Pneumonia detection with chest X-ray images using Machine Learning models

João Carvalho (106310)
DETI
University of Aveiro
Aveiro, Portugal
jpmffc@ua.pt

João Santos (76912)
DETI
University of Aveiro
Aveiro, Portugal
santos.martins.joao@ua.pt

***Abstract*—Machine learning models applied in the medical fields as been study for many year. Yet, with the recent COVID-19 pandemic requiring novel, faster and cheaper diagnosis methods, artificial intelligence gained new strengths in this field. In this work, a study will be carried out to access the efficiency and applicability of machine learning in the early diagnosis of pneumonia and COVID-19. Multiple methods will be implemented and compared against each other, namely by using the transfer learning technique. Although this technique seamed to indicate a potential benefit over simpler convolution neural network models, a clear advantage was not proved.**

***Index Terms*—Pneumonia, Machine Learning, Deep Learning, COVID-19, EfficientNet, transfer learning**

I. Introduction

Acute pulmonary infections like pneumonia can be brought on by bacteria, viruses, or fungi. When pneumonia infects the lungs, it inflames the air sacs and results in pleural effusion, which is when the lung becomes flooded with fluid. Pneumonia accounts for 14% of deaths in children under the age of five years [1].

The majority of cases of pneumonia occur in underdeveloped and developing nations, where there is a shortage of medical resources, excessive population, pollution, and unhygienic environmental conditions. Therefore, preventing the disease from becoming fatal can be greatly assisted by early diagnosis and management. For diagnosis, radiological imaging of the lungs using Computed Tomography (CT), Magnetic Resonance Imaging (MRI), or Radiography (X-ray) is frequently used. An affordable, non-invasive way to examine the lungs is via X-ray images. However, subjective variability might occur during chest scans [2], [3]. As a result, the detection of pneumonia must be automated.

Deep learning is a powerful tool for artificial intelligence that is essential to the resolution of many challenging computer vision issues. Many different picture categorization issues are solved using deep learning models, particularly Convolution Neural Networks (CNNs).

However, these models only operate at their best when they are given a lot of data. It is challenging to obtain such a large amount of labeled data for biomedical image classification problems because doing so necessitates paying expensive and time-consuming medical professionals to classify each image. A solution to get around this problem is transfer learning.

Transfer learning is the approach of using the pertinent components of a machine learning model that has already been trained to solve a different but related problem.

II. State of the Art

Machine Leaning (ML) models applied to X-ray images for disease detection is not a new field of study [4], [5]. Yet, regarding the pandemic of the past couple of years, this field gained some focus once again in the effort to detect the damages causes by the COVID-19 disease.

[6] applied data augmentation to a dataset that included 423 COVID-19, 1485 viral pneumonia and 1579 healthy X-ray images. With various CNN architectures, including MobileNetv2, SqueezeNet, ResNet18, ResNet101, DenseNet201, CheXNet, Inceptionv3, and VGG19 with ImageNet trained weights. The authors also used transfer learning techniques. When only considering the COVID-19 and healthy images, this methodology provided an accuracy of 0.997, a precision of 0.997 and an F1-score of 0.997. On the other hand, if the viral pneumonia images were also considered, the results drop down to an accuracy of 0.979, a precision of 0.975 and an F1-Score of 0.979.

[7] carried out the classification of COVID-19 X-ray images using the Auxiliary Classifier Generative Adversarial Network (ACGAN) technique. The authors resized the input images to a size of 112×112 and fine-tuned the fully connected layer weights. The author added data augmentation to a dataset that already contained 721 images of the healthy and 403 images of COVID-19 during the preprocessing stage. This approach reached an accuracy of 0.960. Although the results of this study are encouraging, the significant reduction of the size of the images may have result on the loss of key information.

[8] applied a technique that uses VGG16 CNN architecture to highlight specific chest radiography regions to identify pneumonia. 250 COVID-19, 3520 healthy and 2753 pneumonia images were classified in their study, achieving a F1-Score of 0.940 and an accuracy of 0.960. Despite the positive outcome, the authors choose not to investigate additional options, such as image preprocessing.

Three Deep Convolution Neural Network (DCNN) architectures were used in the suggested method by [9]. The authors

used a dataset with 50 healthy and 50 COVID-19 patients X-ray images, all of which were resized to 224×224. The authors employed transfer learning models to get around the issue of the small dataset. The developed DCNN was based on pre-trained models that could recognize COVID-19 from standard X-ray images (ResNet50, Inception V3, and Inceptio ResNet V2). The obtained results demonstrated that the pre-trained ResNet50 model provided an accuracy of around 0.980.

[10] proposed a CNN framework to distinguish COVID-19 cases from other Pneumonia (bacteria and virus) and healthy cases, using the COVDIX dataset [11] (which included 5941 chest X-ray images from 2839 patients). Using a cyclical learning rate, the accuracy achieved was of 0.962.

Also using chest X-ray images, [12] demonstrated a Deep Leaning (DL) model that could distinguish COVID-19 from healthy individuals. Three elements formed the basis of the model: The first is a residual CNN with 18 layers called the backbone network. The principle is to take the chest X-ray image and extract the high-level features from it. The second one is powered by the extracted features and is responsible for generating a classification score while the third and last one detects anomalies and outputs an anomaly score. With both scores, the classification is done using a threshold value.

III. Data Description

The dataset used in the current project was available in Kaggle [13]. The referred dataset is composed by X-ray images of patients that have been diagnosed with COVID-19 (1281 images), viral-pneumonia (1656 images), bacterial-pneumonia (3001 images) and that are healthy (3270 images). These images originate from multiple sources (without duplicates) and because so, have different sizes, brightnesses and ratios, as shows on Fig. 1.

A. Statistical Analysis

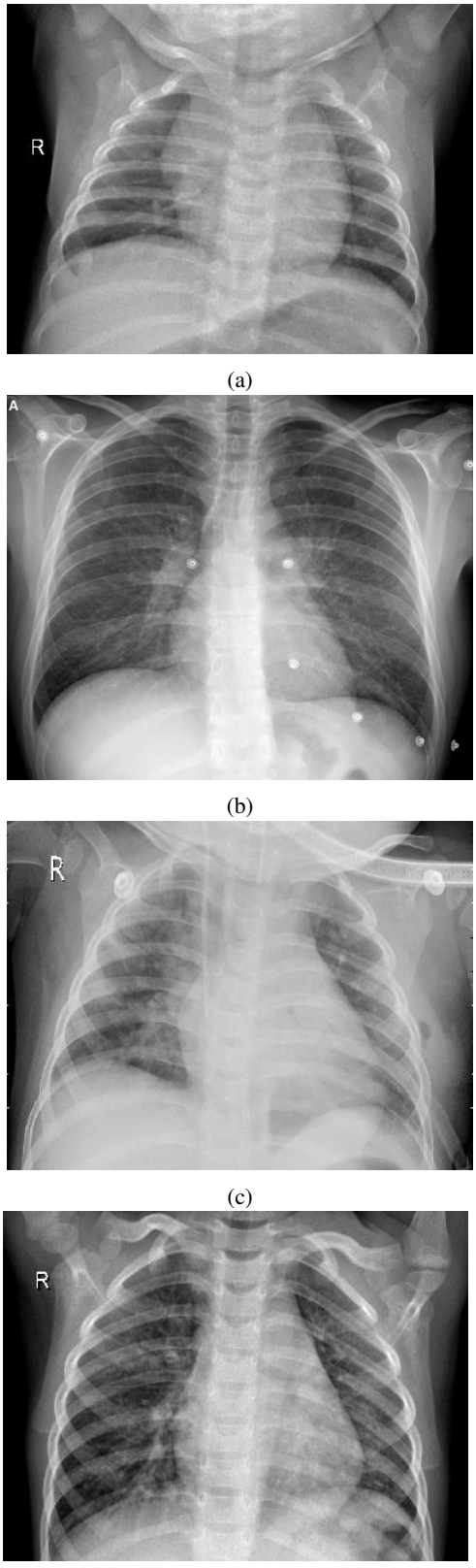
The four classes of the dataset are distributed as follows: Normal 35.51%, Pneumonia Bacterial 32.59%, Pneumonia Viral 17.99% and Covid-19 13.92%. This distribution is displayed, in absolute count, on the histogram of Fig. 2.

IV. Applied Machine Learning Methods

One of the most effective models — i.e., requiring the fewest FLOPS for inference — that achieves State-of-the-Art accuracy on both ImageNet [14] and typical image classification transfer learning tasks is EfficientNet [15].

EfficientNet offers a family of models (B0 to B7) that represent a good trade-off between efficiency and accuracy on a range of scales by introducing a heuristic method for scaling the model. By using such a scaling heuristic, the efficiency-focused base model (B0) can outperform models at all scales without having to perform a thorough grid-search of the hyperparameters.

The authors used the transfer learning technique on top of the EfficientNet, so only the last layer is updated in size and weights. This means that only a small portion of the model parameters are trainable. Namely, 7.56% (331,524 out



(a) (b) (c) (d)

Fig. 1: Example images from the dataset. Patient is healthy (a), with Covid-19 (b), Bacterial Pneumonia (c) and Viral Pneumonia (d).

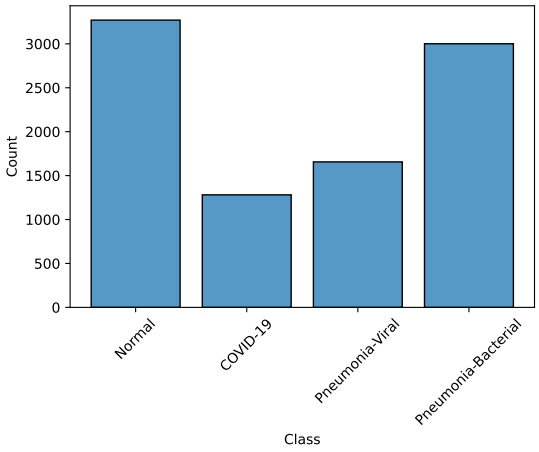


Fig. 2: Histogram of the classes distribution within the dataset.

of 4,383,655) and 3.55% (397,572 out of 11,184,179) for the EfficientNetB0 and EfficientNetB3, respectively.

To infer the benefits of the transfer learning technique, a custom designed CNN was also implemented and trained with the same assumptions as the EfficientNet based models.

A. Data Preprocessing

As described on Table I, the different base models of EfficientNet require different input resolutions and so, the multiple resolutions of the images from the dataset must be resized to the corresponding requirement. For the CNN model the authors chose the same input size has the EfficientNetB3 network.

The textual classes where mapped to numerical values as part of this step.

These are, actually, the only requirements since the EfficientNet expects full RGB images in the range of [0, 255] given that normalization is included as part of the model.

On the other hand, for the CNN model, the input should be a normalized grayscale image.

TABLE I: Image resolution for each EfficientNet base model.

Base Model	Resolution	Channels
EfficientNetB0	224×224	3
EfficientNetB1	240×240	3
EfficientNetB2	260×260	3
EfficientNetB3	300×300	3
EfficientNetB4	380×380	3
EfficientNetB5	456×456	3
EfficientNetB6	528×528	3
EfficientNetB7	600×600	3

B. Data Augmentation

To try avoiding over-fitting, the authors opted to apply some random transformations to some of the base images. These transformations are detailed on Table II.

C. CNN Model

To have a benchmark model, a custom CNN model (see Fig. 3) was implemented with all 5,380,996 parameters trainable.

TABLE II: Values used in the transformations for the data augmentation.

	From	To
Zoom	95%	105%
Brightness	90%	100%
Height Shift	-5%	5%
Width Shift	-5%	5%
Rotation	-10º	10º

The goal of this network is to access if a simpler model with more parameters offers any advantages compared to the models based on the EfficientNet, that have less trainable parameters but have a more capable, pre-trained, backbone.

D. EfficientNet Models

As mentioned previously, the base of the implemented model is the EfficientNet trained with ImageNet dataset. For the problem of this project, the last layer of the base EfficientNet is substituted by a *sub-network* that will be trained with our dataset.

This *sub-network* (see Fig. 4) is composed by a Normalization, a Dropout and two Dense layers. The Normalization layer will normalize the values outputed from the pre-trained network to our specific problem, while the Dropout is there to try to prevent over-fitting the data, but only during the training phase.

E. Training

The training of the models was done in a maximum of 30 epochs. Yet, if in four consecutive epochs, the validation loss did not drop more than 0.01, then the training procedure would reach an early stop.

The authors also decided to reduce the learning rate if the same validation loss did not improve more than 10^{-4} during two consecutive epochs. The initial learning rate was set to 10^{-3} and, at each time a learning rate reduction was required, the previous value was multiplied by 0.1, down to the minimum learning rate of 10^{-7}.

The dataset was, also, split into three subsets: train (81%), validation (10%) and test (9%).

The chosen basis used were the B0 and B3 networks. The EfficientNetB0 was chosen to assess the performance of the most basic (and so, faster to train) model. In counterpart, the EfficientNetB3 was the most complex model of the family that could be trained in a reasonable amount of time.

V. Results

The first major disparity between the training of all models is the amount of epochs taken to reach convergence. As explained above, convergence is reached when the model can not improve the loss on the validation set for four consecutive epochs.

Although the random nature of the training may have some impact on the amount of epochs, it should not explain alone the difference of 10 epochs, i.e., while the model based on the EfficientNetB0 took 17 epochs to reach convergence, the model based on the EfficientNetB3 took 27.

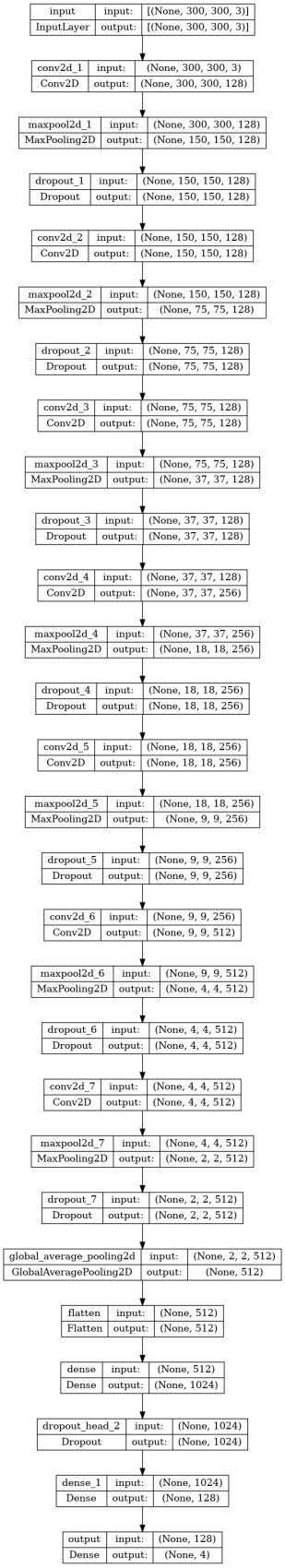


Fig. 3: Full CNN network implemented.

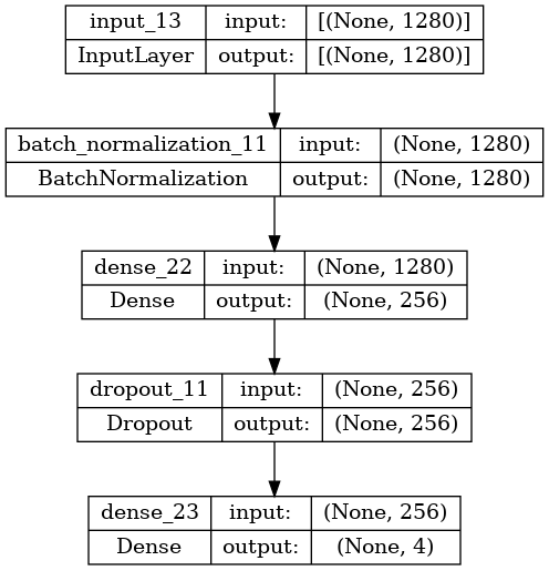


Fig. 4: *Sub-network* attached to the second-to-last layer of EfficientNet.

In contrast, the simpler CNN model – although having far more parameters – reached the early stop criteria at epoch 11. This seams to suggests that the complexity of the model has a greater impact on the training length that the amount of parameters by themselves.

A more detailed comparison between the models follows on the next sections, yet some descriptive metrics of the train models (computed using a macro average) are shown on table III. As expected, the more complex model improves all the metrics compared to all others, at the cost of training time. But, notice, how the simpler CNN model outperforms (in most metrics) the most basic EfficientNet model, despite receiving 67% less pixels as input. This may be the first indication that the pre-trained weights are introducing some added strain on the training and another early stop criteria should be used.

TABLE III: Descriptive metrics for the trained models. Values obtained using macro averages.

	Accuracy	Precision	Recall	F1-score
CNN	0.85	0.83	0.82	0.83
EfficientNetB0	0.84	0.83	0.81	0.82
EfficientNetB3	0.87	0.87	0.85	0.86

A. Loss

The loss function evaluation for either the training and validation data are shown on Figs. 5, 6 and 7 for the CNN, EfficientNetB0 and EfficientNetB3 models, respectively.

Both EfficientNet behaviours are very similar, decreasing sharply in the first few epochs and never diverging from one another.

Interestingly, the CNN model started with far lower losses than the EfficientNet counterparts, almost 75% less, which may be due to pure chance or it is an indication that, in such

early stage, the pre-trained weights are introducing a larger error that the *sub-network* is not yet able to compensate.

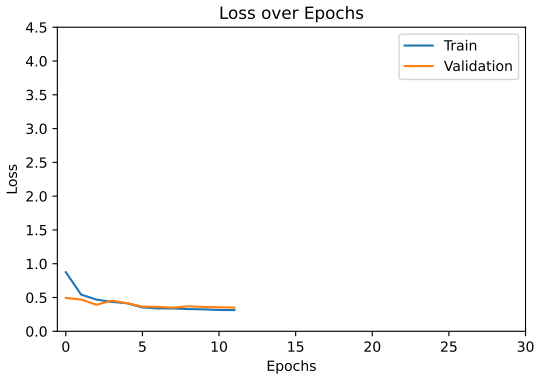


Fig. 5: Loss evolution for the train and validation data, using the CNN model.

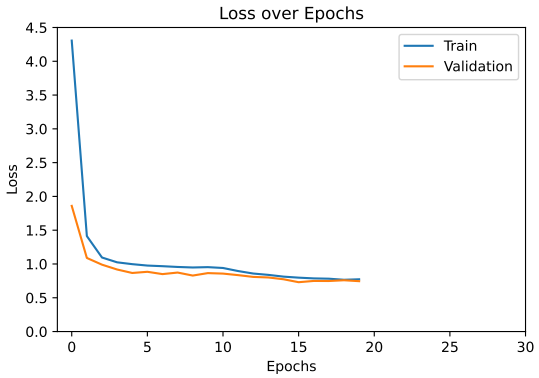


Fig. 6: Loss evolution for the train and validation data, using EfficientNetB0 as basis.

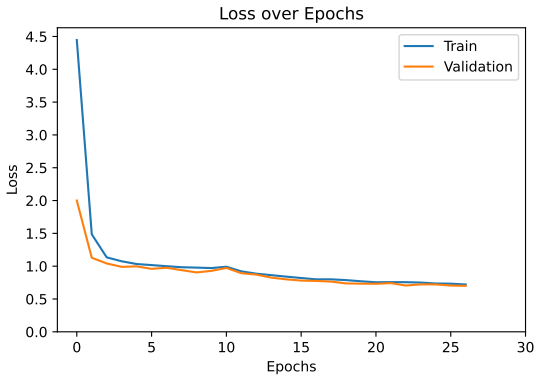


Fig. 7: Loss evolution for the train and validation data, using EfficientNetB3 as basis.

B. Accuracy

Similarly, the evolutions of the accuracy are shown on Figs. 8, 9 and 10.

This metric, for the EfficientNet models, tends to be higher on the validating set, which happens because of the Dropout layer (see Fig. 4). When training, a portion of the features are set to zero, by this layer, to avoid overfitting, at the cost accuracy while training. When validating, this behaviour does not happen and, logically, more features lead to a higher accuracy.

In contrast, the CNN actually switches this trend and ends up with an higher accuracy on the train set than on the validation set. This can be explained twofold: i) in the CNN model there are two Dense layers after the last Dropout layer, while in the EfficientNet models the Dropout layer is directly connected to the output and ii) the random nature of the data split may just selected a ”worst case scenario” for this metric. Solutions and further details will be discussed in the coming sections.

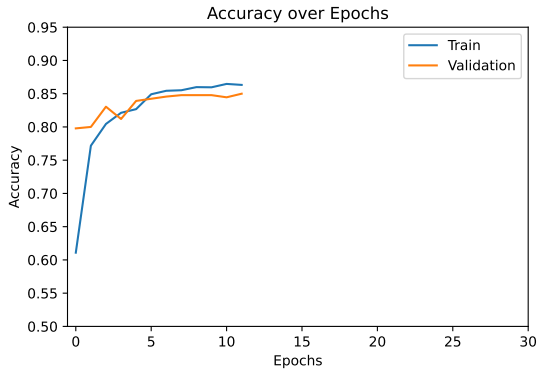


Fig. 8: Accuracy evolution for the train and validation data, using the CNN model.

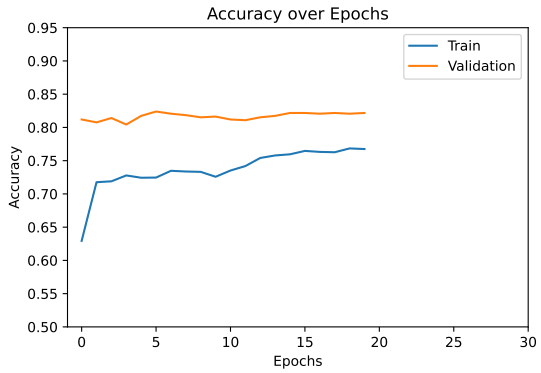


Fig. 9: Accuracy evolution for the train and validation data, using EfficientNetB0 as basis.

C. Learning Rate

As expected, the learning rate behaviours are very similar (see Figs. 11, 12 and 13).

This hyperparameter has a staircase like behaviour that is caused by a stagnation in the validation loss, as explained on section IV-E.

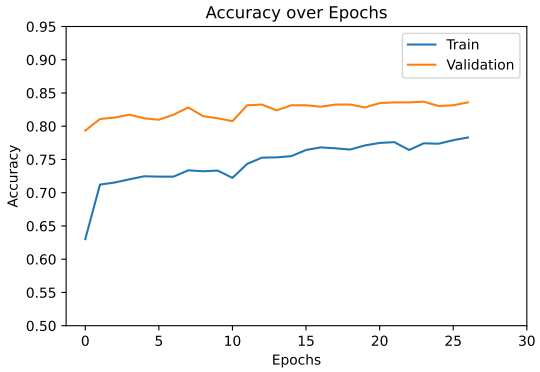


Fig. 10: Accuracy evolution for the train and validation data, using EfficientNetB3 as basis.

Logically, to have four consecutive epochs with stagnation, first there must be two, and that is why all three models have similar curves at the end.

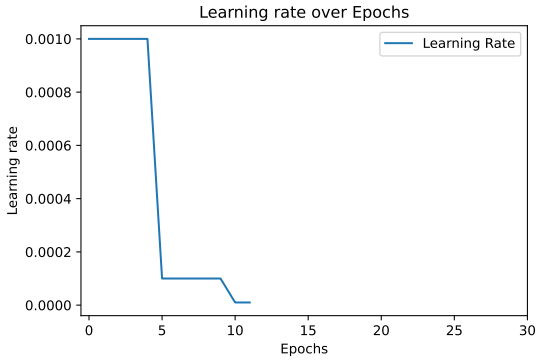


Fig. 11: Learning rate evolution for the train and validation data, using the CNN model.

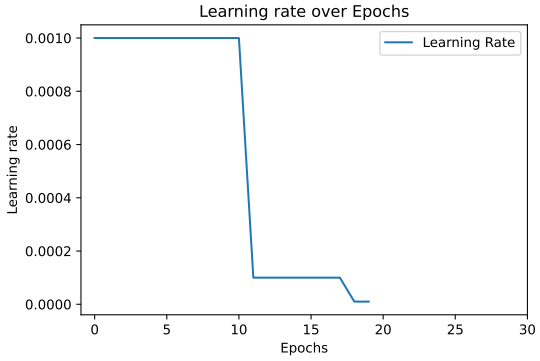


Fig. 12: Learning rate evolution for the train and validation data, using EfficientNetB0 as basis.

D. Confusion

Finally, the confusion matrices are presented on Figs. 14, 15 and 16.

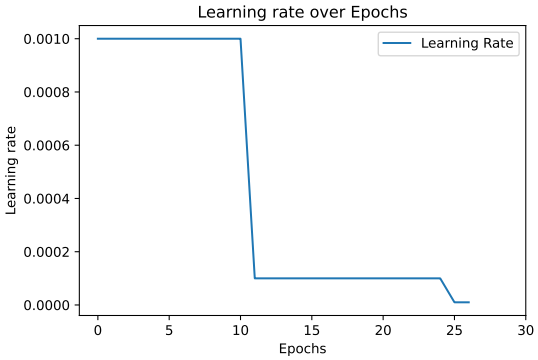


Fig. 13: Learning rate evolution for the train and validation data, using EfficientNetB3 as basis.

These confusion matrices are backed by Table IV. Combined, they state that even though the COVID-19 class was the one with less data, it is the one with the highest F1-score for all models. This metric is lower in both pneumonia classes, which was expected given that, even if the cause is distinct, the disease is the same and the symptoms should be similar, creating a greater difficulty to differentiate them.

It is noticeable that the CNN model implemented reaches higher F1-Scores than the EfficientNetB0 for the Normal and Viral Pneumonia classes, which may be explained by the higher resolution images fed to the prior model.

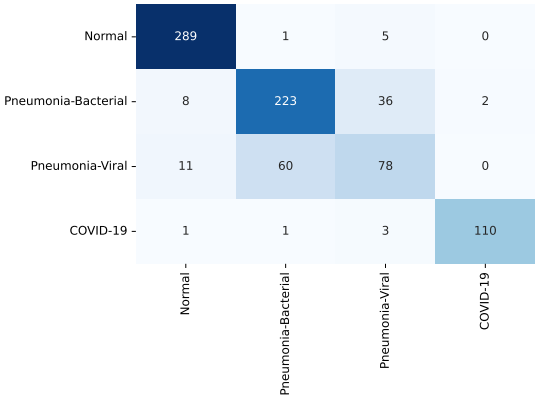


Fig. 14: Confusion matrix for the test data, using the CNN model.

TABLE IV: F1-score comparison of the trained models, for all the classes.

	CNN	EfficientNetB0	EfficientNetB3
Normal	0.96	0.92	0.95
Bacterial Pneumonia	0.81	0.81	0.85
Viral Pneumonia	0.58	0.55	0.65
COVID-19	0.97	0.99	0.97

VI. Conclusion

The advantages of ML, CNNs and the transfer learning technique have become clear.

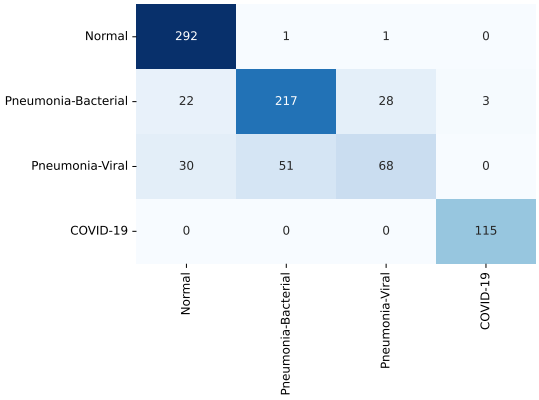


Fig. 15: Confusion matrix for the test data, using EfficientNetB0 as basis.

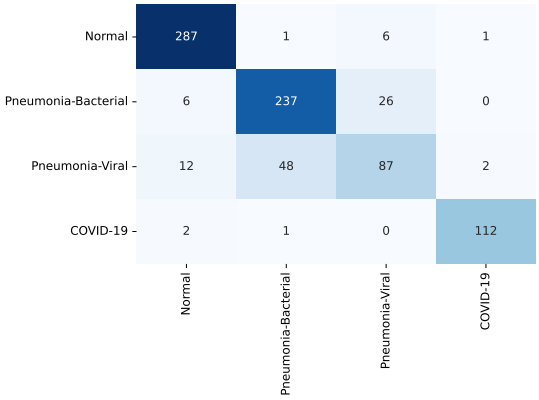


Fig. 16: Confusion matrix for the test data, using EfficientNetB3 as basis.

In this work the authors proved the benefits of using such technologies for an early diagnosis of pneumonia and COVID-19. Although the accuracy reached is not high enough for use in the real world scenarios, room for improvement still exists.

While doctors should never by replaced by ML models, their great contribute to society can be greatly enhanced by working together with this powerful new tools.

Regarding the selected and implemented models, some interesting conclusions can be attained. Contrarily to initially expected by the authors, the transfer learning used in the EfficientNetB0 model provided little or insignificant improvements. This may be because the EfficientNet family is more suited to classification tasks with more categories. The image resolution is also a variable along the models and, certainly, impacts all metrics analysed. Nontheless, the EfficientNetB3 model with transfer learning corresponded to the expectations and delivered the best overall performance.

Moving from the EfficientNetB0 to the EfficientNetB3 provided some improvements, but it remains unclear if the more complex models (like the B5 or even the B7) would provide a bigger jump from the simpler CNN model or not.

Lastly, it has been realized that it is not only the complexity or the number of trainable parameters alone that affect the training time but a balance between these hyper parameters.

A. Future Work

One not explored path in this work was to train the full EfficientNet networks, i.e., the last *sub-network* as well as the backbone network, substituting the pre-trained weights. Given the amount of parameters combined with the size of the dataset this taske would be extremely time intensive, but would clarify the impact of the pre-trained weights on the results achieved.

Additionally, a fixed CNN model should be implemented, with the last Dropout layer directly connected to the output layer, so a direct comparison could be made.

Finally, in insight, the early stop strategy adopted may have led to some inaccuracies in all models. A longer training, with a more difficult to reach stopping criterion, should be adopted to infer the possible benefits that come with it.

References

[1] *Pneumonia*. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/pneumonia.

[2] M. I. Neuman, E. Y. Lee, S. Bixby, *et al.*, "Variability in the interpretation of chest radiographs for the diagnosis of pneumonia in children," *Journal of hospital medicine*, vol. 7, no. 4, pp. 294–298, 2012.

[3] G. J. Williams, P. Macaskill, M. Kerr, *et al.*, "Variability and accuracy in interpretation of consolidation on chest radiography for diagnosing pneumonia in children under 5 years of age," *Pediatric Pulmonology*, vol. 48, no. 12, pp. 1195–1200, 2013.

[4] S. Albahli, H. T. Rauf, A. Algosaibi, and V. E. Balas, "Ai-driven deep cnn approach for multi-label pathology classification using chest x-rays," *PeerJ Computer Science*, vol. 7, e495, 2021.

[5] S. Albahli, H. T. Rauf, M. Arif, M. T. Nafis, and A. Algosaibi, "Identification of thoracic diseases by exploiting deep neural networks," *Neural Netw*, vol. 5, no. 6, 2021.

[6] M. E. Chowdhury, T. Rahman, A. Khandakar, *et al.*, "Can ai help in screening viral and covid-19 pneumonia?" *IEEE Access*, vol. 8, pp. 132 665–132 676, 2020.

[7] A. Waheed, M. Goyal, D. Gupta, A. Khanna, F. Al-Turjman, and P. R. Pinheiro, "Covidgan: Data augmentation using auxiliary classifier gan for improved covid-19 detection," *Ieee Access*, vol. 8, pp. 91 916–91 923, 2020.

[8] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, "Explainable deep learning for pulmonary disease and coronavirus covid-19 detection from x-rays," *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105 608, 2020.

[9] A. Narin, C. Kaya, and Z. Pamuk, "Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 1207–1220, 2021.

[10] M. Farooq and A. Hafeez, "Covid-resnet: A deep learning framework for screening of covid19 from radiographs," *arXiv preprint arXiv:2003.14395*, 2020.

[11] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific Reports*, vol. 10, no. 1, pp. 1–12, 2020.

[12] J. Zhang, Y. Xie, Y. Li, C. Shen, and Y. Xia, "Covid-19 screening on chest x-ray images using deep learning based anomaly detection," *arXiv preprint arXiv:2003.12338*, vol. 27, 2020.

[13] A. Kolas, *3 kinds of pneumonia*, May 2022. [Online]. Available: https://www.kaggle.com/datasets/artyomkolas/3-kinds-of-pneumonia.

[14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.

[15] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, PMLR, 2019, pp. 6105–6114.