



# DETECÇÃO DE COVID-19 EM RADIOGRAFIAS DO TÓRAX

## Projeto Final

PSI5790 – Aprendizado Profundo para Visão Computacional

Natanael Magalhães Cardoso

NUSP: 8914122



**Universidade de São Paulo**

Escola Politécnica

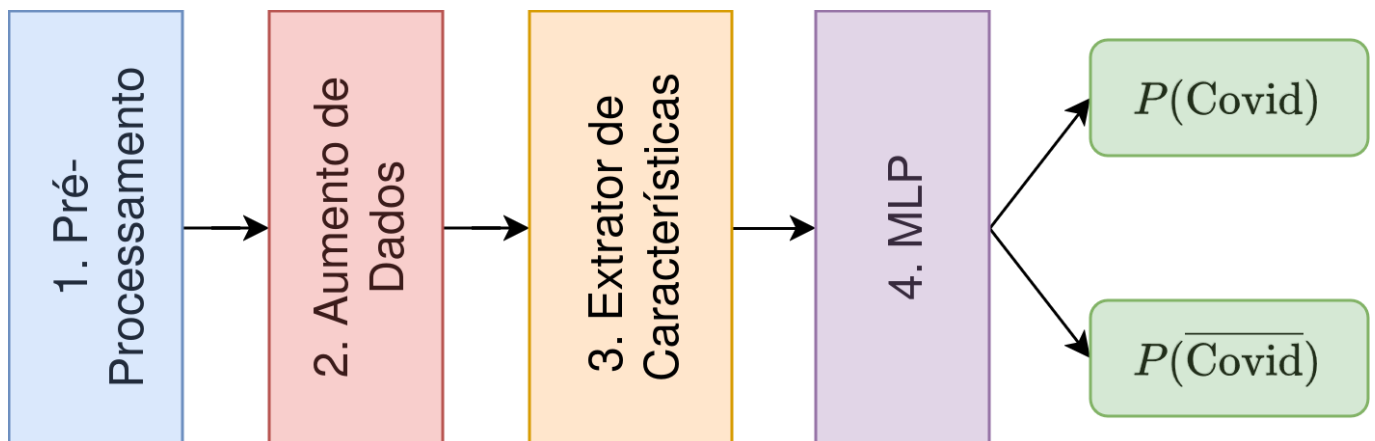
Departamento de Eng. da  
Computação e Sist. Digitais

# 1 Introdução

As Redes Neurais Convolucionais (CNN, [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](#)). De forma análoga, a CNN decompõe a tarefa de reconhecimento de um objeto em subtarefas. Para isso, a CNN divide a tarefa em subníveis de representação das características ([LeCUN et al., 2015](#)). Desta forma, as CNNs são capazes de prever características complexas e são invariantes à escala e à rotação dos dados, o que as torna essenciais na classificação em imagens.

## 2 Metodologia

A arquitetura da rede neural convolucional projetada é esquematizada no diagrama da Fig. 1. Ela é composta por camadas agrupadas em quatro blocos: pré-processamento, aumento de dados, extrator de características e camadas densamente conectadas (MLP). Cada um desses blocos serão discutidos no decorrer dessa Seção.



**Figura 1:** Arquitetura da Rede Neural, composta por camadas agrupadas em 4 blocos, pré-processamento, aumento de dados, extrator de características e camadas densamente conectadas (MLP), retornando as probabilidades para presença (ou não) de Covid-19.

As imagens são inicialmente carregadas como matrizes (uma única banda) de números inteiros sem sinal (unsigned int de 8 bits). O primeiro bloco, transforma as imagens em tensores de 3 bandas e aplica o pré-processamento dependendo do extrator de características

utilizado. Os pixels são escalonados para o intervalo -1 a 1 para ResNet e para o intervalo 0 a 1 para as demais arquiteturas. Esses são os mesmos pré-processamentos utilizados pelos autores das arquiteturas.

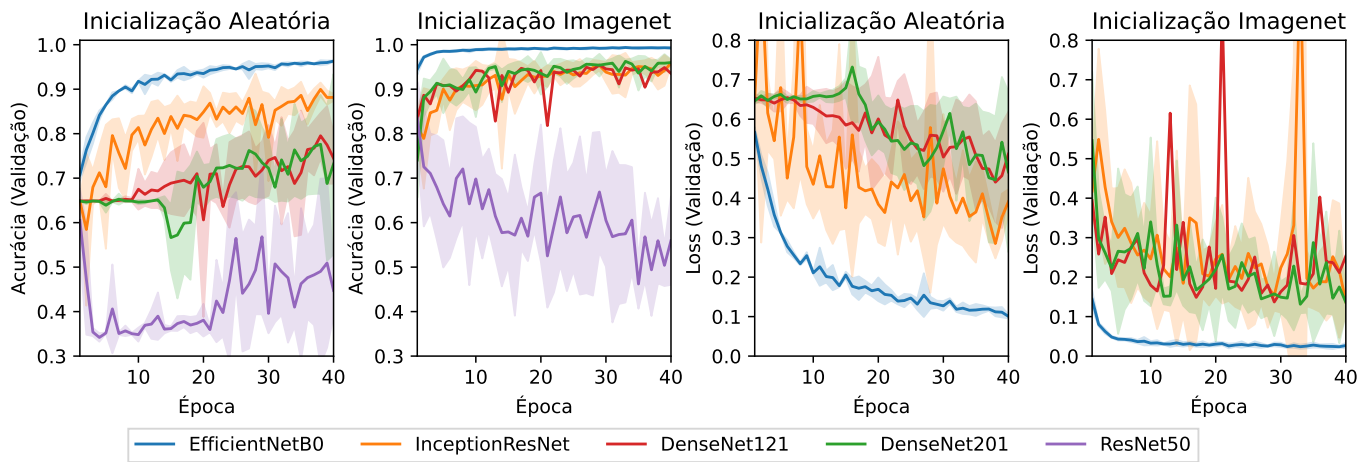
O segundo bloco, é composto por camadas de aumento de dados, que produzem transformações afins aleatoriamente nas imagens de entrada durante o treinamento. Essa técnica é utilizada para aumentar a capacidade de generalização do modelo, evitando que ele “decore” os padrões visuais do conjunto de treinamento. As transformações aplicadas são: espelhamento horizontal, escalonamento, variação de contraste e variação de brilho.

O terceiro bloco é composto um extrator de características, que é uma arquitetura de rede neural convolucional designada para extrair e codificar os padrões visuais da imagem de entrada. Foram selecionadas 5 arquiteturas para serem utilizadas: EfficientNet-B0 (TAN e LE, 2020), ResNet-50 (HE et al., 2016), Inception ResNet (SZEGEDY et al., 2016), DenseNet-121 e DenseNet-201 (HUANG et al., 2018). A saída do extrator de característica foi reduzida aplicando GlobalAveragePooling, gerando um vetor unidimensional para cada exemplo do conjunto de dados. Na Seção 3, é mostrado o desempenho de cada uma delas.

Por fim, o quarto bloco, composto por camadas densamente conectadas (*Multi Layer Perceptron – MLP*), recebem o vetor de características visuais e faz a predição final. Na última camada é usada a função sigmoid como ativação e nas camadas internas é usado ReLU. Foram utilizadas seis configurações diferentes de camadas densas: (1) uma única camada com 1024 unidades, (2) uma única camada com 512 unidades, (3) uma camada com 512 unidades seguida por outra com 256, (4) uma camada com 256 unidades seguida por outra com 256, (5) uma camada com 512 unidades seguida por outra com 512 e (6) nenhuma camada densa.

### 3 Resultados e Discussão

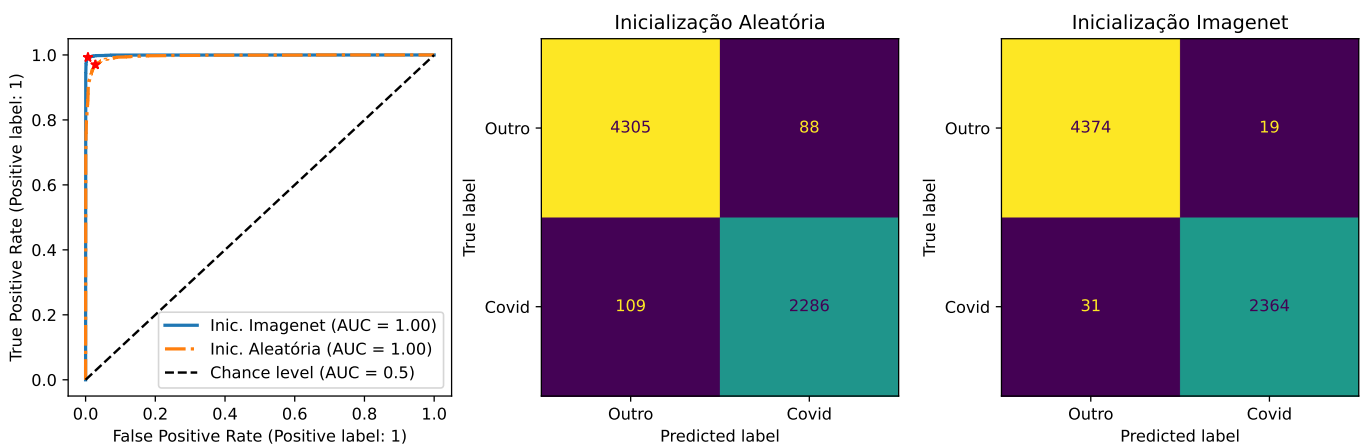
Inicialmente, foram testadas combinações das arquiteturas selecionadas com configurações de blocos MLP. A Fig. 2 mostra o desempenho dos modelos treinados no conjunto de validação agregados por arquitetura. Tanto para inicialização aleatória, quanto para inicialização com pré-treino na Imagenet, a arquitetura EfficientNet B0 superou as demais e, portanto, as análises seguintes refem-se ao modelo utilizando esta arquitetura.



**Figura 2:** Desempenho de várias arquiteturas convolucionais no conjunto de validação. Da esquerda para direita, os painéis mostram os valores de acurácia e da função de custo para inicialização aleatória e transferência de aprendizagem a partir de um modelo treinado na Imagenet. A área sombreada representa um desvio padrão.

Após avaliar os modelos, o que possuiu menor loss no conjunto de validação foi escolhido. Para ambos os casos (inicialização aleatória ou por transferência de aprendizagem), o melhor resultado foi obtido sem utilizar camadas ocultas no bloco MLP. Os modelos foram treinados com otimizador Adam inicializado com taxa de aprendizagem  $4 \times 10^{-5}$ .

A avaliação do modelo no conjunto de teste se deu pela análise da curva ROC e das matrizes de confusão. É notável que o modelo pré-treinado na Imagenet possui maior capacidade preditiva do que o treinado com inicialização aleatória dos pesos. Os valores da área abaixo da curva ROC obtidos foram 0.9924487 e 0.9999885 para inicialização aleatória e transferência de aprendizagem, respectivamente, que no gráfico, foram arredondados para 1



**Figura 3:** Da esquerda para direita: curvas ROC e matrizes de confusão para modelo pré-treinado e inicialização aleatória, respectivamente.

## 4 Conclusão

Neste estudo, foi feita uma análise comparativa do impacto da técnica de transferência de aprendizagem para inicialização dos pesos em uma rede neural convolucional pré-treinada na Imagenet. Ao avaliar os modelos no conjunto de teste usando as curvas ROC e a matriz de confusão, foi constatado que esta técnica garante um aumento significativo na capacidade preditiva do modelo, como discutido na avaliação

Assim, a partir deste estudo, conclui-se que é possível treinar um modelo de alta capacidade preditiva para Covid-19 utilizando redes neurais convolucionais pré-treinadas na Imagenet

## Referências

- HE, KAIMING et al. (2016). “Deep Residual Learning for Image Recognition”. Em: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- HUANG, GAO et al. (2018). *Densely Connected Convolutional Networks*. arXiv: [1608.06993 \[cs.CV\]](https://arxiv.org/abs/1608.06993).
- 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).
- 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>.
- SZEGEDY, CHRISTIAN et al. (2016). *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. arXiv: [1602.07261 \[cs.CV\]](https://arxiv.org/abs/1602.07261).
- TAN, MINGXING e QUOC V. LE (2020). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*. arXiv: [1905.11946 \[cs.LG\]](https://arxiv.org/abs/1905.11946).



**Universidade de São Paulo**

Escola Politécnica

Departamento de Eng. da Computação e Sist. Digitais