	75
ResNet-50	132	2	4	73
EfficientNet-B0	131	3	4	73

7.4 Discussão

A ResNet-18 superou as demais arquiteturas em F_1, acurácia, precisão, revocação e especificidade, com apenas 2 falsos negativos em 77 casos de ceratocone, resultado de alta relevância clínica, já que um ceratocone não detectado pode evoluir sem tratamento. À primeira vista, é contraintuitivo que a ResNet-50, com o dobro da capacidade (24,6 M vs. 11,3 M de parâmetros), tenha desempenho inferior. A explicação está no tamanho do conjunto: com pouco mais de mil olhos de treino, o modelo maior tende a ajustar peculiaridades dos dados e a generalizar menos, um comportamento esperado de maior variância. A EfficientNet-B0, por sua vez, convergiu rapidamente e com apenas 4,3 M de parâmetros, mas obteve o menor F_1; o seu escalonamento composto rende mais em conjuntos de dados volumosos do que neste regime de dados moderado.

Situando os resultados na literatura, a revisão sistemática de Bodmer et al. [2024] reuniu 19 estudos de aprendizado profundo para ceratocone (10 deles na meta-análise exploratória) e relatou sensibilidade agrupada de 97,5% e especificidade de 97,2% para imagens de topografia. A ResNet-18 deste trabalho (sensibilidade = 0,974, especificidade = 0,993) posiciona-se próxima ao limite superior dessa distribuição. Em comparação com Al-Timemy et al. [2021], que empregaram uma construção híbrida de aprendizado profundo sobre múltiplos mapas de topografia corneana, os resultados são compatíveis em magnitude, ainda que datasets, equipamentos e populações difiram. Já Agharezaei et al. [2023] compararam quatro CNNs com aumento de dados via *Variational Autoencoder* (VAE), alcançando AUC-ROC de 0,99 com a VGG16, ao passo que a EfficientNet-B0 foi a de menor acurácia naquele estudo (95%). A EfficientNet-B0 treinada aqui (AUC-ROC = 0,992) atingiu desempenho competitivo sem recorrer à geração sintética de dados.

Uma ressalva importante ao comparar estudos é a heterogeneidade metodológica. Diferenças no equipamento (Orbscan vs. Pentacam vs. Galilei), na composição dos conjuntos e, sobretudo, na estratégia de divisão dos dados afetam fortemente os números relatados. A divisão em nível de paciente adotada aqui [Kapoor and Narayanan, 2023] produz estimativas mais conservadoras e realistas do que a divisão aleatória por olho, ainda comum na literatura, que tende a inflar artificialmente o desempenho reportado.

8 Conclusão

Este trabalho comparou ResNet-18, ResNet-50 e EfficientNet-B0 para a detecção automática de ceratocone a partir dos quatro mapas Orbscan do CornOrb, usando fusão precoce e divisão estrita por paciente. A ResNet-18 obteve os melhores resultados clínicos: $F_1 = 0{,}980$, AUC-ROC = 0,997, revocação = 0,974 e apenas 2 falsos negativos em 77 casos de ceratocone.

As principais contribuições do trabalho são: (1) uma descrição matemática autocontida do modelo, da regressão linear aos blocos residuais, redigida de forma acessível a um leitor sem familiaridade prévia com aprendizado profundo; (2) a comparação de três arquiteturas sob protocolo idêntico, evidenciando que a rede mais compacta entre

as ResNets generaliza melhor neste regime de dados moderado; (3) a divisão rigorosa por paciente via `GroupShuffleSplit`, que evita o vazamento de informação biológica; e (4) a adaptação informada da primeira convolução para quatro canais, preservando o conhecimento pré-treinado. Os resultados são competitivos com o estado da arte [Bodmer et al., 2024, Al-Timemy et al., 2021, Agharezaei et al., 2023].

Como desdobramentos futuros, destacam-se: (i) a geração de mapas de ativação (Grad-CAM) para interpretabilidade clínica das decisões; (ii) a fusão tardia dos mapas com as variáveis tabulares do CornOrb ($K_{\max}$, paquimetria, idade); (iii) a validação cruzada por paciente para estimativas ainda mais robustas; (iv) um estudo aprofundado de custo computacional, incluindo a latência de inferência, o uso de memória e a viabilidade de execução em hardware modesto de ambientes clínicos; e (v) uma parceria com hospitais brasileiros para a coleta de dados nacionais, com aprovação de CEP/CONEP e conformidade com a LGPD.

Referências

Zhila Agharezaei, Reza Firouzi, Samira Hassanzadeh, Siamak Zarei-Ghanavati, Kambiz Bahaadinbeigy, Amin Golabpour, Reyhaneh Akbarzadeh, Laleh Agharezaei, Mohamad Amin Bakhshali, Mohammad Reza Sedaghat, and Saeid Eslami. Computer-aided diagnosis of keratoconus through VAE-augmented images using deep learning. *Scientific Reports*, 13:20586, 2023. doi: 10.1038/s41598-023-46903-5.

Ali H. Al-Timemy, Zahraa M. Mosa, Zaid Alyasseri, Alexandru Lavric, Marcelo M. Lui, Rossen M. Hazarbassanov, and Siamak Yousefi. A hybrid deep learning construct for detecting keratoconus from corneal maps. *Translational Vision Science & Technology*, 10(14):16, 2021. doi: 10.1167/tvst.10.14.16.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: a survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019. doi: 10.1109/TPAMI.2018.2798607.

Nicolas S. Bodmer, Dylan G. Christensen, Lucas M. Bachmann, Livia Faes, Frantisek Sanak, Katja Iselin, Claude Kaufmann, Michael A. Thiel, and Philipp B. Baenninger. Deep learning models used in the diagnostic workup of keratoconus: a systematic review and exploratory meta-analysis. *Cornea*, 43(7):916–931, 2024. doi: 10.1097/ICO.0000000000003467.

Gavin Cairns and Charles N. J. McGhee. Orbscan computerized topography: attributes, applications, and limitations. *Journal of Cataract & Refractive Surgery*, 31(1):205–220, 2005. doi: 10.1016/j.jcrs.2004.09.047.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: a large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Andrea C. Ferdi, Vuong Nguyen, Daniel M. Gore, Bruce D. S. Allan, Jos J. Rozema, and Stephanie L. Watson. Keratoconus natural progression: a systematic re-

view and meta-analysis of 11,529 eyes. *Ophthalmology*, 126(7):935–945, 2019. doi: 10.1016/j.ophtha.2019.02.029.

Daniel A. Godefrooij, G. Ardine de Wit, Cuno S. Uiterwaal, Saskia M. Imhof, and Robert P. L. Wisse. Age-specific incidence and prevalence of keratoconus: a nationwide registration study. *American Journal of Ophthalmology*, 175:169–172, 2017. doi: 10.1016/j.ajo.2016.12.015.

José Álvaro P. Gomes, Farhad Hafezi, Renato Ambrósio Jr., et al. Global consensus on keratoconus and ectatic diseases—edition 2. *Cornea*, 2026. doi: 10.1097/ICO.0000000000004170. Publicado online em 21 de maio de 2026.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

Sayash Kapoor and Arvind Narayanan. Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9):100804, 2023. doi: 10.1016/j.patter.2023.100804.

Mohammed El Amine Lazouni, Leila Ryma Lazouni, Zineb Aziza Elaouaber, Mohammed Ammar, Sofiane Zehar, Mohammed Youcef Bouayad Agha, Ahmed Lazouni, Amel Feroui, Ali H. Al-Timemy, Siamak Yousefi, and Mostafa El Habib Daho. CornOrb: a multimodal dataset of Orbscan corneal topography and clinical annotations for keratoconus detection. *arXiv preprint arXiv:2603.21245*, 2026.

Simon J. D. Prince. *Understanding Deep Learning*. MIT Press, 2023. `https://udlbook.github.io/udlbook/`.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: understanding transfer learning for medical imaging. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE*, 10(3):e0118432, 2015. doi: 10.1371/journal.pone.0118432.

Radhika B. Singh, Shizuka Koh, Namrata Sharma, Michael W. Belin, and Renato Ambrósio Jr. Keratoconus. *Nature Reviews Disease Primers*, 10(1):81, 2024. doi: 10.1038/s41572-024-00565-3.

Nima Tajbakhsh, Jae Y. Shin, Suryakanth R. Gurudu, R. Todd Hurst, Christopher B. Kendall, Michael B. Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: full training or fine tuning? *IEEE Transactions on Medical Imaging*, 35(5):1299–1312, 2016. doi: 10.1109/TMI.2016.2535302.

Mingxing Tan and Quoc V. Le. Efficientnet: rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 6105–6114. PMLR, 2019.