

# Detectando casos de COVID a partir de imagens de Tomografia

Eric Sticchi Zambom

Juliana Félix Carvalho

## Resumo

O SARS-CoV-2 é um novo coronavírus, responsável pela pandemia de COVID-19, o qual já infectou e causou a morte de milhares de pessoas em todo o mundo. Neste artigo exploramos um método de CNN (Convolutional Neural Network), onde analisamos as tomografias computadorizadas de pacientes reais em hospitais do estado de São Paulo para a identificação de infectados pelo SARS-CoV-2.

## COVID-19

De acordo Nogueira (2020) o coronavírus SARS-CoV-2 é o agente etiológico da COVID-19, uma doença altamente infecciosa que afeta principalmente o trato respiratório inferior, cujos primeiros casos foram relatados na China em 2019. Desde então, o vírus se espalhou por todo o mundo, causando uma pandemia que resulta em consequências catastróficas para a saúde e a economia globais, além dos milhões de mortes e sequelas permanentes em muitos sobreviventes, especialmente em países em desenvolvimento como o Brasil.

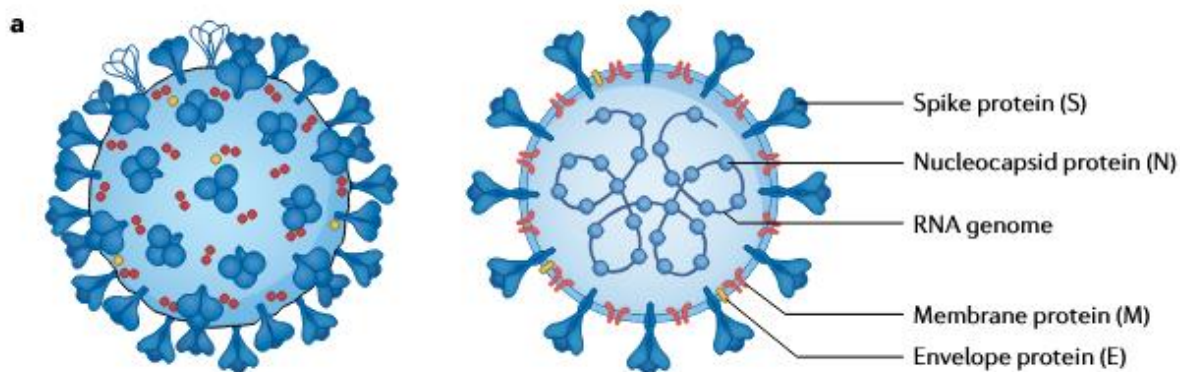
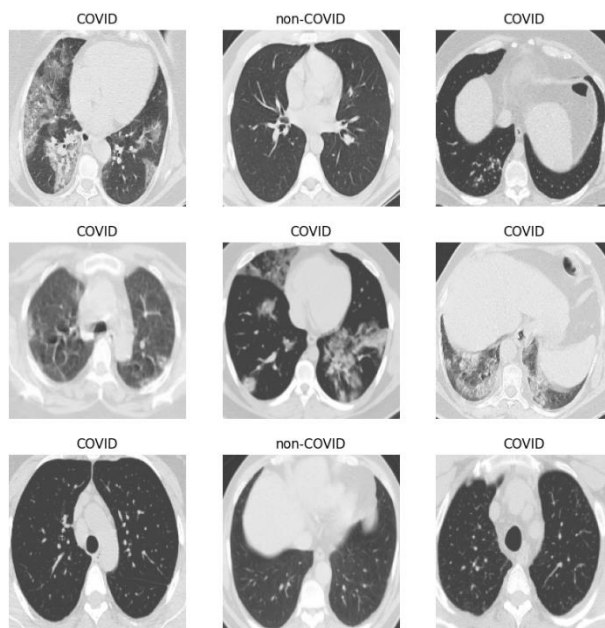


Figura 1 - Estrutura Sars-CoV-2  
Fonte: Adaptado de LAMERS, Mart M (2022)

## Dados Utilizados

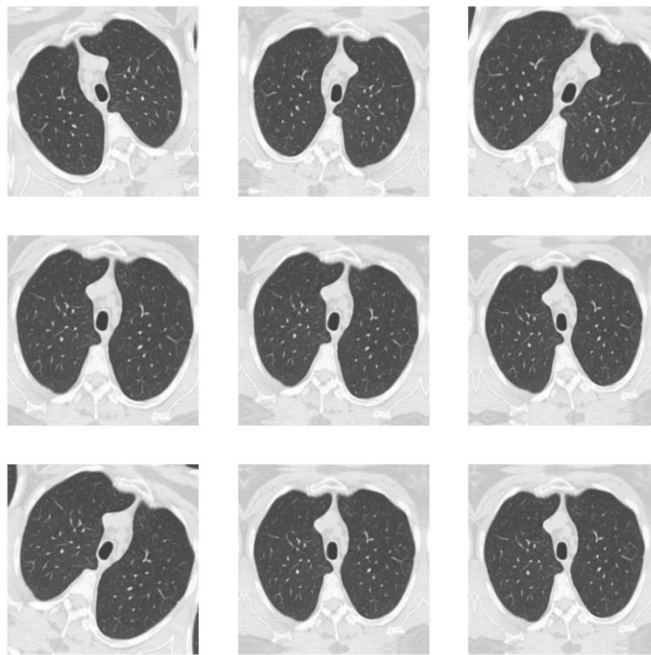
De acordo com Soares...(2020) este conjunto de dados (dataset) de tomografia computadorizada (TC) de acesso público para SARS-CoV-2, contém 1252 exames de TC positivos para infecção por SARS-CoV-2 (COVID-19) e 1230 exames de TC de pacientes não infectados por SARS-CoV-2, totalizando 2482 exames de TC. Esses dados foram coletados de pacientes reais em hospitais de São Paulo, Brasil.



*Figura 2 - Amostra das Imagens do Dataset*  
*Fonte: Soares...(2020)*

## Pré-Processamento de Imagens

Segundo WANG, Jason...(2017) as transformações tradicionais consistem em usar uma combinação de transformações afins para manipular os dados de treinamento. Portanto, para cada imagem de entrada, geramos outras imagens “duplicadas” que são deslocadas, ampliadas/reduzidas, rotacionadas ou invertidas,. Tanto a imagem quanto suas duplicatas são inseridas na rede neural. Para um conjunto de dados de tamanho N, geramos um conjunto de dados de tamanho 10N.



*Figura 3 - Imagens com tranformações simples*  
*Fonte: Soares...(2020)*

### **CNN (Convolutional Neural Network)**

Em concordância com LEE, Hagyeong...(2019) utilizamos uma separação de treinamento para o modelo de 0.2 (20%) para validação e 0.8 (80%) para o treinamento.

Dada a natureza simples de um classificador de imagens com apenas 2 parâmetros, as limitações propostas pelo ambiente de execução (Google Colab) e a falta de orçamento para a utilização de uma GPU (Graphics Processing Unit) dedicada, montamos uma rede neural estruturada da seguinte maneira:

Model: "sequential\_5"

Layer (type)	Output Shape	Param #
sequential_4 (Sequential)	(None, 250, 250, 3)	0
rescaling_5 (Rescaling)	(None, 250, 250, 3)	0
conv2d_9 (Conv2D)	(None, 250, 250, 16)	448
max_pooling2d_9 (MaxPooling2D)	(None, 125, 125, 16)	0
conv2d_10 (Conv2D)	(None, 125, 125, 32)	4,640
max_pooling2d_10 (MaxPooling2D)	(None, 62, 62, 32)	0
conv2d_11 (Conv2D)	(None, 62, 62, 64)	18,496
max_pooling2d_11 (MaxPooling2D)	(None, 31, 31, 64)	0
dropout_1 (Dropout)	(None, 31, 31, 64)	0
flatten_3 (Flatten)	(None, 61504)	0
dense_6 (Dense)	(None, 128)	7,872,640
dense_7 (Dense)	(None, 2)	258

Total params: 7,896,482 (30.12 MB)

Trainable params: 7,896,482 (30.12 MB)

Non-trainable params: 0 (0.00 B)

Figura 4 - Estrutura da CNN  
Fonte própria

O modelo foi treinado por 50 épocas, com as imagens fixadas em um tamanho de 250 x 250 pixels.

```

Epoch 31/50
63/63 ————— 214s 3s/step - accuracy: 0.8910 - loss: 0.2508 - val_accuracy: 0.9093 - val_loss: 0.2134
Epoch 32/50
63/63 ————— 202s 3s/step - accuracy: 0.8868 - loss: 0.2454 - val_accuracy: 0.8730 - val_loss: 0.2449
Epoch 33/50
63/63 ————— 171s 3s/step - accuracy: 0.8844 - loss: 0.2551 - val_accuracy: 0.8548 - val_loss: 0.3329
Epoch 34/50
63/63 ————— 162s 3s/step - accuracy: 0.7578 - loss: 0.5598 - val_accuracy: 0.7117 - val_loss: 0.5304
Epoch 35/50
63/63 ————— 210s 3s/step - accuracy: 0.7369 - loss: 0.5394 - val_accuracy: 0.8831 - val_loss: 0.2882
Epoch 36/50
63/63 ————— 201s 3s/step - accuracy: 0.8792 - loss: 0.2934 - val_accuracy: 0.8810 - val_loss: 0.2604
Epoch 37/50
63/63 ————— 169s 3s/step - accuracy: 0.8767 - loss: 0.2861 - val_accuracy: 0.8891 - val_loss: 0.2514
Epoch 38/50
63/63 ————— 162s 3s/step - accuracy: 0.8917 - loss: 0.2492 - val_accuracy: 0.7177 - val_loss: 0.7730
Epoch 39/50
63/63 ————— 171s 3s/step - accuracy: 0.8705 - loss: 0.3167 - val_accuracy: 0.9073 - val_loss: 0.2355
Epoch 40/50
63/63 ————— 195s 3s/step - accuracy: 0.9003 - loss: 0.2473 - val_accuracy: 0.8750 - val_loss: 0.2946
Epoch 41/50
63/63 ————— 173s 3s/step - accuracy: 0.8751 - loss: 0.2908 - val_accuracy: 0.9012 - val_loss: 0.2282
Epoch 42/50
63/63 ————— 170s 3s/step - accuracy: 0.9108 - loss: 0.2390 - val_accuracy: 0.9173 - val_loss: 0.2131
Epoch 43/50
63/63 ————— 161s 3s/step - accuracy: 0.9020 - loss: 0.2275 - val_accuracy: 0.9093 - val_loss: 0.2207
Epoch 44/50
63/63 ————— 213s 3s/step - accuracy: 0.9040 - loss: 0.2328 - val_accuracy: 0.9012 - val_loss: 0.2419
Epoch 45/50
63/63 ————— 172s 3s/step - accuracy: 0.9033 - loss: 0.2187 - val_accuracy: 0.9234 - val_loss: 0.2018
Epoch 46/50
63/63 ————— 162s 3s/step - accuracy: 0.8991 - loss: 0.2272 - val_accuracy: 0.9274 - val_loss: 0.1907
Epoch 47/50
63/63 ————— 202s 3s/step - accuracy: 0.9043 - loss: 0.2324 - val_accuracy: 0.8891 - val_loss: 0.2402
Epoch 48/50
63/63 ————— 203s 3s/step - accuracy: 0.9004 - loss: 0.2267 - val_accuracy: 0.9254 - val_loss: 0.1989
Epoch 49/50
63/63 ————— 162s 3s/step - accuracy: 0.9114 - loss: 0.2146 - val_accuracy: 0.8952 - val_loss: 0.2508
Epoch 50/50
63/63 ————— 170s 3s/step - accuracy: 0.9117 - loss: 0.2133 - val_accuracy: 0.9173 - val_loss: 0.1972

```

Figura 5 - Treinamento da CNN  
Fonte: Própria

## Resultados

Após a fase de pré-processamento dos dados, com o aumento da base de dados através de transformações simples, o modelo apresentou uma melhora considerável.

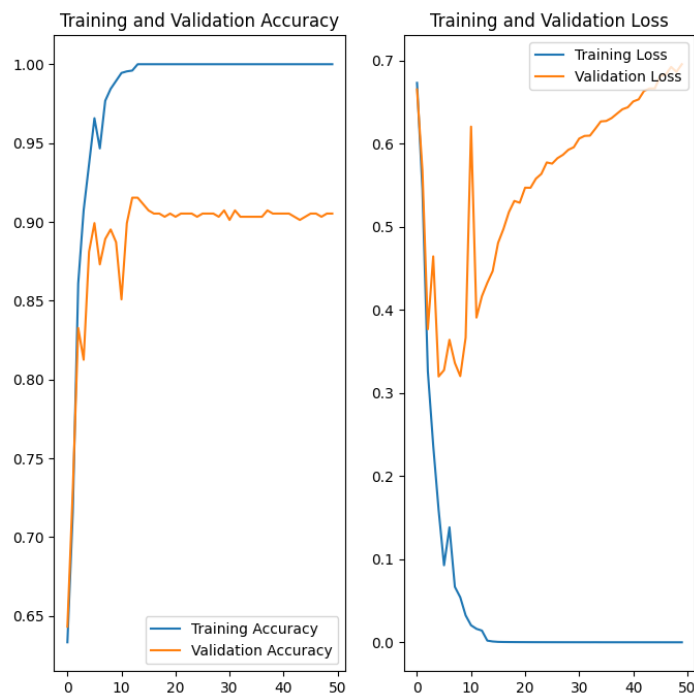


Figura 6 - Modelo de CNN sem pré-processamento  
Fonte: Própria

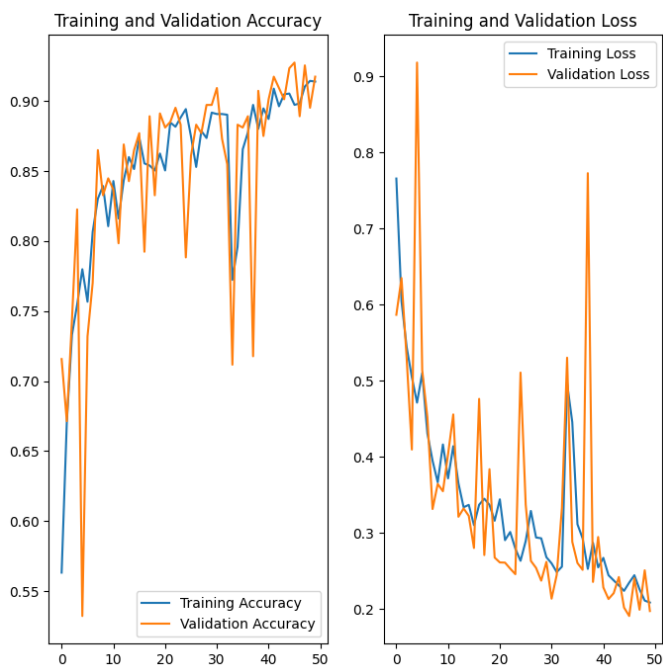
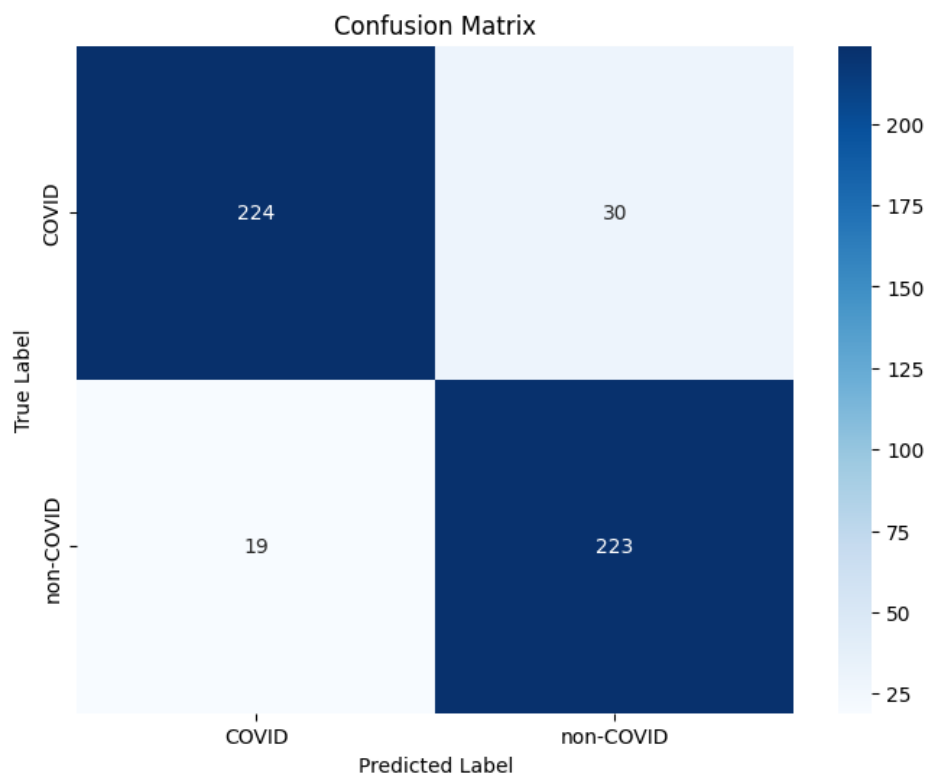


Figura 7 - Modelo CNN depois do pré-processamento de imagens  
Fonte: Própria

Dadas as limitações técnicas previstas o modelo teve de ser limitado a 50 épocas e o dataset utilizado não puderam ser expandidos para uma escala de consideração comercial. No entanto os resultados obtidos indicam que o modelo de classificação de imagens apresentou um desempenho satisfatório. A acurácia de 0,9117 no conjunto de treinamento e de 0,9173 no conjunto de validação, aliadas aos baixos valores de *loss*, sugerem uma boa capacidade de generalização, sem indícios significativos de sobreajuste.



*Figura 8 - Matriz de Confusão*  
*Fonte: Própria*

Esses resultados demonstram que a arquitetura e os hiperparâmetros adotados foram adequados para o problema proposto. Ainda assim, estudos futuros podem explorar ajustes adicionais no modelo ou técnicas de regularização e aumento de dados, visando aprimorar ainda mais o desempenho e a robustez da classificação.

## Referências

- KIM, Changhyun; LEE, Sangjin. CONAN: Markov Image-Based Preprocessing for Robust Screenshot Detection Using Deep Learning. *Digital Object Identifier* 10.1109/ACCESS.2025.3630498, recebido em 9 de outubro de 2025, aceito em 3 de novembro de 2025.
- LEE, Hagyeong; SONG, Jongwoo. Introduction to convolutional neural network using Keras; an understanding from a statistician. *Communications for Statistical Applications and Methods*, 2019, Vol. 26, No. 6, p. 591–610. DOI: 10.29220/CSAM.2019.26.6.591.
- MUMUNI, Alhassan; MUMUNI, Fuseini. CNN Architectures for Geometric Transformation-Invariant Feature Representation in Computer Vision: A Review. *SN Computer Science*, 2021, 2(340). DOI: 10.1007/s42979-021-00735-0.
- WANG, Jason; PEREZ, Luis. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. Stanford University, 2017.
- MIKOŁAJCZYK, Agnieszka; GROCHOWSKI, Michał. Data augmentation for improving deep learning in image classification problem. *Department of Control Systems Engineering, Faculty of Electrical and Control Engineering, Gdańsk University of Technology*.
- NOGUEIRA, José Vagner Delmiro. CONHECENDO A ORIGEM DO SARS-COV-2 (COVID 19). **Revista Saúde e Meio Ambiente**, v. 11, n. 2, p. 115-124, 2020.
- Soares, Eduardo, Angelov, Plamen, Biaso, Sarah, Higa Froes, Michele, and Kanda Abe, Daniel. "SARS-CoV-2 CT-scan dataset: A large dataset of real patients CT scans for SARS-CoV-2 identification." medRxiv (2020). doi: <https://doi.org/10.1101/2020.04.24.20078584>.
- Angelov, P., & Soares, E. (2020). Towards explainable deep neural networks (xDNN). *Neural Networks*, 130, 185-194.
- LAMERS, Mart M.; HAAGMANS, Bart L. SARS-CoV-2 pathogenesis. **Nature reviews microbiology**, v. 20, n. 5, p. 270-284, 2022.