Optimization of the Fused-DenseNet-Tiny Model Applied to the Detection of COVID-19 and Pneumonia in Chest X-Ray Images

Eder Silva dos Santos Júnior[1], **Salomão Machado Mafalda**[1], **Roger Fredy Larico Chavez**[1], **Ana Beatriz Alvarez**[1]

[1]Center of Exact and Technological Sciences
Federal University of Acre – Rio Branco, AC – Brazil

{eder.junior,salomao.mafalda}@sou.ufac.br, {roger.chavez,ana.alvarez}@ufac.br

Abstract. *The COVID-19 pandemic, faced in recent years, has highlighted the need for new forms of testing that help in the early diagnosis of the disease. An alternative gaining more and more prominence is the use of Artificial Neural Networks, algorithms capable of being trained to identify this and other diseases. However, these architectures have a high demand for hardware resources, making their implementation difficult. In this work, we propose the optimization of the Fused-DenseNet-Tiny Convolutional Neural Network for the detection of COVID-19 and Pneumonia in x-ray images, using the pruning technique. After pruning, a model with a size about 3 times smaller and an accuracy of 97.17% was obtained, without significant loss.*

1. Introduction

The pandemic of the new coronavirus (COVID-19), a disease caused by the SARS-CoV-2 virus, is considered the most serious public health problem of this generation, because it is a disease with a high rate of dissemination and capable of presenting several symptoms that can be fatal for those infected, such as severe respiratory failure. [Braga et al. 2020]. Currently, vaccination on a global scale is advancing every day, but testing remains essential to combat the disease, because early diagnosis and treatment at an early stage of someone who has contracted the disease, decreases the risk of worsening symptoms [Almeida 2021]. In addition, a good amount of testing allows us to locate more precisely the regions that are most impacted by the virus, so that the necessary measures can be taken by the institutions responsible for public health [Almeida 2021]. The same thought applies to the prevention of pneumonia, which affects about 450 million people worldwide each year [Oliveira et al. 2016].

According to [Beduin 2021], in this context, many new forms of testing have emerged that use Artificial Intelligence (AI), and more specifically Deep Learning (DL) algorithms, to analyze and extract features and information from databases. Even if not sufficiently accurate for direct diagnosis of COVID-19, these tests can help to streamline health care systems as a support tool in countries where more accurate clinical tests are difficult to obtain, or as primary screening. The Figure 1 shows how difficult it is to visually identify one of the diseases, without a medical diagnosis or a test of this kind.

Thus, one can work on automatically generating pneumonia and COVID-19 diagnoses from medical chest X-ray images using these Deep Learning methods. This

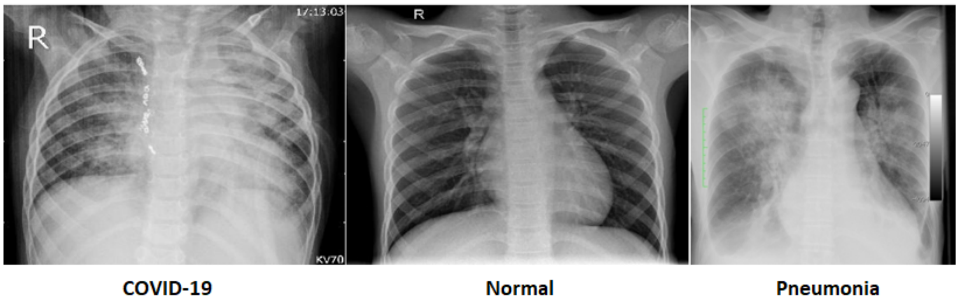


Figure 1. Samples from the image bank of each of the classes.

area covers many techniques useful for performing image classification tasks, such as this research, which aims to classify input data into pneumonia, COVID-19 or normal [Hosaki and Ribeiro 2021]. The approach that is adopted for this, enabling the radiographs to be processed and recognized, is Convolutional Neural Networks (CNN), which are intelligent computer systems inspired by the functioning of biological neural networks [Haykin 2001]. State-of-the-art CNN architectures guarantee very expressive accuracies in the detection of Pneumonia and COVID-19 in chest X-ray images. Considering the study done in [Santos Júnior 2022], for example, an accuracy of 99.25% was obtained using ResNet50 for the same dataset.

However, despite being powerful and efficient tools, neural networks have a disadvantage, which is the high demand for hardware resources required for their implementation [Montes et al. 2021]. According to [Silva et al. 2022], researchers have been searching for model compression techniques to reduce the intense computational demand, either of processing or memory, without significantly worsening the results. For this, an alternative is the adoption of Artificial Neural Networks (ANN) pruning, proposed by [LeCun et al. 1989], which makes it possible to optimize the models without loss (or minimal loss) of accuracy.

In this work, we propose the optimization of the Fused-DenseNet-Tiny Convolutional Neural Network for the detection of COVID-19 and Pneumonia in x-ray images, using the pruning technique. Model compression was used to reduce computational cost while maintaining good performance. Thus, an architecture that works well on devices with low hardware resources can be obtained.

2. Materials and Methods

2.1. Database

Second [Beduin 2021], data acquisition and organization is one of the most important parts in Machine Learning. Considering the data types, important aspects of the model's architecture are defined. In this case, the dataset selected for training and validation of the optimized architecture was the *Curated X-Ray Dataset*, which consists of a compilation of x-ray image sets developed by [Sait 2020]. This is a public dataset that is available on the *Kaggle*[1] platform for downloading and computational testing.

The database encompasses a total of 9208 radiographs, separated into three classes: Normal, Covid-19 and Pneumonia. The images are divided into two subsets:

[1] https://www.kaggle.com/datasets/francismon/curated-covid19-chest-xray-dataset

training and validation, being distributed in 80% separate for training and 20% for validation. Table 1 show a detailed description of the dataset division, explaining the number of images in each class.

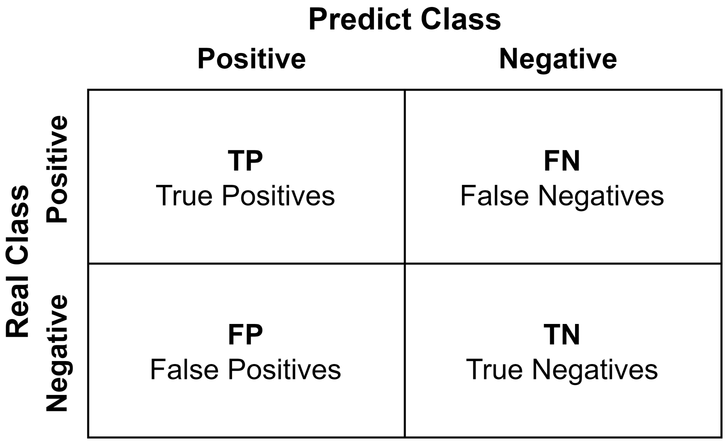
Table 1. Distribution of data by class.

Class	**Train (80%)**	**Validation (20%)**	**Total (100%)**
Normal	2616	654	3270
COVID-19	1025	256	1281
Pneumonia	3726	931	4657
Total	**7367**	**1841**	**9208**

The data indicate that there are a total of 3270 x-ray samples from the Normal class, 4657 from patients diagnosed with Pneumonia, and 1281 from patients diagnosed with COVID-19. The acquired images are in JPG (Joint Photographic Group) format, with varying dimensions, and were resized according to the convolutional neural network architecture used, to have the appropriate input dimensions.

2.2. Validation Metrics

According to [Monard and Baranauskas 2003] the confusion matrix plays an essential role in visualizing the assertiveness of models in classification tasks, since it correlates the actual classes with those that were predicted, as illustrated in Figure 2. TP being the number of True Positives, TN the number of True Negatives, FP the number of False Positives, and FN the number of False Negatives.



		Predict Class	
		Positive	**Negative**
Real Class	**Positive**	**TP** True Positives	**FN** False Negatives
	Negative	**FP** False Positives	**TN** True Negatives

Figure 2. Illustration of a confusion matrix.

The performance metrics for the validation step can be extracted from the confusion matrix and, in this study, the ones selected are the accuracy ou ACC, accuracy, Recall and F1-Score, mathematically expressed by the Equations 1, 2, 3 e 4. The metrics, defined according to [Iqbal and Aftab 2020], are explained in the following.

- Accuracy: In classification problems, is the number of correct predictions made by the model over all the types of predictions made. This metric provides a measure of overall model performance, in terms of number of hits, and can be calculated using Equation 1.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Precision: Precision is the ratio of true positives (TP) to the total number of samples that classified as positive. To calculate the accuracy with respect to a certain class (for example, positive class), the Equation 2 is used.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

- Recall: Is the ratio of correct predictions to the total number of samples in that class being evaluated. This metric indicates how many examples of a class, relative to the total present in the set, were identified by the model. The sensitivity is given by Equation 3.

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

- F1-Score: It is the harmonic mean between accuracy and sensitivity. Since its value is high, it means that the obtained accuracy is relevant, that is, the values of TP, TN, FP and FN do not present great distortions. Then, this metric can also be interpreted as a measure of the reliability of the accuracy.

$$F1 - Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{4}$$

2.3. Fused-DenseNet-Tiny model

The paper focuses on the optimization of a specific model, proposed in [Montalbo 2021] and denominated Fused-DenseNet-Tiny[2], what is a truncated DenseNet architecture with partial layer freeze and feature fusion. This architecture stands out by having a low amount of parameters (1,231,235), when compared to other CNNs, in addition to performing well in detecting Pneumonia and COVID-19. Training and validating the model in the image bank Curated X-Ray Dataset, the author obtained an accuracy of 97.99%, precision of 98.38%, recall of 98.15% and F1 Score of 98.26%, even being competitive with some State of the Art models, such as DenseNet and EfficientNeB0.

Therefore, it can be stated that Fused-DenseNet-Tiny is a very promising architecture with low computational cost, however, to perform model inference and run on mobile devices, especially those with low hardware resources, it is possible to further reduce insignificant parameters. In this research, it is desired to optimize the architecture by means of a type of model compression technique called pruning, with the method presented in the next subsection.

2.4. Fused-DenseNet-Tiny model optimization

The model optimization followed a line of reasoning similar to that adopted in [Montes et al. 2021], whose process for optimizing the architecture consisted of three main steps: (I) removal of insignificant weights, (II) removal of the casings used during

[2] https://github.com/francismontalbo/fused-densenet-tiny

pruning and, finally, (III) applying a compression algorithm, as shown in Figure 3. This step by step was implemented in Python using *Tensorflow Model Optimization*, which is a Tensorflow toolkit for optimizing Machine Learning models. In the process, basically, the Tensorflow Model Optimization pruning algorithm is used, which makes some weights close to zero null, based on a certain threshold, the effect of which can be seen in Figure 4. According to [Goldbarg 2021], in the iterative algorithm of conventional magnitude pruning, this threshold is given mathematically by Equation 5. Where β_k is the threshold of k-th layer, α is the constant that establishes the aggressiveness of pruning and σ_k is the standard deviation of k-th layer.

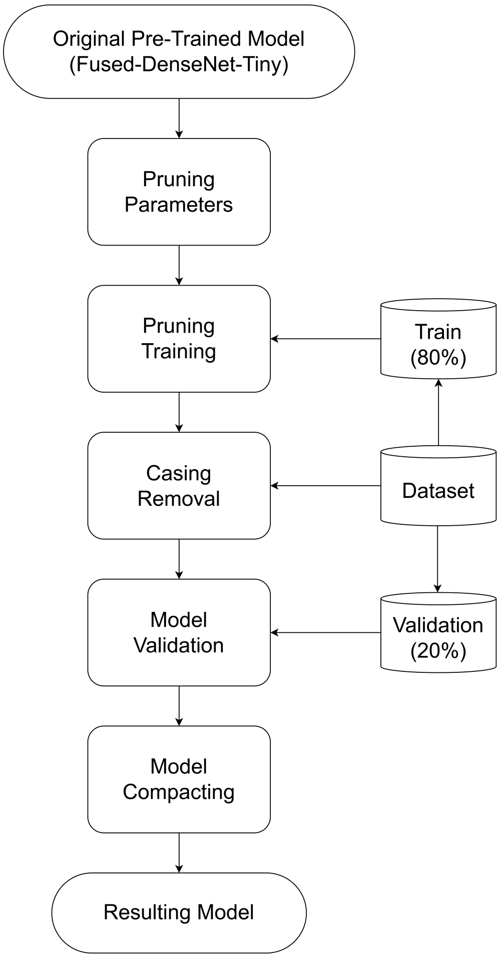


Figure 3. Stages of the implemented optimization process.

$$\beta_k = \alpha \cdot \sigma_k \tag{5}$$

In its execution, this algorithm creates a conditional structure that selects the weight if it is greater than or equal to β_k, or zeroes the weight if it is less than β_k. Then, the resulting weights of the architecture, denoted by $C(n)$, can be described by Equation 6, after applying the pruning function $P(.,.)$ to n-th weight $W(n)$ [Goldbarg 2021].

$$C_k(n) = P(W_k(n), \beta_k) = \begin{cases} w_k(n), & \text{se } |w_k(n)| \geq \beta_k \\ 0, & \text{se } |w_k(n)| < \beta_k \end{cases} \tag{6}$$

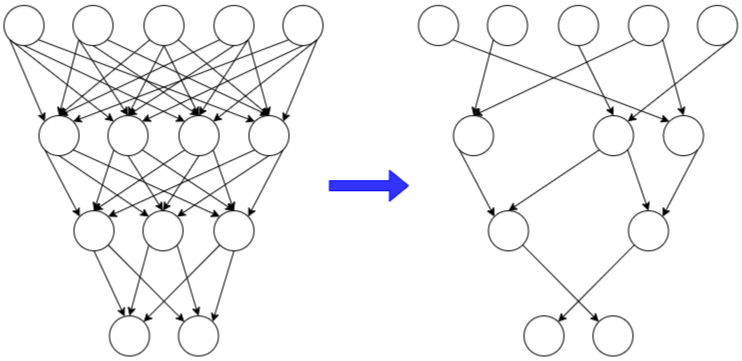


Figure 4. Illustration of the application of pruning to a neural network.

To go through this process, the chosen architecture, already trained with the hyperparameters from Table 2, should be submitted to a new training, with a reduced number of epochs (epochs = 5), where the Fused-DenseNet-Tiny will be pruned with an initial sparsity of 30%. This means that 30% of the parameters are set to zero and increase throughout training until the final sparsity of 80%. The *strip_pruning* function was also used for part (II), referring to the removal of the wrappers, and the *zipfile* package compression function for step (III). It was possible to check, at the end of the experiment, the size of the model with compacted pruning and compare it with the size of the original model, along with the selected performance metrics, allowing us to observe what effects this technique has for CNN. The parameters for pruning the model, which were adopted, are contained in Table 3.

Table 2. Training hyperparameters of the original model.

Hyperparameter	Value
Learning Rate	0.0001
Batch Size	16
Optimizer	Adam
Dropout	0.5
Epochs	25

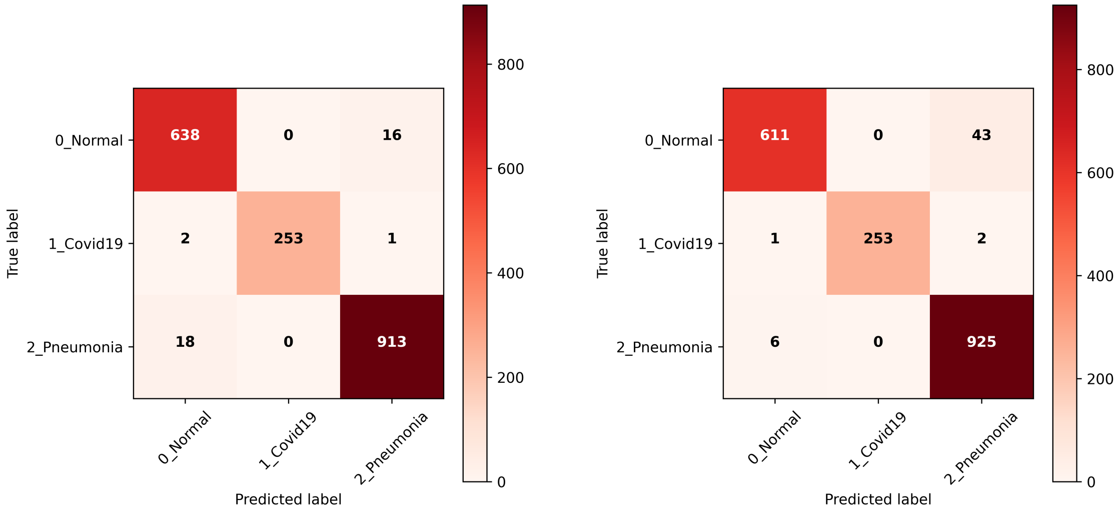
Table 3. Pruning parameters applied to the model.

Parameter	Value
Initial Sparsity	30%
Final Sparsity	80%
Optimizer	Adam
Epochs	5

3. Results and Discussions

In Figure 5, the confusion matrices of the validation of the original Fused-DenseNet-Tiny model before pruning (Figure 5a), and of the optimized Fused-DenseNet-Tiny model, after pruning (Figura 5b). Looking at the main diagonal of these matrices, one can see that the pruning algorithm maintained the amount of correct classifications of COVID-19 cases, while increasing the number of correct diagnoses of Pneumonia patients, which is a very significant result.

There is an attenuation of hits for the Normal class, however, it is important to note that the focus is on identifying people with one of the diseases. In particular, having few false positives for the Normal class is important, so that a low number of patients are misleadingly classified as normal, since not diagnosing the disease can make it worse. The pruned model showed only 7 false positives for the Normal class. These 7 errors can also be seen as the false negatives of the COVID-19 class and the Pneumonia class, so that the sensitivity of these classes related to these quantities.



(a) Original Fused-DenseNet-Tiny

(b) Fused-DenseNet-Tiny pruned

Figure 5. Confusion matrix of the model validation.

Briefly, the Table 4 presents the performance of the optimized architecture, for each class, so that an individual analysis of the metrics of different classes, before and after pruning, can be performed. The global result is exposed in Table 5, which contemplates the averages of each metric, where a column was added for comparison between model sizes, to emphasize the effect of optimization.

In fact, an efficient model was maintained for diagnosing Pneumonia and COVID-19, identifying 253 cases of patients with COVID-19 out of 256 (98.83% of cases), and 926 cases of patients with Pneumonia out of 932 (99% of cases).

However, this research was focused on reducing the model, in terms of computational cost, and it was possible to reduce the size (in Megabytes - MB) of the Neural Network from 4.635MB to 1.540MB, as shown in bold in Table 5, representing about 3 times less memory footprint. In compressed form, the file is even smaller, at 1.458MB. The pruned architecture obtained an accuracy of 97.12%, a precision of 98.04%, a sensitivity of 97.16% and an F1-Score of 97.56%.

The Table 6 compares the model with other similar studies, i.e., also aimed at identifying COVID-19 and/or Pneumonia on chest X-ray images, or CXR (Curated X-

Table 4. Metrics obtained for each class in the validation stage.

Class: Normal				
Model	**ACC (%)**	**Precision**	**Recall**	**F1-Score**
Original Fused-DenseNet-Tiny	97.56	0.97	0.97	0.97
Fused-DenseNet-Tiny Pruned	93.28	0.99	0.93	0.96
Class: COVID-19				
Model	**ACC (%)**	**Precision**	**Recall**	**F1-Score**
Original Fused-DenseNet-Tiny	98.83	1.00	0.99	0.99
Fused-DenseNet-Tiny Pruned	98.83	1.00	0.99	0.99
Class: Pneumonia				
Model	**ACC (%)**	**Precision**	**Recall**	**F1-Score**
Original Fused-DenseNet-Tiny	98.07	0.98	0.98	0.98
Fused-DenseNet-Tiny Pruned	99.36	0.95	0.99	0.97

Table 5. Comparison of overall model performance.

Model	**Overall Performance**				
	ACC (%)	Precision	Recall	F1-Score	Size (MB)
Original Fused-DenseNet-Tiny	97.99	0.98	0.98	0.98	4.635
Fused-DenseNet-Tiny Pruned	97.17	0.98	0.97	0.98	**1.540**
Fused-DenseNet-Tiny pruned and compacted	97.17	0.98	0.97	0.98	**1.458**

Ray), in a similar way to comparison made by [Montalbo 2021]. CNN stands out among these architectures, with an accuracy second only to original Fused-DenseNet-Tiny, but with a difference of only 0.82%, and with a computational cost about 3 times lower, as mentioned.

The Figure 6 shows the sizes of other State of the Art models (in MB), to give you a sense of how small the architecture is compared to other high performance deep neural networks in the bibliography, where DenseNet121 [Huang et al. 2017] with 33MB, EfficientNetB0 [Tan and Le 2019] with 29MB, the InceptionV3 [Szegedy et al. 2016] with 92MB, the ResNet152V2 [He et al. 2016] with 232MB, the XCeption [Chollet 2017] with 88MB, the MobileNetV2 [Sandler et al. 2018] with 14MB, the VGG16 [Simonyan and Zisserman 2014] with 528 and finally the InceptionResNetV2 [Szegedy et al. 2017] with 215MB, in the Keras format, with the pre-trained weights.

4. Conclusions

The present work dealt with the application of an optimization technique to an CNN model, called Fused-DenseNet-Tiny, which was proposed by [Montalbo 2021] and trained on the chest x-ray image bank *Curated X-Ray Dataset* [Sait 2020], for the task of classifying patients into Normal, COVID-19 and Pneumonia classes. The pre-trained algorithm was submitted to a model compression method, known as magnitude pruning,

Table 6. Comparison of the architecture performance with other studies.

Model	ACC (%)	Class
This work	97.17	Normal, COVID-19, Pneumonia
Fused-DenseNet-Tiny [Montalbo 2021]	97.99	Normal, COVID-19, Pneumonia
COVID-Net [Wang et al. 2020]	93.30	Normal, COVID-19, Pneumonia
Modified ResNet-18 [Al-Falluji et al. 2021]	96.37	Normal, COVID-19, Pneumonia
ECOVNet-EfficientNetB3 [Garg et al. 2022]	97.00	Normal, COVID-19, Pneumonia
Modified Xception [Singh et al. 2020]	95.70	Normal, COVID-19, Pneumonia
DarkCovidNet [Ozturk et al. 2020]	87.02	Normal, COVID-19, Pneumonia
DeTraC-ResNet18 [Abbas et al. 2021]	95.12	Normal, COVID-19, SARS
Hierarchical EfficientNetB3 [Luz et al. 2021]	93.51	Normal, COVID-19, Pneumonia

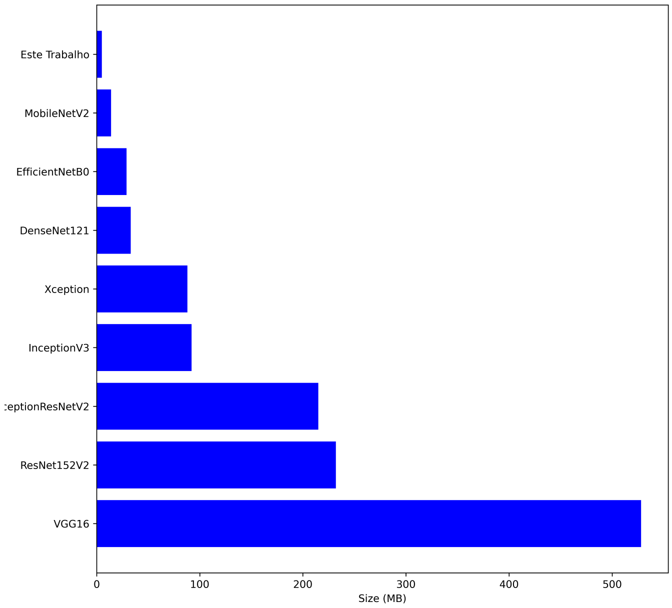


Figure 6. Comparison between the size of the architecture and other State of the Art models.

aiming to reduce the weights in order to obtain a simpler and more compact architecture, meeting, for example, memory and latency constraints, without a significant loss of accuracy.

The pruned architecture achieved good performance metrics, in validation, reaching an accuracy of 97.17%, below the original model result (97.99% in this case) by only 0.82%. Furthermore, considering that an algorithm is desired that can become an aid in the diagnosis of Pneumonia and COVID-19, the individual metrics of these two classes represent good results, especially regarding sensitivity, which is a very important measure in medical applications. It is known that the worst type of error you can have is the false negative, which must be close to 0, meaning that the sensitivity, inversely proportional, must be high. With the pruned model, a sensitivity of approximately 99% for the COVID-19 class and 99% for the Pneumonia class was obtained, indicating a true positive rate close to 100% and that the number of false negatives is low, as shown in Table 4. Only in 7 predictions sick people were confused by the algorithm with those of the Normal class, 1 of which had COVID-19 and the other 6 had Pneumonia, as shown in the confusion matrix of Figure 5b.

In addition, the architecture performed highly relative to other similar studies, which also had the purpose of classifying COVID-19 and/or Pneumonia on chest x-ray images, according to Table 6, besides being a significantly reduced model, when compared to other architectures that are in the State of the Art. The obtained model presented low computational cost, maintaining good performance, in such a way that in a physical file, the size of the architecture can reach about 1.5MB, 3 times smaller than the original size. The smaller this memory footprint, the better it is for automatic diagnostic generation tools, since there is a growing demand for smart devices and embedded systems that operate mostly in real time and whose processing power is limited, motivating the creation of neural networks adapted to devices with limited memory, processor or battery power.

In future work, it is suggested to use other model compression approaches by pruning, to investigate the effects that other types of optimization algorithms can bring to the architecture, considering the same model and dataset. In this research, was used magnitude pruning from the *Tensorflow Model Optimization* package in all layers of the neural network, but there are other alternatives to be tested, for example, implementing pruning in specific layers of the network. Since there are layers that impact the outcome more than others, exclusively selecting these layers can provide better model performance. There is also interest in examining other model compression techniques, such as quantization, which can reduce the amount of memory required to store and load the model by mapping the weights, normally encoded as 32-bit floating-point, to representations with reduced precision, such as 16 bits, for example.

Acknowledgments

This study is made possible thanks to the support and resources provided by the Motorola project: Pesquisa Aplicada em Visão e Inteligência Computacional (PAVIC). This work had the privilege of being supported by PAVIC, which made it possible and has promoted the expansion of knowledge in research area.

References

Abbas, A., Abdelsamea, M. M., and Gaber, M. M. (2021). Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network. *Applied Intelligence*, 51:854–864.

Al-Falluji, R. A., Katheeth, Z. D., and Alathari, B. (2021). Automatic detection of covid-19 using chest x-ray images and modified resnet18-based convolution neural networks. *Computers, Materials, & Continua*, pages 1301–1313.

Almeida, B. (2021). Covid-19: Mesmo com a vacina, ainda é importante a testagem. Disponível em: `https://medicinasa.com.br/importancia-testagem/`. Acesso em: 19 de fevereiro de 2023.

Beduin, I. R. O. (2021). Detecção da covid-19 em imagens de raio-x: construindo um novo modelo de aprendizado profundo utilizando automl.

Braga, I. O., Cunha, C. C., Palácio, M. A. V., Brito, S. B. P., and Takenami, I. (2020). Pandemia da covid-19: o maior desafio do século xxi. *Vigilância Sanitária em Debate: Sociedade, Ciência & Tecnologia*, 8(2):54–63.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258.

Garg, A., Salehi, S., La Rocca, M., Garner, R., and Duncan, D. (2022). Efficient and visualizable convolutional neural networks for covid-19 classification using chest ct. *Expert Systems with Applications*, 195:116540.

Goldbarg, M. A. S. d. S. (2021). Análise de técnicas de compressão em redes neurais profundas por poda em dataset de imagens. B.S. thesis, Universidade Federal do Rio Grande do Norte.

Haykin, S. (2001). *Redes neurais: princípios e prática*. Bookman Editora.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer.

Hosaki, G. Y. G. Y. and Ribeiro, D. F. (2021). Deep learning: ensinando a aprender.

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

Iqbal, A. and Aftab, S. (2020). A classification framework for software defect prediction using multi-filter feature selection technique and mlp. *International Journal of Modern Education & Computer Science*, 12(1).

LeCun, Y., Denker, J., and Solla, S. (1989). Optimal brain damage. *Advances in neural information processing systems*, 2.

Luz, E., Silva, P., Silva, R., Silva, L., Guimarães, J., Miozzo, G., Moreira, G., and Menotti, D. (2021). Towards an effective and efficient deep learning model for covid-19 patterns detection in x-ray images. *Research on Biomedical Engineering*, pages 1–14.

Monard, M. C. and Baranauskas, J. A. (2003). Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, 1(1):32.

Montalbo, F. J. P. (2021). Diagnosing covid-19 chest x-rays with a lightweight truncated densenet with partial layer freezing and feature fusion. *Biomedical Signal Processing and Control*, 68:102583.

Montes, G. C., Lattari, L. G., and Coelho, A. M. (2021). Redes neurais convolucionais otimizadas para a detecçao de supernovas. In *Anais do XV Brazilian e-Science Workshop*, pages 1–8. SBC.

Oliveira, F. M. d. J. et al. (2016). *Impacto da pneumonia grave no sistema nervoso centralefeitos de curto e longo prazo*. PhD thesis.

Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O., and Acharya, U. R. (2020). Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in biology and medicine*, 121:103792.

Sait, U. (2020). Curated chest x-ray image dataset for covid-19. *Kaggle Repository*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Santos Júnior, E. S. d. (2022). Classificação de pneumonia e covid-19 em imagens de raio-x do tórax através de técnicas de deep learning.

Silva, J. V. S., Matos, L. M., Santos, F., Cerqueira, H. O. M., Macedo, H., Prado, B. O. P., da Silva, G. J. F., and Bispo, K. A. (2022). Combinação de técnicas de compressão de modelos profundos. *IEEE Latin America Transactions*, 20(3):458–464.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Singh, K. K., Siddhartha, M., and Singh, A. (2020). Diagnosis of coronavirus disease (covid-19) from chest x-ray images using modified xceptionnet. *Romanian Journal of Information Science and Technology*, 23(657):91–115.

Szegedy, C., Ioffe, S., Vanhoucke, V., and Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR.

Wang, L., Lin, Z. Q., and 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.