



# SEGMENTAÇÃO DE IMAGENS DE RAIO-X DO PULMÃO USANDO REDES NEURAIS CONVOLUCIONAIS

## Projeto Final

PTC5892 – Processamento de Imagens Médicas

Natanael Magalhães Cardoso

PROFESSOR: Prof. Dr. Sergio S. Furuie



**Universidade de São  
Paulo**

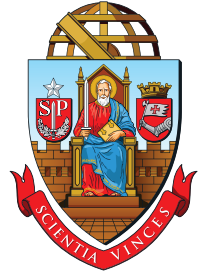
Escola Politécnica

Departamento de Eng. da  
Computação e Sist. Digitais



**UNIVERSIDADE DE SÃO PAULO**  
**ESCOLA POLITÉCNICA**

Departamento de Eng. da Computação e Sist. Digitais



# Segmentação de Imagens de Raio-X do Pulmão usando Redes Neurais Convolucionais

Natanael Magalhães Cardoso<sup>★</sup>, ✉

## I N F O

★ Departamento de Engenharia de Computação e Sistemas Digitais, Escola Politécnica, Universidade de São Paulo.

✉ contato@natanael.net

**Palavras-chave:** base de conhecimento, visão computacional, vision transformers, aprendizagem profunda, processamento de imagens médicas, raio-X, segmentação.

## R E S U M O

Este trabalho aborda o desenvolvimento e avaliação de um modelo baseado em Redes Neurais Convolucionais (CNNs) para a segmentação de imagens de raio-X do pulmão. O estudo avalia a aplicação de técnicas de Aprendizado Profundo (DL), como a Transferência de Aprendizagem e o Aumento Artificial de Dados, explorando a arquitetura EfficientUNet. O objetivo é desenvolver um modelo que, a partir das imagens de raio-X, faça a predição da segmentação do pulmão quanto da infecção de Covid-19, em duas imagens distintas. Na avaliação do modelo no conjunto de teste, foi obtido 91,35% e 88,46% na métrica de similaridade estrutural (SSIM) para imagens de segmentação do pulmão e da infecção, respectivamente.

---

## 1. INTRODUÇÃO

---

A Aprendizagem Profunda (DL, [GOODFELLOW et al., 2016](#)) é um segmento específico dentro da área de Aprendizado de Máquina (ML) e, por conseguinte, da área de Inteligência Artificial. Consiste no desenvolvimento de redes neurais artificiais que são combinadas em um número significativamente maior do que as redes neurais tradicionais. Este tipo de técnica se transformou no estado-da-arte do reconhecimento de padrões em imagens devido a um tipo específico de rede neural conhecida como convolucional. As redes neurais convolucionais (CNNs, [LECUN et al., 2015](#)), são inspiradas e propostas com certa analogia ao processamento das imagens realizadas no córtex visual de mamíferos.

O processo começa quando um estímulo visual alcança a retina e equivale a um sinal que atravessa regiões específicas do cérebro. Essas regiões são responsáveis pelo reconhecimento de cada uma dessas características correspondentes ([JOHNSON et al., 2015](#)). Os neurônios biológicos das primeiras regiões respondem pela identificação de formatos geométricos primários, enquanto neurônios das camadas finais têm a função de detectar características mais complexas, formadas pelas formas simples anteriormente reconhecidas ([VEDALDI e LENC, 2015](#)). Características com padrões muito específicos do objeto são estabelecidas depois que o procedimento se repete. De forma análoga, a CNN decompõe a tarefa de reconhecimento de um objeto em subtarefas. Para isso, durante a aprendizagem, a CNN divide a tarefa em subníveis de representação das características, posteriormente aprendendo a reconhecer novas amostras da mesma classe ([LECUN et al., 2015](#); [VEDALDI e LENC, 2015](#)). Desta forma, as CNNs são capazes de prever características complexas sem a necessidade de um pré processamento e são invariantes à escala e à rotação dos dados, o que torna essencial a classificação em imagens.

As CNNs já são amplamente utilizadas em diversas áreas do conhecimento para resolver problemas relacionados a visão computacional, como classificação de software malicioso ([REZENDE et al., 2018](#)), de plantas ([ABAS et al., 2018](#)), de imagens de satélite ([MAHDIANPARI et al., 2018](#)) e até de músicas ([SOLANO-ROJAS et al., 2020](#)). Na área médica, há uma vasta gama de utilização das CNNs, como a classificação de células cancerígenas ([KRIEGSMANN et al., 2020](#)), de tumores cerebrais ([BELAID e LOUDINI, 2020](#)), de câncer de mama ([LI et al., 2020](#)), de esclerose múltipla ([WANG e ZHANG, 2020](#)) e de eletrocardiogramas ([NONAKA e](#)

SEITA, 2020).

Neste trabalho, será usada uma arquitetura EfficientUnet para desenvolver um modelo que, a partir das imagens de raio-X, faça a predição da segmentação tanto do pulmão quanto da infecção de Covid-19, em duas imagens distintas.

---

## 2. CONJUNTO DE DADOS

---

Para treinar o modelo, é usado um conjunto de dados de imagens de raio-X do pulmão compilado por pesquisadores de Qatar University (TAHIR, CHOWDHURY, KHANDAKAR et al., 2021; TAHIR, CHOWDHURY, QIBLAWEY et al., 2022). Este conjunto de dados contém 11.956 imagens de Covid-19, 11.263 imagens de outras infecções e 10.701 imagens de pacientes saudáveis. Cada elemento do conjunto é composto por três imagens: raio-X do paciente, segmentação do pulmão e segmentação da infecção.

Assim, dois grupos de imagens foram criados: covid e não-covid. O primeiro com as 11.956 imagens de Covid-19 e o segundo com as demais imagens. Subsequentemente, o conjunto de dados foi dividido em três subconjuntos: treino (70%), validação (15%) e teste (15%).

---

## 3. MÉTODO

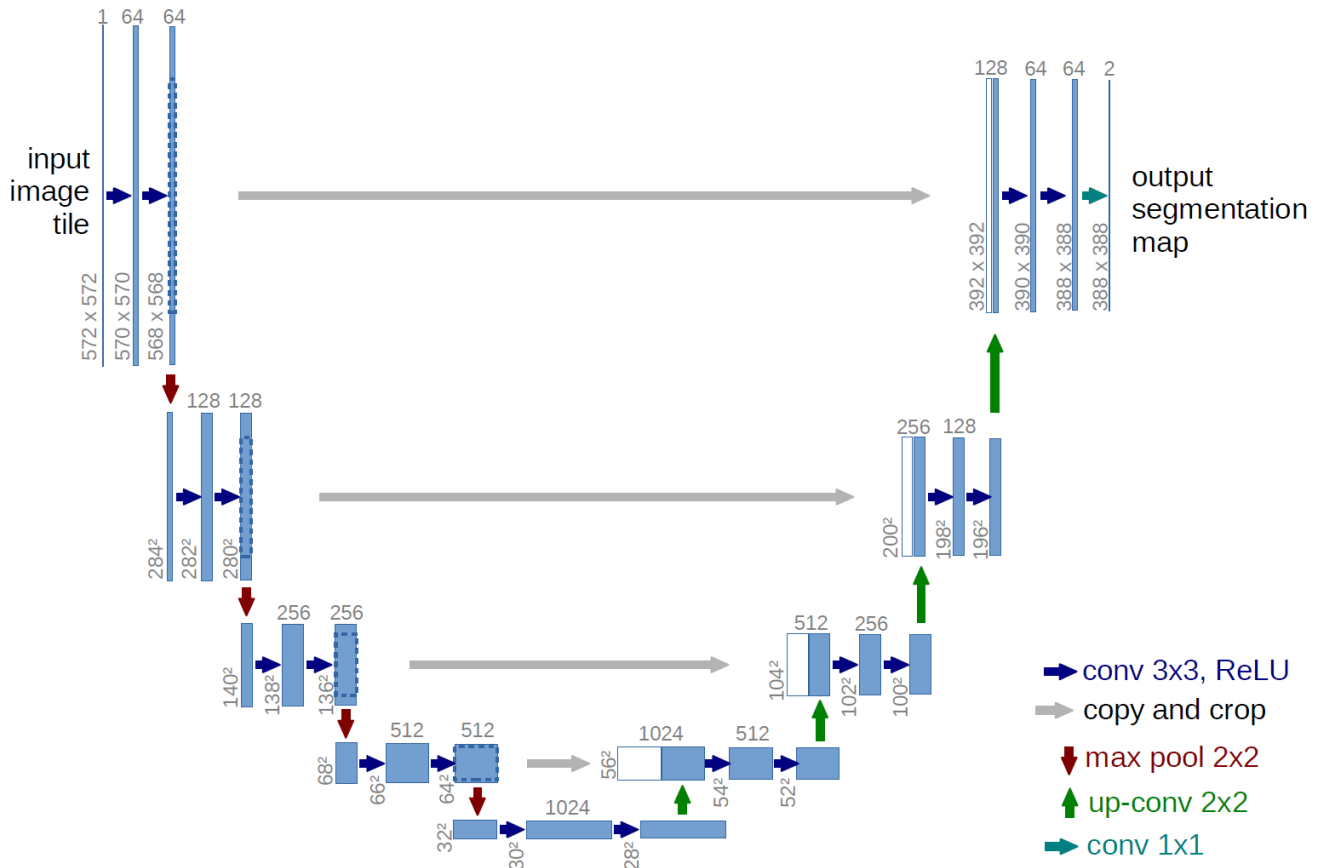
---

### 3.1. ARQUITETURAS DE REDES NEURAIAS

#### 3.1.1 UNet

A arquitetura de rede neural UNet, mostrada na Fig. 1, foi projetada para tarefas de segmentação semântica em imagens biomédicas, como a segmentação de células ou órgãos em imagens de microscopia (RONNEBERGER et al., 2015). Ela foi criada para resolver um dos principais desafios da segmentação de imagens: a necessidade de captar detalhes finos enquanto preserva a estrutura global da imagem.

A UNet é composta por duas partes principais: o caminho de contração (encoder) e o caminho de expansão (decoder). O primeiro é responsável por extrair características da imagem, utilizando uma sequência de camadas de convolução seguidas de funções de



■ **Figura 1:** Arquitetura da Rede Neural Convolucional U-Net, criada para segmentação de imagens médicas. Fonte: RONNEBERGER ET AL., 2015.

ativação (geralmente ReLU) e operações de pooling para reduzir gradualmente a resolução espacial da imagem, mas aumentar a profundidade da informação extraída. Essa seção é similar a outras arquiteturas de redes neurais convolucionais, como a VGG.

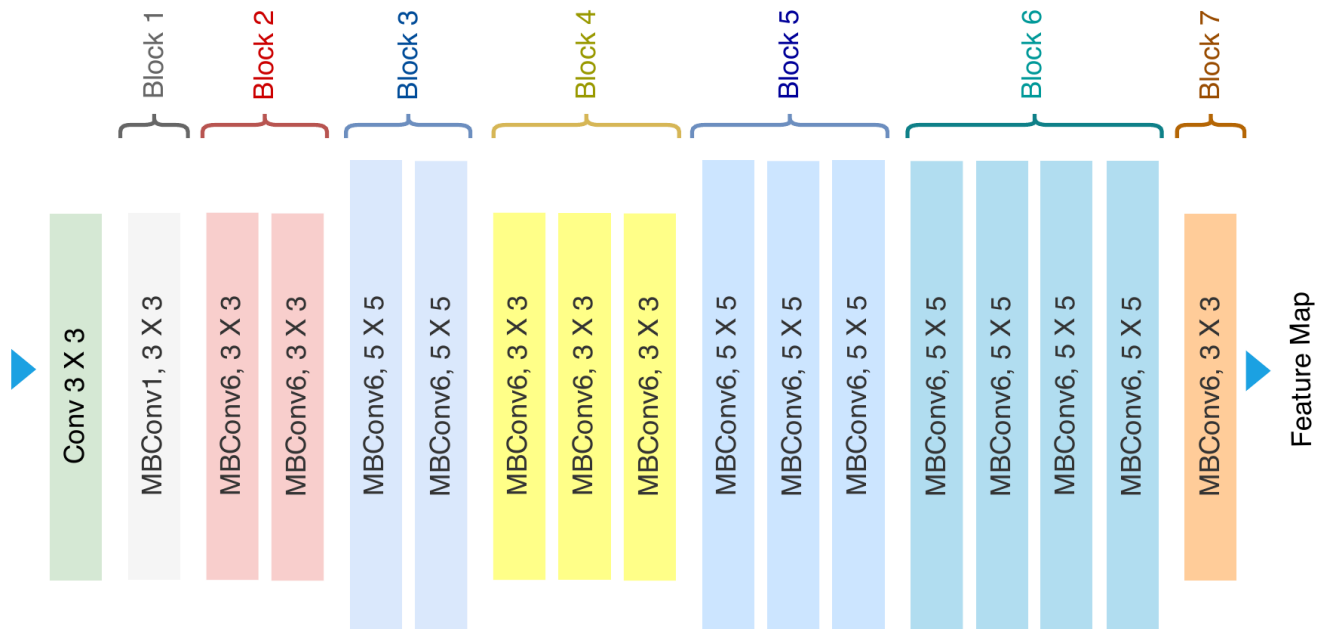
Após a extração das características, a UNet começa a expandir a imagem de volta à sua resolução original (decoder), utilizando operações de upsampling e convoluções. O diferencial é que, a cada etapa de upsampling, as informações extraídas em estágios equivalentes do encoder são concatenadas (skip connections) para ajudar na reconstrução detalhada da imagem segmentada.

A utilização de skip connections permite que a UNet combine a localização precisa de características espaciais da camada correspondente do encoder com as informações mais contextuais da camada do decoder, resultando em segmentações precisas mesmo em bordas finas ou estruturas complexas.

Comparada a arquiteturas de segmentação anteriores, a UNet é relativamente eficiente,

requerendo menos dados anotados para alcançar bons resultados. Isso é crucial em contextos biomédicos onde anotar grandes quantidades de dados pode ser difícil e caro.

### 3.1.2 EfficientNet



■ **Figura 2:** Arquitetura da Rede Neural Convolutacional Efficient. Fonte: TAN e LE, 2020.

A arquitetura EfficientNet, proposta por TAN e LE, 2020 e mostrada na Fig. 2, foi introduzida como uma abordagem para projetar CNNs mais eficientes em termos de desempenho e recursos computacionais. A principal inovação da EfficientNet é sua metodologia de escalonamento, que permite criar redes de diferentes tamanhos que mantêm um equilíbrio ótimo entre precisão e eficiência. O resultado é uma família de modelos (EfficientNet-B0 a EfficientNet-B7) que variam em tamanho e capacidade, mas mantêm uma eficiência notável em termos de precisão por FLOP (Floating Point Operation).

A EfficientNet baseia-se em um conceito chamado compound scaling, que propõe o escalonamento conjunto das três principais dimensões de uma rede neural: profundidade, largura e resolução da imagem de entrada. Em arquiteturas anteriores, o aumento do desempenho geralmente envolvia apenas uma dessas dimensões, o que poderia levar a um uso ineficiente dos recursos computacionais.

A profundidade refere-se ao número de camadas na rede. Aumentar a profundidade geralmente melhora a capacidade da rede de capturar características complexas, mas

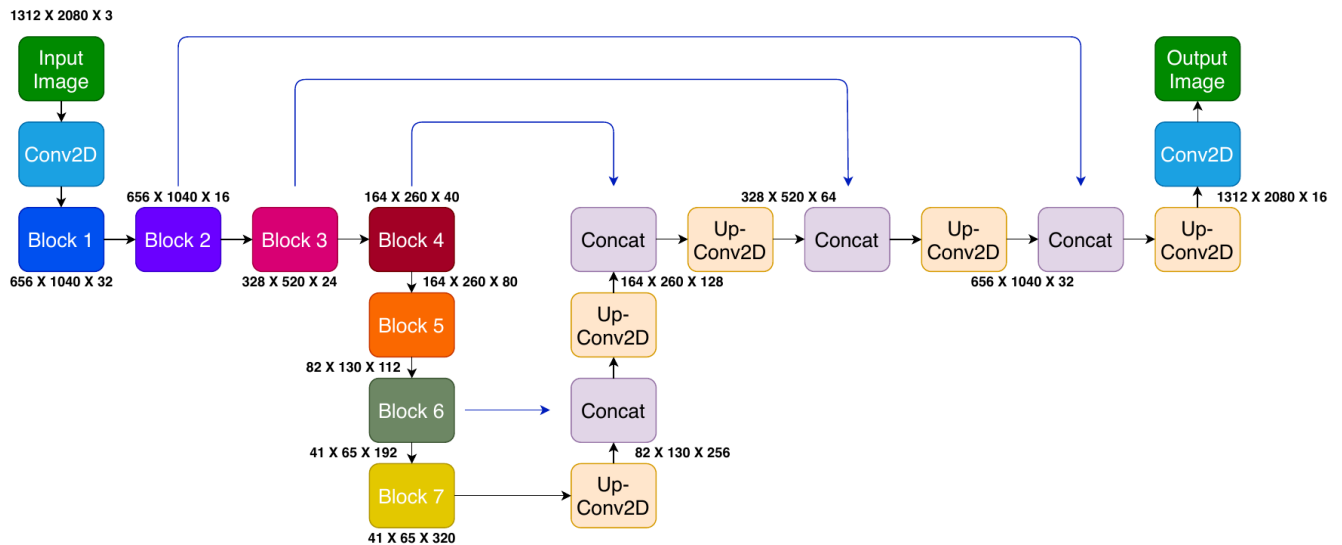
também aumenta o custo computacional e o risco de overfitting.

A largura refere-se ao número de filtros em cada camada. Aumentar a largura permite capturar mais informações em cada camada, mas também aumenta a quantidade de parâmetros e o custo computacional.

A resolução da imagem de entrada pode ser aumentada para capturar mais detalhes, mas isso também aumenta o custo computacional e o tempo de treinamento.

Ao equilibrar cuidadosamente a profundidade, a largura e a resolução, a EfficientNet oferece um desempenho superior com menos recursos computacionais em comparação com outras arquiteturas, como ResNet ou Inception. Desta forma, a EfficientNet pode ser ajustada para diferentes aplicações e limites de recursos, desde dispositivos móveis com capacidades limitadas até sistemas de ponta com recursos abundantes.

### 3.1.3 EfficientUNet



■ **Figura 3:** Arquitetura da Rede Neural Convolutiva EfficientUnet, criada para segmentação de imagens de satélite. Fonte: AHMED E SABAB, 2022.

A arquitetura EfficientUNet, mostrada na Fig. 3, é uma combinação das arquiteturas EfficientNet e UNet, projetada para melhorar a eficiência e a precisão em tarefas de segmentação de imagens, especialmente em contextos onde os recursos computacionais são limitados, mas a precisão é crucial (AHMED e SABAB, 2022). Essa arquitetura é projetada para cenários onde é necessário manter um bom desempenho com uso otimizado de recursos, como em dispositivos móveis ou em aplicações em tempo real.

A EfficientUNet utiliza a EfficientNet (Seção 3.1.2) como a parte do encoder para extrair características da imagem de entrada. A EfficientNet é escolhida por sua eficiência computacional e capacidade de capturar representações ricas da imagem com menos parâmetros em comparação com outras arquiteturas tradicionais.

A parte do decoder segue a estrutura típica da UNet (Seção 3.1.1), onde as características extraídas são progressivamente aumentadas em resolução para reconstruir uma imagem segmentada. Como na UNet original, o decoder da EfficientUNet também incorpora skip connections entre as camadas correspondentes do encoder e do decoder, ajudando a preservar detalhes importantes da imagem.

Aproveitando a eficiência da EfficientNet, a EfficientUNet oferece um modelo leve e rápido, que é particularmente útil para implementações em dispositivos com recursos limitados, como dispositivos móveis ou sistemas embarcados.

A combinação da capacidade de extração de características da EfficientNet com o mecanismo de segmentação da UNet resulta em segmentações precisas, o que é crucial em tarefas que exigem alta exatidão, como na área médica. Este trabalho usa esta arquitetura para segmentar as imagens de raio-X do pulmão.

## 3.2. TRANSFERÊNCIA DE APRENDIZAGEM

A transferência de aprendizagem (TORREY e SHAVLIK, 2010) é uma técnica onde um modelo treinado em uma tarefa é reutilizado ou ajustado para uma nova tarefa. Ao invés de treinar um modelo do zero com um grande volume de dados, a transferência de aprendizagem permite aproveitar o conhecimento previamente adquirido em outra tarefa relacionada, o que pode acelerar o processo de treinamento e melhorar a performance do modelo, especialmente quando há limitação de dados.

A transferência de aprendizagem tem se tornado uma prática fundamental no treinamento de redes neurais convolucionais (CNNs), especialmente para tarefas de visão computacional como classificação, detecção de objetos e segmentação de imagens. Um dos principais motivos é que treinar uma CNN do zero requer grandes quantidades de dados rotulados e recursos computacionais consideráveis, o que nem sempre é viável.

Neste trabalho, será testada a inicialização dos pesos provenientes de uma rede pré-



treinada usando a base de dados *ImageNet*<sup>1</sup> traz uma grande melhoria na precisão dos resultados da classificação. Essa base de dados possui milhões de imagens de objetos do cotidiano.

### 3.3. AUMENTO ARTIFICIAL DE DADOS

Aumento artificial de dados (YAEGGER et al., 1996) é a aplicação de transformações afins nas imagens a cada época no conjunto de treinamento, por exemplo rotação, reflexão, translação e mudança de escala. As matrizes das equações (1, 2, 3 e 4) definem as transformações usadas.

A transformação  $R(\theta)$  rotaciona a imagem por um ângulo  $\theta$ .

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

A Transformação  $H$  faz a reflexão horizontal da imagem.

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

A transformação  $T(x, y)$  aplica translação por  $x$  pixel na direção horizontal e  $y$  pixels na direção vertical.

$$T(x, y) = \begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

A transformação  $S(x, y)$  aplica cisalhamento na imagem.

---

<sup>1</sup><http://www.image-net.org/>

$$S(x, y) = \begin{bmatrix} 1 & x & 0 \\ y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

As transformações nas imagens são feitas remapeando as coordenadas dos píxeis originais aplicando uma combinação das matrizes das equações (1, 2, 3 e 4) em cada píxel da imagem original usando a equação (5), onde  $M$  é a matriz das transformações combinadas, calculada a partir da multiplicação das matrizes de transformação anteriores,  $(x, y)$  a coordenada do píxel da imagem original e  $(x^*, y^*)$  a coordenada transformada do píxel.

$$\begin{bmatrix} x^* \\ y^* \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5)$$

Além disso, ainda é aplicada uma interpolação bilinear como *anti-aliasing* (BOVIK, 2009; McREYNOLDS e BLYTHE, 2005). Durante o treinamento da rede, novas imagens de entrada são geradas a cada época a partir da transformação das imagens originais. Diversas transformações podem ser geradas substituindo  $M$  da equação por combinações (multiplicação matricial) das transformações das equações (1, 2, 3 e 4). Tais transformações não mudam a interpretação da classe da imagem original, pois o espaço visual é invariante a elas. Logo, o objetivo de aplicar estas transformações nas imagens de entrada da rede é deixar que o algoritmo infira tal invariância, criando, assim, uma “noção” do espaço visual, o que resulta no aumento do potencial de generalização da rede (CHOLLET, 2017; SIMARD et al., 2003). Frequentemente são relatados bons resultados com o uso desta técnica (II e DINC, 2020; MITTAL et al., 2019; NONAKA e SEITA, 2020), principalmente quando existe grande similaridade entre as classes.

### 3.4. HIPERPARÂMETROS

Nesta seção descrevemos as definições dos principais conceitos, no contexto de aprendizagem profunda, que serão úteis para o entendimento dos métodos aqui utilizados. A

função de ativação, função de custo, o otimizador, o learning rate, o número de épocas, além do número de camadas dos modelos, são importantes parâmetros responsáveis pela construção do modelo definido a seguir.

### Função de ativação

A função de ativação é responsável por adicionar não-linearidade à rede. Sem ela, a saída de uma camada seria apenas uma transformação linear dos dados de entrada e a rede não seria beneficiada pelo empilhamento de diversas camadas lineares, pois isso não aumentaria o espaço de hipóteses. Logo, a função de ativação viabiliza representações mais complexas da rede, uma vez que define a complexidade de um modelo e, conseqüentemente, sua capacidade de generalização (CHOLLET, 2017). Neste trabalho, a função  $\text{ReLU}(x) := \max(0, x)$  é usada nas camadas densas dos classificadores, a equação tangente hiperbólica é usada nas camadas densas do meta-modelo e a função Softmax (BRIDLE, 1990) foi usada na última camada, tanto dos classificadores quanto do meta-modelo.

### Função de Custo

A função de custo é utilizada com o objetivo de determinar o quão longe o modelo está do esperado, definindo a necessidade de atualização dos pesos da rede. Utilizamos a função Entropia Cruzada (*Cross-Entropy*)

$$J(\mathbf{w}) = -\frac{1}{N} \sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)] \quad (6)$$

onde  $y_n$  representa a probabilidade da classe dada pelo conjunto de treinamento do objeto  $n$  e  $\hat{y}_n$  representa a previsão da rede para este mesmo objeto.

### Otimizador

O otimizador é um algoritmo iterativo com objetivo de minimizar a função de custo. Uma escolha típica é o método de gradiente descendente e suas demais variações. Este tipo de algoritmo tem um parâmetro livre relacionado ao passo da iteração conhecido como taxa de aprendizado ou *learning rate*. Neste trabalho foram testados diversos algoritmos considerados como estado-da-arte dos otimizadores como Adam (KINGMA e BA, 2017), NAdam (DOZAT, 2015), RAdam (LIU et al., 2020) e RMSprop (HINTON et al., 2012).

### Número de Épocas

O Número de épocas se referem a quantidade de vezes que o dataset de treino foi utilizado completamente no processo de otimização iterativa da função de custo. Um número de épocas adequado é necessário para que a função de custo seja minimizada.

### Tamanho do Batch

O processo de otimização acontece em batches, cada iteração para minimizar a função custo é realizada com um número fixo de amostras, quando todas as amostras de treino são utilizadas se completa uma época.

### Unidades de neurônios na última camada

A última camada da rede antes da camada de saída é responsável por condensar toda a informação extraída da rede para o processo de classificação final. Por esta razão, a quantidade de neurônios nessa camada pode ser particularmente sensível para a performance da rede. Neste trabalho utilizamos diferentes valores de neurônios para encontrar a quantidade que pode gerar a melhor performance.

### Dropout

*Dropout* [SRIVASTAVA et al., 2014](#) é uma técnica de regularização muito utilizada em redes neurais por seu bom desempenho e baixo custo computacional. Aplicar esta regularização em uma camada consiste em eliminar aleatoriamente uma taxa dos neurônios de saída desta camada durante o treinamento, sendo geralmente escolhido um valor entre 0.2 e 0.5 para esta taxa [CHOLLET, 2017](#).

## 3.5. TREINAMENTO

Quatro diferentes modelos foram treinados usando diferentes técnicas de aprendizagem de máquina apresentadas nessa Seção. A Tabela 1 sumariza as técnicas empregadas para o treinamento de cada modelo. Na próxima Seção, será discutida a melhora de desempenho com a aplicação de cada uma das técnicas.

**Tabela 1:** Relação de modelos treinados

Modelo	Método
Baseline	Modelo treinado com inicialização aleatória de pesos
Dataaug	Modelo treinado com aumento artificial de dados (Seção 3.3)
ImageNet	Modelo treinado com transferência de aprendizagem (Seção 3.2)
Todos	Modelo treinado com aumento artificial de dados (Seção 3.3) e com transferência de aprendizagem (Seção 3.2)

## 3.6. MÉTRICAS DE AVALIAÇÃO

### 3.6.1 Coeficiente de Dice

Coeficiente de Dice é uma métrica utilizada para avaliar a similaridade entre duas amostras, sendo amplamente empregada em tarefas de segmentação de imagens. Sua definição é expressa na eq. (7).

$$Dice(A, B) = \frac{2\|A \cap B\|}{\|A\| + \|B\|} \quad (7)$$

onde  $\|A \cap B\|$  é o número de elementos (ou pixels) que estão na interseção das duas amostras (conjunto de predições e conjunto de verdadeiros positivos), enquanto  $\|A\|$  e  $\|B\|$  são os tamanhos dos conjuntos predito e real, respectivamente.

O Coeficiente de Dice mede a sobreposição entre a segmentação prevista pelo modelo e a segmentação real (referência). Ele varia de 0 a 1, onde 0 indica nenhuma sobreposição e 1 indica sobreposição perfeita. Essa métrica é particularmente útil em segmentação de imagens porque leva em consideração tanto a precisão quanto a sensibilidade, oferecendo uma visão equilibrada de quão bem o modelo está performando.

Sendo assim, o Coeficiente de Dice mede a similaridade espacial entre a segmentação prevista e a segmentação real, ou seja, quão bem as regiões identificadas pelo modelo correspondem às regiões reais na imagem. Em outras palavras, é uma medida de acurácia espacial, que avalia a qualidade da segmentação realizada pelo modelo em termos de sobreposição entre as áreas previstas e as áreas verdadeiras.

### 3.6.2 Índice de Jaccard

O Índice de Jaccard, também conhecido como Coeficiente de Jaccard ou *Intersection over Union* (IoU), é uma métrica utilizada para avaliar a similaridade entre dois conjuntos, sendo amplamente empregada na avaliação de modelos de segmentação de imagens. Essa métrica oferece uma medida clara e intuitiva da sobreposição entre a segmentação prevista pelo modelo e a segmentação real. Quanto maior a sobreposição entre as regiões previstas e as reais, maior o valor do IoU, que varia de 0 a 1, onde 0 indica nenhuma sobreposição e 1 indica sobreposição perfeita. Dessa forma, sua definição é a razão entre a interseção dos conjuntos A e B e a união dos mesmos, como mostra a eq. (8).

$$Jaccard(A, B) = IoU(A, B) = \frac{\|A \cap B\|}{\|A \cup B\|} \quad (8)$$

onde  $\|A \cap B\|$  é o número de elementos (ou pixels) na interseção entre o conjunto de predições e o conjunto de verdadeiros positivos, e  $\|A \cup B\|$  é o número total de elementos presentes na união dos conjuntos predito e real.

### 3.6.3 Índice de Similaridade Estrutural

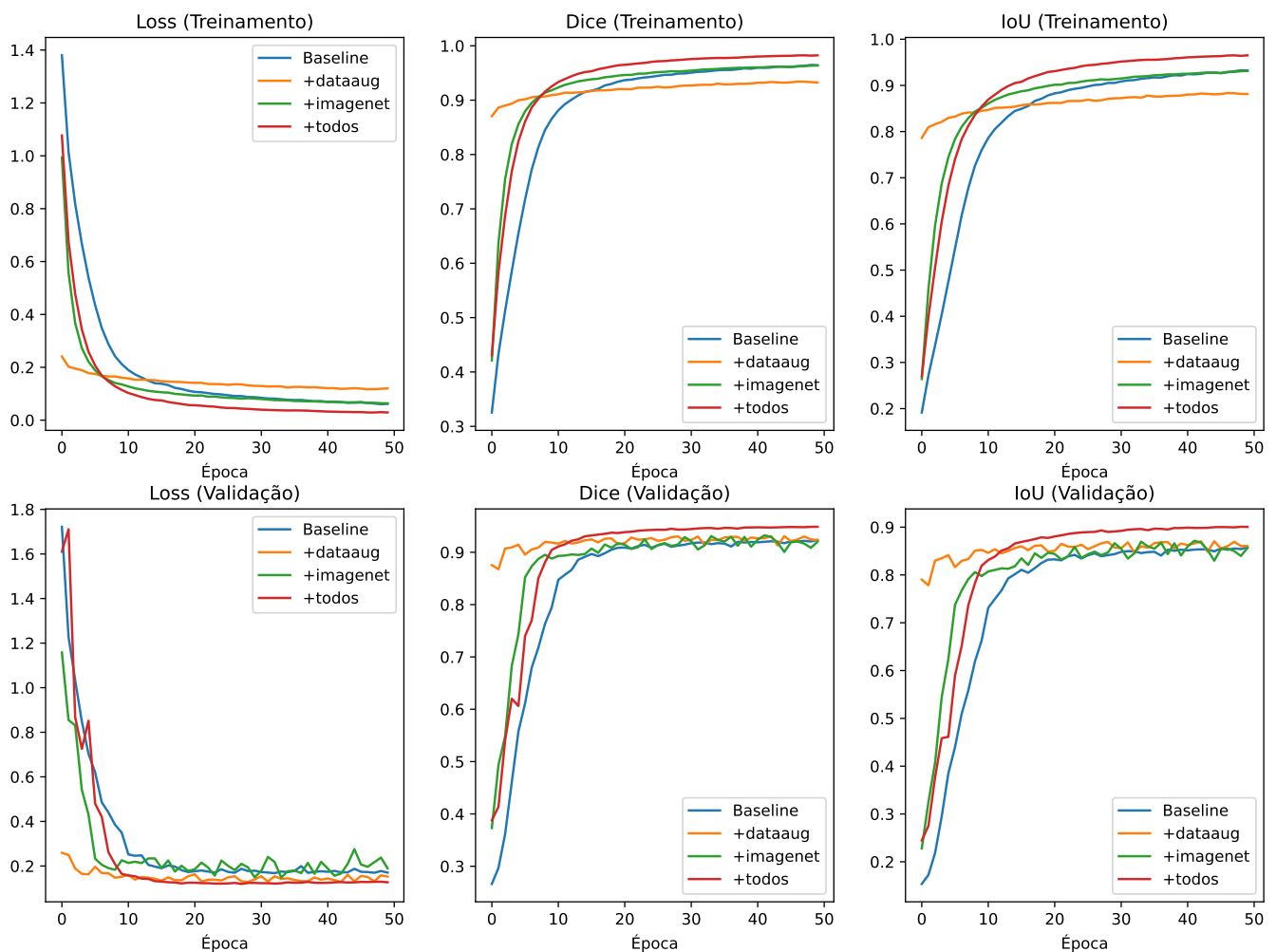
O Índice de Similaridade Estrutural (ou *Structural Similarity Index*, SSIM) é uma métrica utilizada para medir a similaridade perceptual entre duas imagens. Diferente de métricas tradicionais como MSE (*Mean Squared Error*) ou PSNR (*Peak Signal-to-Noise Ratio*), que avaliam a diferença ponto a ponto entre duas imagens, o SSIM é projetado para capturar as diferenças estruturais percebidas, considerando fatores como luminância, contraste e estrutura.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (9)$$

onde  $\mu_x$  e  $\mu_y$  são as médias das intensidades das janelas  $x$  e  $y$ ;  $\sigma_x^2$  e  $\sigma_y^2$  são as variâncias das intensidades;  $\sigma_{xy}$  é a covariância entre  $x$  e  $y$ ;  $c_1$  e  $c_2$  são constantes para estabilizar a divisão.

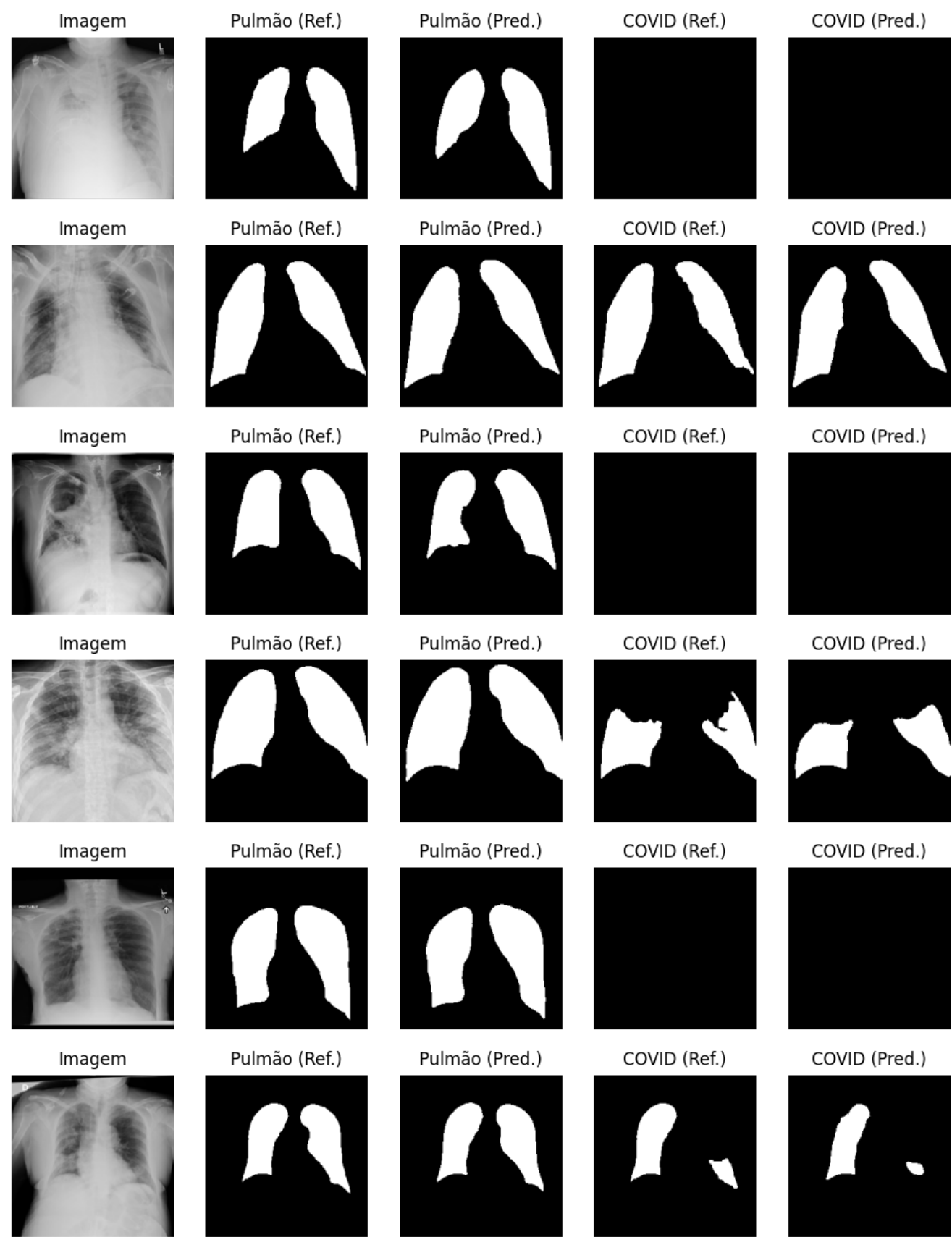
## 4. RESULTADOS

O modelo EfficientUnet (Seção 3.1.3) foi treinado no conjunto de treinamento e avaliado no conjunto de validação (informações sobre esses conjuntos estão na Seção 2). Durante o treinamento, foram monitoradas as métricas de avaliação definidas na Seção 3.6 e estão dispostas na Fig. 4, que mostra a evolução do treinamento para os modelos definidos na Seção 3.5: *baseline*, *dataaug*, *imagenet* e *todos*. É possível verificar que o modelo apresenta uma boa capacidade de generalização.



**Figura 4:** Evolução do treinamento comparada com o conjunto de validação

Para visualização dos resultados, a Fig. 5 mostra segmentações aleatórias feitas pelo modelo, comparadas com os valores de referência.



■ **Figura 5:** Predições no conjunto de teste



A Tabela 2 sumariza o desempenho do modelo no conjunto de teste. O modelo em que é aplicado tanto transferência de aprendizagem (Seção 3.2) quanto aumento artificial de dados (Seção 3.3) apresenta melhor desempenho em todas as métricas.

■ **Tabela 2:** Sumário das métricas para os diversos modelos

Modelo	Loss	Dice	IoU	SSIM (Pulmão)	SSIM (Covid)
Baseline	0.1820	0.9111	0.8386	0.8870	0.8151
+Data Aug	0.1280	0.9412	0.8863	0.8960	0.8552
+Imagenet	0.14922	0.9296	0.8674	0.9123	0.8787
+Todos	<b>0.1278</b>	<b>0.9412</b>	<b>0.8863</b>	<b>0.9135</b>	<b>0.8846</b>

---

## 5. DISCUSSÃO E CONCLUSÃO

---

O trabalho realizado demonstrou a eficácia de utilizar redes neurais convolucionais, especificamente a arquitetura UNet, combinadas com a técnica de transferência de aprendizagem para a tarefa de segmentação de imagens de raio-X do pulmão. A utilização de pesos pré-treinados na base ImageNet mostrou-se vantajosa, permitindo uma melhoria significativa nos resultados de segmentação, principalmente em contextos de dados limitados. O uso de técnicas adicionais, como aumento de dados e regularização com dropout, contribuiu ainda mais para a otimização do modelo, refletindo em métricas de desempenho superiores. Estes resultados ressaltam a importância das abordagens utilizadas, especialmente em aplicações críticas como a detecção de condições pulmonares, incluindo a COVID-19. No futuro, investigações adicionais podem explorar a generalização destas técnicas para outras modalidades de imagens médicas e para outras arquiteturas de redes neurais, visando sempre melhorar a acurácia e a robustez dos modelos.

---

## REFERÊNCIAS

---

ABAS, MOHAMAD AQIB HAQMI et al. (out. de 2018). “VGG16 for Plant Image Classification with Transfer Learning and Data Augmentation”. Em: *International Journal of Engineering & Technology* 7.4.11, p. 90. DOI: [10.14419/ijet.v7i4.11.20781](https://doi.org/10.14419/ijet.v7i4.11.20781). URL: <https://doi.org/10.14419/ijet.v7i4.11.20781>.

- AHMED, TASHIN e NOOR HOSSAIN NURI SABAB (nov. de 2022). “Classification and understanding of cloud structures via satellite images with EfficientUNet”. Em: *Authorea Preprints*. DOI: [10.1002/essoar.10507423.1](https://doi.org/10.1002/essoar.10507423.1).
- BELAID, OUIZA NAIT e MALIK LOUDINI (2020). “Classification of Brain Tumor by Combination of Pre-Trained VGG16 CNN”. Em: *Journal of Information Technology Management* 12.2, pp. 13–25. ISSN: 2008-5893. DOI: [10.22059/jitm.2020.75788](https://doi.org/10.22059/jitm.2020.75788). eprint: [https://jitm.ut.ac.ir/article\\_75788\\_e36c948ee9258c82b9398f136692f3f5.pdf](https://jitm.ut.ac.ir/article_75788_e36c948ee9258c82b9398f136692f3f5.pdf). URL: [https://jitm.ut.ac.ir/article\\_75788.html](https://jitm.ut.ac.ir/article_75788.html).
- BOVIK, ALAN C. (2009). “Basic Gray Level Image Processing”. Em: *The Essential Guide to Image Processing*. Elsevier, pp. 43–68. DOI: [10.1016/b978-0-12-374457-9.00003-2](https://doi.org/10.1016/b978-0-12-374457-9.00003-2). URL: <https://doi.org/10.1016/b978-0-12-374457-9.00003-2>.
- BRIDLE, JOHN (1990). “Training Stochastic Model Recognition Algorithms as Networks can Lead to Maximum Mutual Information Estimation of Parameters”. Em: *Advances in Neural Information Processing Systems*. Ed. por D. TOURETZKY. Vol. 2. Morgan-Kaufmann. URL: <https://proceedings.neurips.cc/paper/1989/file/0336dcbab05b9d5ad24f4333c70/Paper.pdf>.
- CHOLLET, FRANCOIS (2017). *Deep Learning with Python*. 1st. USA: Manning Publications Co. ISBN: 1617294438.
- DOZAT, TIMOTHY (2015). “Incorporating Nesterov Momentum into”. Em.
- GOODFELLOW, IAN, YOSHUA BENGIO e AARON COURVILLE (2016). *Deep Learning*. MIT Press.
- HINTON, GEOFFREY, NITISH SRIVASTAVA e KEVIN SWERSKY (2012). *On the Variance of the Adaptive Learning Rate and Beyond*.
- II, DAVID WILLIAM EDWARDS e IMREN DINC (nov. de 2020). “Classification of Protein Crystallization Images using EfficientNet with Data Augmentation”. Em: *CSBio '20: Proceedings of the Eleventh International Conference on Computational Systems-Biology and Bioinformatics*. ACM. DOI: [10.1145/3429210.3429220](https://doi.org/10.1145/3429210.3429220). URL: <https://doi.org/10.1145/3429210.3429220>.
- JOHNSON, JUSTIN, ANDREJ KARPATHY e LI FEI-FEI (nov. de 2015). “DenseCap: Fully Convolutional Localization Networks for Dense Captioning”. Em: *arXiv*. DOI: [10.48550/arXiv.1511.07571](https://doi.org/10.48550/arXiv.1511.07571).
- KINGMA, DIEDERIK P. e JIMMY BA (2017). *Adam: A Method for Stochastic Optimization*. arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG].

- KRIEGSMANN, MARK, CHRISTIAN HAAG, CLEO-ARON WEIS et al. (jun. de 2020). “Deep Learning for the Classification of Small-Cell and Non-Small-Cell Lung Cancer”. Em: *Cancers* 12.6, p. 1604. DOI: [10.3390/cancers12061604](https://doi.org/10.3390/cancers12061604). URL: <https://doi.org/10.3390/cancers12061604>.
- LECUN, YANN, YOSHUA BENGIO e GEOFFREY E. HINTON (2015). “Deep learning”. Em: *Nature* 521.7553, pp. 436–444. DOI: [10.1038/nature14539](https://doi.org/10.1038/nature14539). URL: <https://doi.org/10.1038/nature14539>.
- LI, XIA et al. (2020). “Classification of breast cancer histopathological images using interleaved DenseNet with SENet (IDSNet)”. Em: *PLoS ONE* 15. DOI: [10.1371/journal.pone.0232127](https://doi.org/10.1371/journal.pone.0232127).
- LIU, LIYUAN et al. (2020). *On the Variance of the Adaptive Learning Rate and Beyond*. arXiv: [1908.03265](https://arxiv.org/abs/1908.03265) [cs.LG].
- MAHDIANPARI, MASOUD et al. (jul. de 2018). “Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery”. Em: *Remote Sensing* 10.7, p. 1119. DOI: [10.3390/rs10071119](https://doi.org/10.3390/rs10071119). URL: <https://doi.org/10.3390/rs10071119>.
- MCREYNOLDS, TOM e DAVID BLYTHE (2005). “Antialiasing”. Em: *Advanced Graphics Programming Using OpenGL*. Elsevier, pp. 169–184. DOI: [10.1016/b978-155860659-3.50012-3](https://doi.org/10.1016/b978-155860659-3.50012-3). URL: <https://doi.org/10.1016/b978-155860659-3.50012-3>.
- MITTAL, ANSH et al. (dez. de 2019). “Data augmentation based morphological classification of galaxies using deep convolutional neural network”. Em: *Earth Science Informatics* 13.3, pp. 601–617. DOI: [10.1007/s12145-019-00434-8](https://doi.org/10.1007/s12145-019-00434-8). URL: <https://doi.org/10.1007/s12145-019-00434-8>.
- NONAKA, NAOKI e JUN SEITA (dez. de 2020). “Electrocardiogram Classification by Modified EfficientNet with Data Augmentation”. Em: *2020 Computing in Cardiology Conference (CinC)*. Computing in Cardiology. DOI: [10.22489/cinc.2020.063](https://doi.org/10.22489/cinc.2020.063). URL: <https://doi.org/10.22489/cinc.2020.063>.
- REZENDE, EDMAR et al. (2018). “Malicious Software Classification Using VGG16 Deep Neural Network’s Bottleneck Features”. Em: *Advances in Intelligent Systems and Computing*. Springer International Publishing, pp. 51–59. DOI: [10.1007/978-3-319-77028-4\\_9](https://doi.org/10.1007/978-3-319-77028-4_9). URL: [https://doi.org/10.1007/978-3-319-77028-4\\_9](https://doi.org/10.1007/978-3-319-77028-4_9).

- RONNEBERGER, OLAF, PHILIPP FISCHER e THOMAS BROX (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv: 1505.04597 [cs.CV]. URL: <https://arxiv.org/abs/1505.04597>.
- SIMARD, PATRICE Y., DAVE STEINKRAUS e JOHN C. PLATT (2003). “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis”. Em: *Proceedings of the Seventh International Conference on Document Analysis and Recognition - Volume 2*. ICDAR '03. USA: IEEE Computer Society, p. 958. ISBN: 0769519601. DOI: 10.5555/938980.939477.
- SOLANO-ROJAS, BRAULIO, RICARDO VILLALÓN-FONSECA e GABRIELA MARÍN-RAVENTÓS (2020). “Alzheimer’s Disease Early Detection Using a Low Cost Three-Dimensional Densenet-121 Architecture”. Em: *Lecture Notes in Computer Science*. Springer International Publishing, pp. 3–15. DOI: 10.1007/978-3-030-51517-1\_1. URL: [https://doi.org/10.1007/978-3-030-51517-1\\_1](https://doi.org/10.1007/978-3-030-51517-1_1).
- SRIVASTAVA, NITISH et al. (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. Em: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- TAHIR, ANAS M., MUHAMMAD E. H. CHOWDHURY, AMITH KHANDAKAR et al. (dez. de 2021). “COVID-19 infection localization and severity grading from chest X-ray images”. Em: *Comput. Biol. Med.* 139, p. 105002. ISSN: 0010-4825. DOI: 10.1016/j.combiomed.2021.105002.
- TAHIR, ANAS M., MUHAMMAD E. H. CHOWDHURY, YAZAN QIBLAWEY et al. (2022). *COVID-QU-Ex Dataset*. DOI: 10.34740/KAGGLE/DSV/3122958. URL: <https://www.kaggle.com/dsv/3122958>.
- TAN, MINGXING e QUOC V. LE (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv: 1905.11946 [cs.LG].
- TORREY, LISA e JUDE SHAVLIK (2010). “Transfer learning”. Em: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, pp. 242–264.
- VEDALDI, ANDREA e KAREL LENC (2015). “MatConvNet: Convolutional Neural Networks for MATLAB”. Em: *Proceedings of the 23rd ACM International Conference on Multimedia*. MM '15. Brisbane, Australia: Association for Computing Machinery, pp. 689–692. ISBN: 9781450334594. DOI: 10.1145/2733373.2807412. URL: <https://doi.org/10.1145/2733373.2807412>.

- WANG, SHUI-HUA e YU-DONG ZHANG (jun. de 2020). “DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple Sclerosis Classification”. Em: *ACM Trans. Multimedia Comput. Commun. Appl.* 16.2s. ISSN: 1551-6857. DOI: [10.1145/3341095](https://doi.org/10.1145/3341095). URL: <https://doi.org/10.1145/3341095>.
- YAEGER, LARRY, RICHARD LYON e BRANDYN WEBB (1996). “Effective Training of a Neural Network Character Classifier for Word Recognition”. Em: *Proceedings of the 9th International Conference on Neural Information Processing Systems*. NIPS’96. Denver, Colorado: MIT Press, pp. 807–813.



**Universidade de São Paulo**

Escola Politécnica

Departamento de Eng. da Computação e Sist. Digitais