

Otimizando uma CNN baseada em DenseNet para o diagnóstico de COVID19

Gabriel de Jesus S. Costa¹, Marcus Vinicius Oliveira¹, Mario Freitas¹, Matheus de Lima Bessa¹, Geraldo Braz Junior¹, João Dallyson S. de Almeida¹, Anselmo C. Paiva¹

¹Núcleo de Computação Aplicada – Universidade Federal do Maranhão (UFMA)
65.080-805 – São Luís – MA – Brasil

{gabriel.jesus, marcus.oliveira, mario.freitas, matheus.levy}@discente.ufma.br
{gerald, paiva, Jdallyson}@nca.ufma.br

Abstract. *The main method utilized to diagnose COVID-19 it's the RT-PCR, but this method takes time to generate results, so it is necessary to have methods for a quick and effective diagnosis of the virus. One of these methods that proves to be very efficient is the use of Convolutional Neural Networks, that can diagnose the disease through chest x-ray images or CT scans. Thus, our work evaluates the use of the Tree-structured Parzen Estimator algorithm to optimize and build a CNN model specialized in the diagnosis of COVID-19 based on the DenseNet architecture. Thus the model constructed in this paper, got a promising result with an accuracy of 96% and an AUC of 0,96, showing the effectiveness of the optimization algorithm in the construction of the network.*

Resumo. *O principal método utilizado para o diagnóstico de COVID-19 é o RT-PCR, mas esse método ainda leva de 1 a 3 dias para gerar resultados, por isso se fazem necessários métodos que façam um diagnóstico rápido e efetivo do vírus. Um desses métodos que se mostra bastante eficiente é a utilização de Convolutional Neural Networks, que podem fazer o diagnóstico da doença por meio de radiografias do tórax ou tomografias computadorizadas. Assim, o nosso trabalho avalia a utilização do algoritmo Tree-structured Parzen Estimator para realizar a otimização de um modelo CNN especializado no diagnóstico de COVID-19 baseada na arquitetura DenseNet. Assim, o modelo construído neste trabalho atingiu um resultado promissor com uma acurácia de 96%, demonstrando a efetividade da utilização do algoritmo de otimização na construção da rede.*

1. Introdução

O principal método utilizado para o diagnóstico de COVID-19 é o real-time reverse transcription polymerase chain reaction (RT-PCR), que é feito utilizando amostras retiradas da nasofaringe ou da orofaringe. Ainda que esse seja o teste referência para o diagnóstico de COVID-19, ele leva um longo tempo para gerar resultados e vários estudos mostram que os resultados iniciais do teste para pacientes com COVID-19 resultam em falsos negativos [Alsharif e Qurashi, 2021]. Assim, torna-se importante a utilização de um método auxiliar no diagnóstico do vírus que seja rápido e possua uma boa precisão, como a utilização de Convolutional Neural Networks (CNN) em imagens de raio-x.

As CNNs são redes neurais parecidas em seu funcionamento com as Artificial Neural Networks (ANN). ANNs são sistemas de processamento computacional que funcionam de forma parecida com um sistema nervoso [O'Shea e Nash, 2015]. Eles são formados por um grande número de nós interconectados, denominados neurônios, que aprendem por meio de uma informação dada como entrada para melhorar o resultado em sua saída. A principal diferença entre as CNNs e as ANNs é que as redes convolucionais têm a possibilidade de aceitar imagens e vídeos como sua entrada, fazendo com que essa rede então seja mais indicada para trabalhos com esse tipo de mídia. Existem diversos modelos diferentes de CNN, sendo diferenciados por sua arquitetura e sua especialidade.

Vários trabalhos recentes utilizam CNNs para o diagnóstico de COVID-19 através de imagens de raio-x ou tomografia computadorizada (CT). Aslan et al, (2021) utiliza segmentação automática de radiografias do tórax através de uma rede ANN, passando essas imagens a cinco tipos de CNN diferentes para extração de características e por fim utilizando quatro tipos de classificadores diferentes para classificação das imagens entre COVID-19, pneumonia viral e normal. Já em seu outro trabalho Aslan et al, (2021) propõe duas arquiteturas utilizando como base a rede AlexNet, sendo que a segunda arquitetura também conta com a utilização de uma camada de Bidirectional Long Short-Term Memories (BiLSTM), que leva em consideração características temporais em seu treinamento. Costa et al, (2021) avalia o uso de redes transformers para o diagnóstico de covid em imagens de CT. Al-Rakhami et al, (2021) apresenta uma arquitetura que combina redes CNN e RNN para o diagnóstico de COVID-19 através de radiografias do tórax, classificando entre COVID-19, pneumonia e normal. Foram utilizadas quatro modelos CNN para a extração de características e utilizada a RNN para a classificação das imagens, tendo como sua melhor arquitetura a VGG19. Pavlova et al, (2021) apresenta em seu trabalho uma modificação da rede COVID-Net, a COVID-Net CXR-2 para classificação em 2 classes, sendo elas casos positivos e negativos de COVID-19. Além disso, no trabalho também é apresentado um novo dataset público de imagens de tomografia do tórax que é composto por 19203 imagens, divididas em casos positivos e negativos de COVID-19.

Neste trabalho desenvolvemos e avaliamos a construção de uma rede especializada para o diagnóstico de COVID-19 baseada na arquitetura Densely Connected Convolutional Network (DenseNet) tendo sua construção regida por um processo de otimização usando o algoritmo Tree-structured Parzen Estimator (TPE).

O trabalho é dividido nas seguintes seções: A seção 2 descreve a metodologia proposta, a seção 3 consiste na apresentação e discussão dos resultados encontrados e a seção 4 apresenta a conclusão do trabalho.

2. Metodologia

A metodologia proposta neste trabalho consiste na avaliação da construção de uma rede especializada em diagnóstico de COVID-19 através de radiografias do tórax com base na arquitetura DenseNet utilizando um processo de otimização. O diagrama que representa o método proposto está representado na Figura 1, que treina e avalia diversas arquiteturas diferentes montadas com a ajuda do algoritmo TPE.

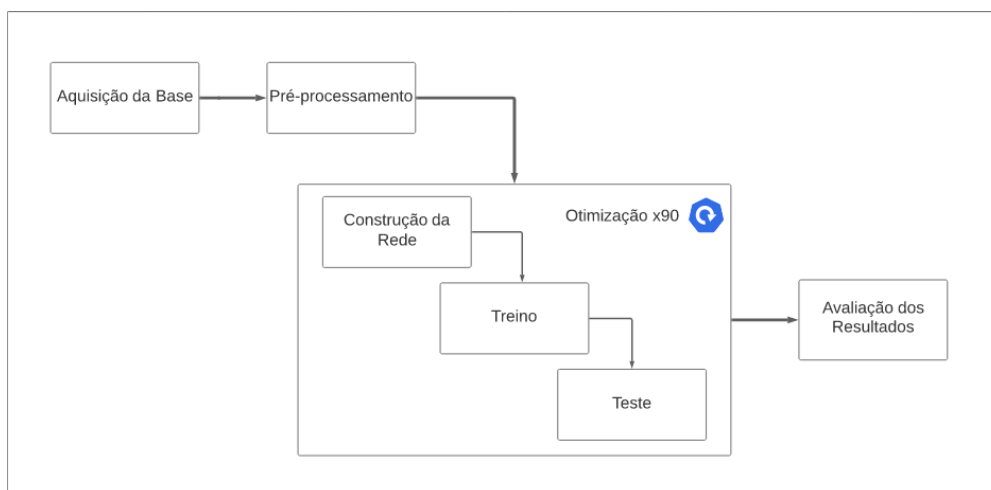


Figura 1. Diagrama da metodologia proposta.

A primeira etapa do método é a aquisição e pré-processamento da base que é utilizada nos experimentos. Os dados são então utilizados no treinamento de uma rede baseada em DenseNet, otimizada via TPE. Ao final de cada *trial* de otimização o resultado é analisado, e caso cumpra os requisitos, a rede é salva.

2.1. Aquisição da Base e Pré-Processamento

A base de dados utilizada para esse estudo foi a COVIDx CXR-2 [Pavlova et al, 2021]. A base contém imagens de radiografias do tórax pertencentes a duas classes, negativa para COVID-19 e positiva para COVID-19. Para a realização dos experimentos realizados nesta pesquisa, foi mantida a mesma configuração da base trabalho original para fins de comparação com os trabalhos relacionados. Sendo assim, a base é dividida em dois subsets, treino e teste, o de treino contendo um total de 30482 imagens, sendo 16490 classificadas como positiva e 13992 classificadas como negativa. Já o subset de teste contém 400 imagens, sendo 200 positivas e 200 negativas. Na Figura 2 podem ser vistos alguns exemplos de imagens presentes no dataset.

Na etapa de pré-processamento, a partir da base de dados, foram criados os datasets de treino, validação e teste. O dataset de validação foi criado utilizando 20% da base de treino e para os dois datasets foram aplicados redimensionamento das imagens para 224x224 e embaralhamento das imagens, sem nenhum outro tipo de pré-processamento. Já no dataset de testes, foi aplicado apenas o redimensionamento.

2.2. Construção e Otimização da Rede

Neste trabalho é utilizada uma rede baseada na arquitetura DenseNet para a classificação de radiografias do tórax entre duas classes, casos positivos ou negativos de COVID-19.

A *Dense Convolutional Network (DenseNet)* [Huang et al, 2017] foi uma das primeiras redes neurais convolucionais a implementar o treinamento de camadas mais profundas. Com o aprofundamento das redes surgiu um novo problema, que ao final de

seu treinamento percebia-se que o gradiente de entrada dessa rede havia desaparecido. Várias arquiteturas profundas apresentaram soluções para esse problema, como a ResNet [He et al, 2011] que criava conexões entre suas camadas para passar esse gradiente ou a Stochastic depth [Huang et al, 2016] que eliminava camadas aleatoriamente para proporcionar um melhor fluxo desse gradiente.

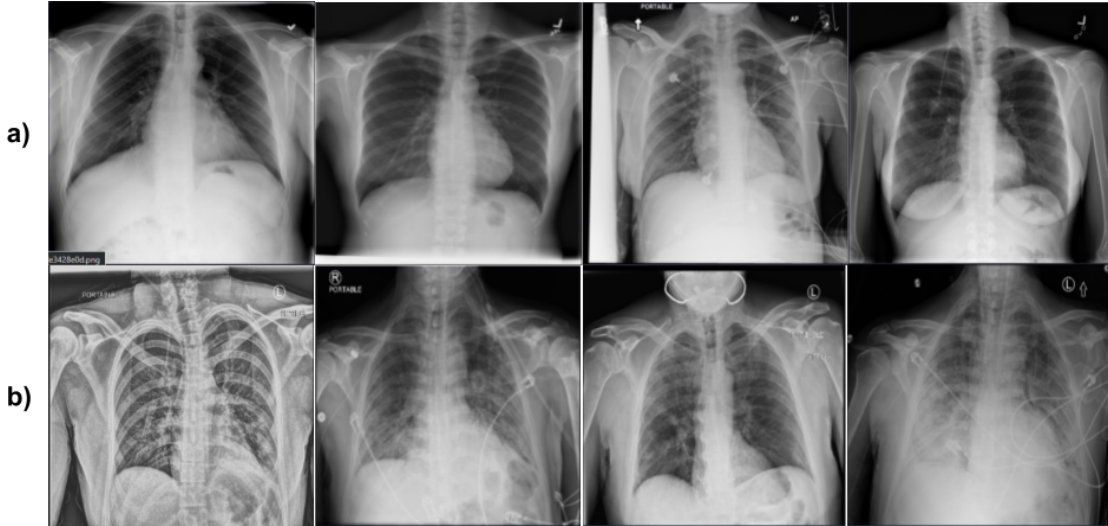


Figura 2. Exemplos de imagens da base de dados utilizada. a) representa casos negativos de COVID-19 e b) representa casos positivos.

A DenseNet por sua vez propôs uma solução diferente, para que ocorresse o maior fluxo possível de informação entre as camadas sua arquitetura conectava todas as camadas umas com as outras, assim cada camada recebe informações de todas as camadas anteriores e por sua vez passa suas informações para todas as camadas posteriores. Assim, uma rede tradicional do tipo *feed-forward*, como a ResNet, que possui um número L de camadas, conecta a saída de sua camada de número l a entrada de sua camada $l + 1$, gerando assim uma transição entre camadas definida pela equação:

$$x_l = H_l(x_{l-1}). \quad (1)$$

Já na ResNet a conexão é feita de uma camada para todas as camadas posteriores, assim cada camada receberia informações da diretamente anterior a ela assim como de todas as anteriores, então a equação de transição se modifica, ficando:

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (2)$$

Então, na Equação 2, $[x_0, x_1, \dots, x_{l-1}]$ se refere à concatenação dos mapas de características produzidos pelas camadas $l_0, \dots, l-1$. Ainda na Equação 2, H_l representa a função composta da DenseNet que é formada por 3 operações, uma *batch normalization*, uma *rectified linear unit (ReLU)* e uma *convolução de 3×3* .

Para a otimização da rede foi utilizado o algoritmo *Tree-structured Parzen Estimator (TPE)* [Bergstra et al, 2011], que é um algoritmo que otimiza o critério de

expected improvement (EI) [Jones, 2001]. Expected improvement é a expectativa de que em um modelo M de $f : X \rightarrow \mathbb{R}^N$ a função $f(x)$ irá exceder (negativamente) um limite y^* :

$$EI_{y^*}(x) = \int_{-\infty}^{\infty} \max(y^* - y, 0) pM(y|x) dy \quad (3)$$

O algoritmo TPE é um dos métodos do algoritmo SMBO [Hutter et al, 2011], que se diferenciam pela forma que é construído o modelo $P(x|y)$, onde y é o valor da função utilizando o hiperparâmetro x . No TPE a função $P(x|y)$ ao invés de ser representada diretamente é representada pela equação:

$$P(x|y) = \frac{P(x|y)^* P(y)}{P(y)} \quad (4)$$

Assim, $P(x|y)$ é expressada por:

$$P(x|y) = l(x) \text{ se } y < y^* \text{ ou } g(x) \text{ se } y \geq y^* \quad (5)$$

Por fim, através das Equações 4 e 5 temos que a equação da EI será representada por:

$$EI_{y^*}(x) = \frac{\gamma y^* l(x) - l(x) \int_{-\infty}^{y^*} p(y) dy}{\gamma l(x) + (1-\gamma) g(x)} \alpha \left(\gamma + \frac{g(x)}{l(x)} (1 - \gamma) \right)^{-1} \quad (6)$$

A Equação 6 nos mostra que para ocorrer a maximização do EI precisamos de pontos x com alta probabilidade em $l(x)$ e baixa probabilidade em $g(x)$.

2.3. Avaliação da Rede

Para nosso experimento foi utilizada a framework Optuna [Akiba et al, 2019] que implementa o algoritmo de otimização TPE. Na etapa de otimização foram utilizadas 3 espaços de busca de hiperparâmetros, o primeiro consiste em 3 elementos e definem o número de camadas para a construção da rede, sendo elas: [6,12,24,16], [6,12,32,32] e [6,12,48,32]. Cada um dos quatro números representa a quantidade de blocos de convolução 1×1 seguido por um 3×3 para os quatro blocos densos presentes na arquitetura. O segundo espaço de busca consiste em quatro valores e definem a taxa de aprendizado da rede, sendo eles: 0.1, 0.001, 0.0001 e 0.00001. Já o terceiro espaço de busca representa a taxa de crescimento da arquitetura DenseNet e se divide em dois casos diferentes, caso o número de camadas da rede fosse de [6,12,48,32] o espaço de busca consiste nos números inteiros de 4 a 28 e caso fosse qualquer outro número de camadas o espaço consiste nos números inteiros de 4 a 48. Foi então escolhida como métrica a ser maximizada durante a otimização, a acurácia do modelo.

Após serem escolhidos os hiperparâmetros, a rede é construída e então ela passa pelo processo de treino, validação e teste utilizando os *datasets* criados anteriormente. Cada um dos modelos foi treinado por um total de 20 épocas utilizando e um callback simples de paciência 4, para evitar que persistisse em treinamentos sem progresso. No treinamento foi utilizado um tamanho de batch de 32 e o otimizador utilizado foi o Adam [Kingma e Ba, 2014] e a função de perda Cross Entropy [De Boer et al, 2005],

com as implementações feitas utilizando a biblioteca Keras. Por fim, para a etapa de avaliação dos modelos criados foram utilizadas as métricas clássicas: acurácia, recall, f1-score e precisão.

$$\text{Acurácia} = \frac{\text{Verdadeiro Positivo} + \text{Verdadeiro Negativo}}{\text{Total}}$$

$$\text{Recall} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Negativo}}$$

$$\text{Precisão} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo} + \text{Falso Positivo}}$$

$$\text{F1 Score} = \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}}$$

3. Resultados

Essa seção tem como foco a apresentação dos resultados obtidos com a aplicação do nosso método e a comparação desses resultados com outros trabalhos.

Na Figura 3 é apresentado o gráfico da evolução da acurácia dos modelos durante a otimização. O otimizador foi executado por um total de 90 trials, gerando assim 90 modelos diferentes com seus respectivos hiperparâmetros e ao final de cada trial o modelo era salvo caso resultasse em um valor de acurácia superior a 90%.

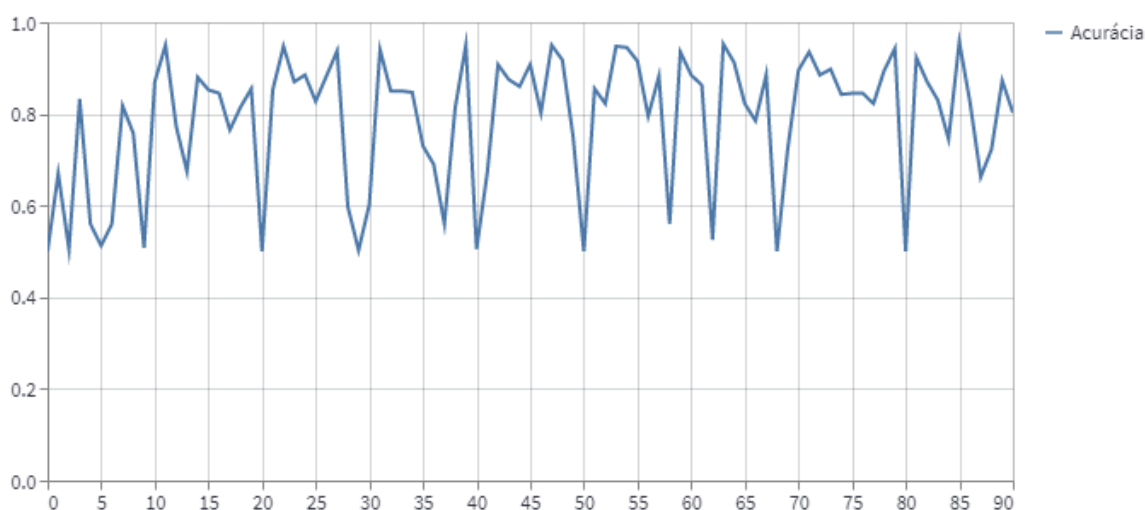


Figura 3. Evolução da acurácia do modelo durante os trials da otimização.

Verifica-se que vários modelos atingiram a meta mínima de 90%. Entretanto apenas o melhor foi selecionado. Ao fim das execuções o modelo mais promissor atingiu uma acurácia de 96% com uma combinação de camadas de [6,12,24,16], uma taxa de aprendizado de 1e-05 e uma taxa de crescimento de 45. Assim, na Figura 4 pode ser vista a representação da rede construída. Na Figura 5 é possível observar a matriz de confusão desse modelo encontrado, e pode-se ver que sua maior quantidade de erros é na classe negativa, com 12 erros, enquanto houveram apenas 5 erros de casos positivos de COVID-19. Já na Figura 6 é possível ver a curva ROC produzida pelo modelo, onde é apresentada uma Area Under Curve (AUC) de 0,96.

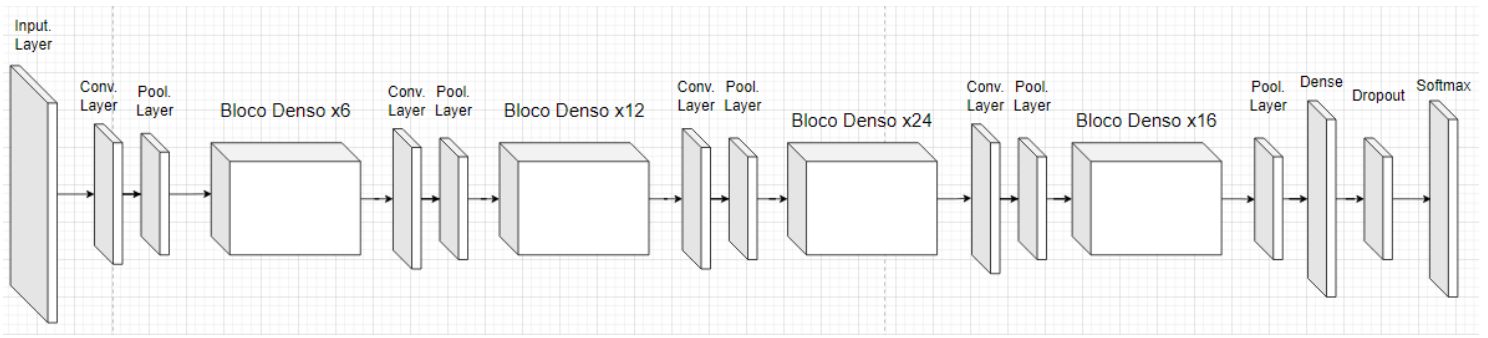


Figura 4. Representação do modelo final construído.

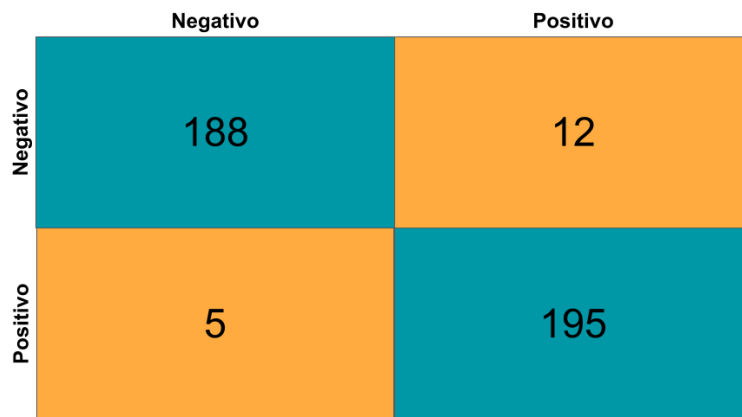


Figura 5. Matriz de confusão do melhor modelo encontrado através da otimização.

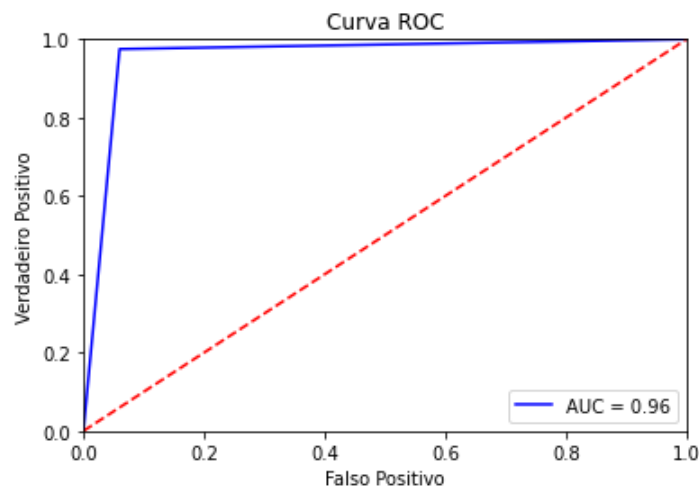


Figura 6. Curva ROC do melhor modelo encontrado.

Na Tabela 1 são apresentadas as métricas detalhadas para a melhor rede construída pelo processo de otimização. Na tabela estão presentes as métricas de precisão, recall e f1-score para cada uma das duas classes diagnosticadas pelo modelo, além da média das duas. Como a base de testes está balanceada, com 200 imagens em cada classe, foram apresentados apenas os valores da média aritmética, já que eram os mesmos da média ponderada. Nela é possível ver a precisão para os casos normais se

mostrou um pouco superior ao dos casos de COVID-19, com uma diferença de 0,3 pontos, assim conseguindo atingir uma média de precisão de 96%.

Tabela 1. Métricas para o melhor modelo (Acurácia: 0,96).

Classe	Precisão	Recall	F1-Score
Normal	0,97	0,94	0,96
Covid	0,94	0,97	0,96
Média Macro	0,96	0,96	0,96

Também foi feita a comparação das métricas encontradas através da nossa otimização com os resultados de outros trabalhos relacionados. Esses valores estão sendo apresentados na Tabela 2, e utilizam a métrica de acurácia para a comparação entre os trabalhos. Vale ressaltar que nosso trabalho é o COVID-Net CXR-2 classificam as imagens entre casos positivos e negativos de COVID-19, enquanto os outros classificam entre 3 classes, sendo elas COVID-19, pneumonia não causada por COVID-19 e casos normais.

Tabela 2. Comparação dos melhores modelos encontrados com outros trabalhos.

Modelo	Acurácia	Recall	Precisão	F1-Score
DenseNet-SVM [Aslan et al, 2021]	96,29	0,96	0,96	0,96
mAlexNet + BiLSTM [Aslan et al, 2021]	98,70	0,98	0,98	0,98
VGG19-RNN [Al-Rakhami et al, 2021]	99,86	0,99	0,99	0,99
COVID-Net [Wang et al, 2020]	93,30	-	-	-
COVID-Net CXR-2 [Pavlova et al, 2021]	96,30	0,95	0,97	0,96
COVID Dense-Net	96,00	0,96	0,96	0,96

Analisando nossos resultados em comparação aos outros trabalhos, é possível perceber que nossos valores são bastante competitivos com os outros modelos apresentados, ainda que ligeiramente abaixo. Comparando diretamente com o modelo apresentado por Pavlova et al (2021), que utiliza o mesmo dataset e classifica no mesmo número de classes, pode-se ver que o nosso modelo se mostra equiparável, com uma diferença de apenas 0.3%. Isso mostra que a construção de um modelo baseado na arquitetura DenseNet tem potencial para apresentar resultados competitivos com outros modelos estado da arte no diagnóstico de COVID-19.

4. Conclusão

Neste trabalho foi apresentada a construção de um modelo CNN especializado no diagnóstico de COVID-19 baseado na arquitetura DenseNet. A rede foi construída com a utilização de um método de otimização utilizando o algoritmo *Tree-structured Parzen Estimator* e avaliada após seu treinamento e teste utilizando radiografias do tórax.

Os resultados obtidos após a otimização se mostraram promissores e competitivos com algumas das redes estado da arte no diagnóstico de COVID-19, com poucos pontos percentuais de diferença. A rede final chegou a uma acurácia máxima de 96% e um recall de 97% para os casos de COVID-19, mostrando que é possível construir uma rede com bons resultados utilizando a otimização com o algoritmo TPE.

Quanto às limitações do trabalho, a maior delas foi o tempo disponível para a realização da pesquisa, se mostrando um limitador da quantidade de experimentos que poderiam ser realizados. Assim, como trabalhos futuros, pretendemos fazer uma otimização mais robusta da rede, otimizando um maior número de parâmetros e testando essa otimização em outros modelos além da DenseNet, além da utilização de um ensemble de classificadores. Também é desejável testar outras técnicas de pré-processamento e fazer um treinamento mais extensivo com uma quantidade maior de épocas para avaliar os resultados. Por fim, temos interesse em aplicar técnicas de auto-supervisão e verificar sua eficácia na detecção de doenças respiratórias através de radiografia do tórax.

Agradecimentos

Os autores agradecem às instituições brasileiras que ajudaram no desenvolvimento desta pesquisa: Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e Fundação de Amparo à Pesquisa e ao Desenvolvimento Científico e Tecnológico do Maranhão (FAPEMA) pelo financiamento Apoio.

Referências

- Alsharif, W., & Qurashi, A. (2021). Effectiveness of COVID-19 diagnosis and management tools: A review. *Radiography*, 27(2), 682-687.
- O'Shea, Keiron, and Ryan Nash. "An introduction to convolutional neural networks." arXiv preprint arXiv:1511.08458 (2015).
- Pavlova, M., Terhijan, N., Chung, A. G., Zhao, A., Surana, S., Aboutaleb, H., ... & Wong, A. (2021). COVID-Net CXR-2: An enhanced deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. arXiv preprint arXiv:2105.06640.
- Aslan, M. F., Unlarsen, M. F., Sabanci, K., & Durdu, A. (2021). CNN-based transfer learning–BiLSTM network: A novel approach for COVID-19 infection detection. *Applied Soft Computing*, 98, 106912.

- Al-Rakhami, M. S., Islam, M. M., Islam, M. Z., Asraf, A., Sodhro, A. H., & Ding, W. (2021). Diagnosis of COVID-19 from X-rays using combined CNN-RNN architecture with transfer learning. *MedRxiv*, 2020-08.
- Aslan, M. F., Sabanci, K., Durdu, A., & Unlarsen, M. F. (2022). COVID-19 diagnosis using state-of-the-art CNN architecture features and Bayesian Optimization. *Computers in Biology and Medicine*, 105244.
- Costa, G., Paiva, A., Braz Júnior, G., & Ferreira, M. (2021). COVID-19 automatic diagnosis with CT images using the novel Transformer architecture. In *Anais do XXI Simpósio Brasileiro de Computação Aplicada à Saúde*, (pp. 293-301). Porto Alegre: SBC. doi:10.5753/sbcas.2021.16073
- Wang, L., Lin, Z. Q., & Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images. *Scientific Reports*, 10(1), 1-12.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4700-4708).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016, October). Deep networks with stochastic depth. In *European conference on computer vision* (pp. 646-661). Springer, Cham.
- Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in neural information processing systems*, 24.
- Jones, D. R. (2001). A taxonomy of global optimization methods based on response surfaces. *Journal of global optimization*, 21(4), 345-383.
- Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011, January). Sequential model-based optimization for general algorithm configuration. In *International conference on learning and intelligent optimization* (pp. 507-523). Springer, Berlin, Heidelberg.
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019, July). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2623-2631).
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- De Boer, P. T., Kroese, D. P., Mannor, S., & Rubinstein, R. Y. (2005). A tutorial on the cross-entropy method. *Annals of operations research*, 134(1), 19-67.