UNIVERSITÉ MOHAMMED V - Rabat

Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes

Master de Recherche

Cloud And High Performance Computing

Automatic Detection Of COVID-19 Using Convolutional Neural Networks And X-Ray Images

Realized By :

AHSAINE Khadija

Pr. OUACHA Ali	FSR	*President*
Pr. EZ-ZAHOUT Abderrahmane	FSR	*Examinator*
Pr. ZBAKH Mostapha	ENSIAS	*Supervisor*
Pr. TADONKI Claude	Mines ParisTech	*Supervisor*

Academic Year : 2020- 2021

ABSTRACT

The Covid-19 first occurred in Wuhan, China, in December 2019. After that, the virus spread all around the world. The usage of sophisticated artificial intelligence technology (AI) and the radiological images of COVID-19 cases can provide experts with auxiliary diagnosis suggestions, reducing the burden of experts to a certain amount and speeding up the diagnostic process. In this study, transfer learning methods have been implemented, with six pre-trained deep convolutional neural networks models: VGG16, ResNet50, InceptionV3, InceptionResNetV2, MobileNet, and EfficientNetB0 on a dataset of chest X-ray images of patients with COVID-19 disease, viral pneumonia, and normal cases. Transfer learning is an ambitious task, and it results in impressive outcomes for identifying distinct patterns in tiny datasets of medical images. The experimental results show that MobileNet pre-trained model achieved a relatively ideal effect, and the diagnostic accuracy reached 96.05% outperforming all other models.

Keywords : CNN, COVID-19, X-ray images, Trasfer learning, Pre-trained models, Google Colab, GPU

ACKNOWLEDGMENT

It is pleasant to repay a debt of gratitude to all the people whose intervention during this project facilitated its completion. My most sincere thanks go to my academic supervisors Mr. Mostapha ZBAKH and Claude TADONKI, who helped me, directed me to deliver good quality work. Moreover, encouraged me to complete my goal and improve my knowledge. We would also like to thank the entire teaching team of the Master CloudHPC for ensuring the program's theoretical and practical parts.

Our gratitude goes to those who have provided emotional support for this work: family and friends.

Table Of Content

List Of Figures

List Of Tables

INTRODUCTION

Coronaviruses (CoV) are a series of viruses that cause infections of the respiratory system, such as Middle East Respiratory Syndrome (MERS-CoV) and Severe Acute Respiratory Syndrome (SARS-CoV). COVID-19 is a new type discovered in 2019 and caused by a virus called Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-CoV-2). Covid-19 has become a Global Health Emergency of International Concern since its emergence in Wuhan, Hubei Province, China in December 2019. The number of deaths from the infection is escalating at a worrying rate, and many health systems worldwide are struggling to cope. The World Health Organization (WHO) suggests some precautionary measures to control the spread of this viral infection, respect the social distance and isolate the affected and suspected cases are among them. A critical step, in this case, is an effective and accurate screening of the COVID-19 patients, so positive cases obtain timely treatment and get quarantined. Reverse-transcription polymerase chain reaction (RT-PCR) testing is currently the most used technique for COVID-19 detection. The high sensitivity of RT-PCR testing is limited by the lack of test kits, and t it is a very time-consuming (few hours to a day or two), laborious, and complicated manual process. Hence, there is an increasing need to use rapid and reliable examination techniques that can be used as an alternative to RT-PCR. Some studies have suggested using imaging techniques such as Computed Tomography (CT) or X-rays scans of the chest to look for visual signs associated with Covid-19 infection.

Usually, the CT imaging technology demands more time than the X-ray, and real-time X-ray imaging can significantly accelerate infection screening. In addition, many less developed regions may not have high-quality CT scanners. Since X-ray material is low in cost and simple to function, most outpatient clinics and institutions have X-ray equipment as the necessary imaging equipment. Compared with CT imaging, X-ray imaging is the most common and widely used diagnostic imaging technique and plays an essential role in clinical nursing and epidemiological research [1]. Therefore, this paper selected chest X-ray images to diagnose and screen COVID-19. However, radiologists and experts have a different understanding of the same image based on their personal clinical experience when analyzing X-ray images. Therefore, there is an urgent need for an artificial intelligence-based diagnosis system to help radiologists interpret images faster and more accurately.

Integrating computer-aided diagnosis (CAD) based on deep learning methods into radiologist diagnostic systems will reduce the workload of doctors and increases reliability and quantitative analysis. Deep learning techniques and medical imaging are becoming more and more research fields[2].

The structure of this paper is as follows: a brief overview of CNN architectures is mentioned in Section 1. The methodology of the work is illustrated in Section 2, including dataset and proposed network. Section 3 defines the results of this research as well as the discussions of the obtained results. The conclusion of this analysis is summarized in Section 4.

Chapter I : Convolutional Neural Networks Overview

1. Related Works

Since the start of the COVID-19 pandemic, researchers quickly shared their effort on combating it by focusing on developing a vaccine in one hand and detecting COVID-19 using PCR and imaging systems on the other hand [3]. Here, we review studies devoted to using radiography images and deep learning, especially convolutional neural networks (CNNs), to aid in rapid detection of the infected cases and help in their quick isolation, which will reduce the infection spread. With 224 COVID-19 verified images, Apostolopoulos et al. [4] realized an automatic decision system to predict Covid-19 from X-Ray images using transfer learning with convolutional neural networks. The study intends to evaluate the performance of state-of-the-art Convolutional Neural Network architectures proposed over recent years for medical image classification. The results showed that VGG-19 outperforms other CNNs models. Narin et al. [5] implemented three different deep CNN models: ResNet-50, InceptionV3, and Inception-ResNetV2, where they also used transfer learning to detect COVID-19 and accomplished 98% COVID-19 identification using chest X-ray imaging in conjunction with the ResNet 50 model. Zhang et al. [1] develop a deep learning model applied on chest X-ray images that can detect COVID-19 with high sensitivity, affording fast and reliable scanning.

Ali et al.[6] built a deep convolutional neural network (CNN) based on ResNet50, InceptionV3, and Inception-ResNetV2 models to classify Chest X-ray images to COVID-19 and normal classes. They reached a good correlation between CT image results and PCR approach. They used a dataset that contains 50 Chest X-ray images of COVID-19 patients; it has been obtained from an open-source GitHub repository shared by (Dr. Joseph Cohen[7]). Wang et al. [8] created a new model architecture COVID-net and established a larger dataset called COVIDx (consisting of 13,800 chest X-ray images). The purpose is to distinguish between normal, pneumonia, and COVID-19 X-ray images. The results showed that the diagnostic accuracy of COVID-19 achieved 92.4%. El-Din Hemdan et al. [9] compared several traditional deep learning classification frameworks and pre-trained the model in the ImageNet dataset [10] to identify normal and COVID-19. They selected 25 images for healthy patients and 25 for COVID-19 positive patients. In the model chosen by the authors, VGG19 and DenseNet had similar performance, with an F1-Score of 0.89 for normal and 0.91 for COVID-19. Sahinbas et al. [11] used X-ray images to diagnose COVID-19 and worked on VGG16, VGG19, ResNet, DenseNet, and InceptionV3 models. Farooq et al. [12] proposed a fine-tuned ResNet-50 architecture, which classifies chest X-rays images into normal, COVID-19, bacterial pneumonia, and viral pneumonia. Compared with COVID-net [8], the authors report better accuracy. In [13] , Alqudah et al. used two different methods to diagnose COVID-19 using chest X-ray images. The first one used AOCTNet, MobileNet and ShufeNet CNNs. Secondly, the features of their images have been extracted, and they have been using softmax classifier, K nearest neighbor (kNN), support vector machine (SVM), and random forest (RF) algorithms to classify them. Barstugan et al. [14] classified X-ray images to predict COVID-19 using five different feature extraction methods: Grey-Level Co-occurrence Matrix (GLCM), Local Directional Patterns (LDP), Grey-Level Run Length Matrix (GLRLM), Grey-Level Size Zone Matrix (GLSZM), and Discrete Wavelet Transform (DWT). The obtained features have been classified using SVM. During the classification process, k-fold cross-validation methods were used.

While many proposals have already been developed and implemented for SARS-CoV-2 recognition, there is still a possibility for performance improvement for different datasets. Therefore, this study aims to establish an automatic prediction system of COVID-19 using deep convolution neural network-based pre-trained transfer models and chest X-ray images. For this purpose, we have used ResNet50, VGG16, MobileNet, EfficientB0, InceptionV3, and InceptionResNetV2 pre-trained models to obtain higher prediction accuracies for three different cases : normal (healthy), COVID-19, and viral pneumonia (non-COVID-19) patients.

The results of the proposed study are summarized below:

- This work has been studied and compared with 6 different CNN models.
- A high-accuracy prediction system has been proposed to radiologists to automatically diagnose and detect patients with COVID-19.
- It has used more data than many other studies in the literature.

2. Deep Transfer Learning Architecture

Transfer learning is an approach to solve the large training data requirement issue for image classification problems. Transfer learning refers to the approach that utilizes models trained on one problem as a starting point to solve related ones. This approach can significantly reduce the data required for domain-specific tasks. It presents a promising solution to reduce the training data requirement and training time consumed [29], and it is widely used with convolutional neural networks (CNN). CNN architectures consist of a set of different layers:

a) Convolutional Layer

The convolutional layer is the base layer of CNN. It is responsible for extracting the features of the images. Inside this layer, the input image is passed through a filter to generate the feature map. Numbers of filters are slid through the images to extract low and high-level features [15]. The stride parameter is the number of steps tuned for moving over the input matrix. The output of the convolutional layer can be presented as [5]:

$$x_j^{\,l} = f\left(\sum_{a=1}^{N} w_j^{l-1} * y_a^{l-1} + b_j^l\right)$$

Where x_j^l is the $j-th$ feature map in layer l, w_j^{l-1} indicates $j-th$ kernels in layer $l-1$, y_a^{l-1} represents the $a-th$ feature map in layer $l-1$, b_j^l indicates the bias of the $j-th$ feature map in layer l, N is the number of total features in layer $l-1$, and $(*)$ represents vector convolution process.

b) Pooling Layer

The pooling layer is the second layer after the convolutional layer. It is usually applied to reduce the size after the convolution of feeding images by applying corresponding mathematical computation. It can be used in between two convolutional layers. If applying a fully connected layer after two convolutional layers without applying average or max pooling, then the computations and amount of the parameters will be very high. In this study, we used max-pooling. The max-pooling selects only the maximum value using the matrix size specified in each feature map, reducing the output neurons.

c) Flatten Layer

After finishing the previous two steps, we are supposed to have a pooled feature map by now. Flatten layer is used before the fully connected layer, reducing data to a single dimension.

d) Fully-Connected Layer

The fully-connected layer is the last and most crucial layer of CNN. This layer works like a multi-layer perceptron, where neurons have full connections to all activation functions in the previous layer. Rectified Linear Unit (ReLU) activation function is usually used on fully-connected layer, while the Softmax activation function is used to predict the class in the last layer of fully-connected layer. Mathematical representation of these two activation functions are as follow [2]:

$$ReLU\ (x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

$$Soft\max(x_i) = \frac{e^{x_i}}{\sum_{y=1}^{m} e^{x_y}}$$

where x_i and m represent input data and the number of classes, respectively.

3. Pre-Trained Models

In recent years, state-of-the-art CNNs and deep learning techniques, including pre-trained networks, have dominated the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Starting from the famous AlexNet in 2012 [16], more advanced architectures such as VGG-16, ResNet50, InceptionV3, InceptionResNetV2, MobileNet, and EfficientNet represented some of the highest performing techniques over the past few years [17]. These models, besides pre-trained weights and other functions, are provided by the Keras library [18].

a) VGGNet

Visual Geometry Group Network (VGG) was developed by Oxford Robotics Institute's Karen Simonyan and Andrew Zisserman [20]. It was presented at the 2014 Large Scale Visual Recognition Challenge (ILSVRC2014). The VGGNet performed very well on the ImageNet dataset. In order to enhance image extraction functionality, the VGGNet used smaller filters of 3×3, opposed to AlexNet 11×11 filter. There are two versions of this architecture; namely, VGG16 and VGG19 have different depths and layers. VGG19 is deeper than VGG16. However, the number of parameters for VGG16 is lower and thus less expensive than VGG19 to train the network [9].

b) ResNet

ResNet architectures are suggested by He et al. [19] from Microsoft and won 2015 ILSVRC. The main innovation in ResNet architectures is the use of residual layers and skip connections to solve the problem of vanishing gradients that may result in stopping the weights in the network to further update/change. This is mainly a problem in deep networks because the gradient value can vanish, i.e., shrink to zero, when several chain rules are applied consecutively. Skipping connections will help gradients flow backward directly from end layers to initial layer filters enabling CNN models to deepen with 152 layers.

c) InceptionNet

Inception network or GoogLeNet is the winner in the 2014 Image net challenge with 93.3% top-5 accuracy. It was a 22-layer network [21], and it has proved to be more computationally efficient, both in terms of the number of parameters generated and the resources consumed by the network. Any changes in the Inception network need to assure computational advantages; therefore, the adaptation of the model for different use cases turns out to be a problem due to the uncertainty of the new network’s efficiency. Inception v3 architecture suggested several techniques for optimizing the network to loosen the constraints for easier model adaptation, among these techniques: factorized convolutions, regularization, dimension reduction, and parallelized computations [22].

d) Inception-ResNetV2

Inception-ResNetV2 was proposed by Szegedy et al. [23] in 2016, combining the idea of inception blocks and residual layers. The purpose of using residual connections is to prevent the problem of degradation caused by deep networks and reduce the training time. The InceptionResNetV2 model used here contains 20 Inception-ResNet blocks that empower the network to become 164 layers deep, and we use the pre-trained weights in these layers to assist our mission of detecting COVID19 in X-Ray images [3].

e) MobileNet

MobileNet is a class of CNN that Google open-sourced. As the name applied, the MobileNet model is designed to be used in mobile applications, and it is TensorFlow's first mobile computer vision model. MobileNet uses depthwise separable convolutions. It significantly reduces the number of parameters compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks.

f) EfficientNet

The paper [24] released by Google introduces a new principle method to scale up ConvNets. To get better accuracy, CNN needs to have a careful balance between depth width and resolution. However, the scaling-up process of ConvNets has never been understood. The most common way was to scale up ConvNets by their depth or width. Another less common method was to scale up models by image resolution. So far, we only scale one dimension at a time [25]. There are eight alternative versions of EfficientNet (B0 to B7), and even the simplest one, EfficientNetB0, is outstanding. It achieves a 77.1% Top-1 accuracy performance With 5.3 million parameters.

4. Conclusion

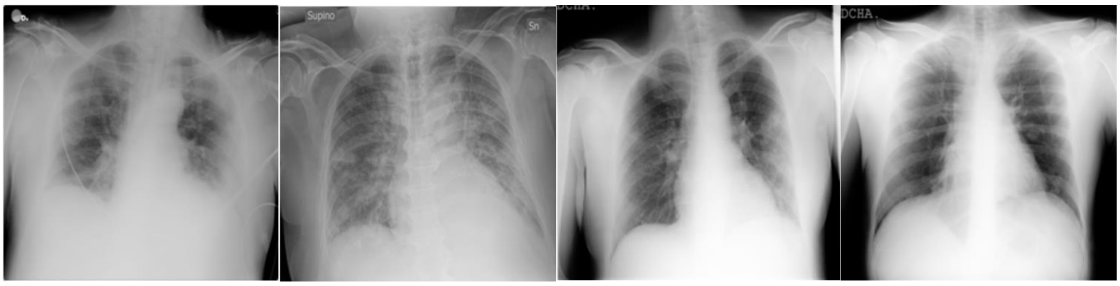
Transfer learning is a widely used technique in image recognition applications to minimize the time consumed during the training and overcome the problem of lack of data. This Chapter represents an overview about CNN. It touched the different layers and some efficient architectures of convolutional neural networks.

Deep learning methods are widely used in medical applications, but one of the most considerable difficulties researchers face is the insufficient number of available datasets. Deep learning models often require a lot of data. Labeling this data by experts is both costly and time-consuming. The biggest benefit of using the transfer learning method is that it allows the training of models with fewer samples and requires less calculation costs. With the transfer learning technique, the weights gained by the pre-trained model on a large dataset are transferred to the model to be trained.

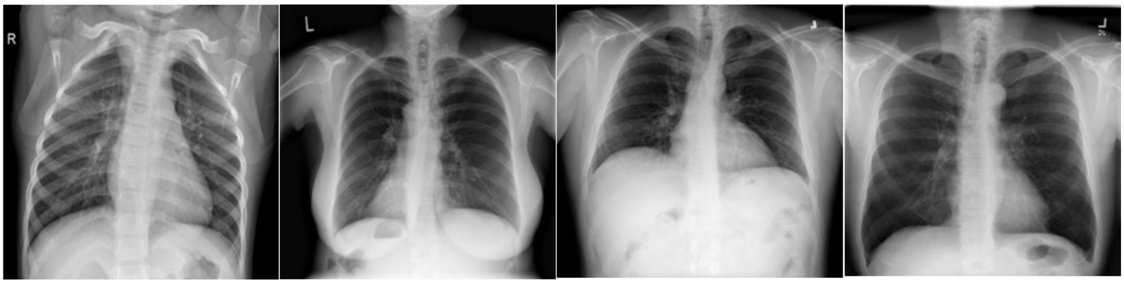
1. Dataset Description

The benchmark data set [26] used in our experimental evaluation has been created by A team of researchers from Qatar University, the University of Dhaka, Bangladesh, and their collaborators from Pakistan and Malaysia in collaboration with medical doctors. It consists of four categories: COVID-19, Normal, Viral Pneumonia, and other lung infections. This dataset is released in stages. The first release contained 219 COVID-19, 1341 normal, and 1345 viral pneumonia chest X-ray (CXR) images. In the first update, the COVID-19 class has been increased to 1200 CXR images. In the 2nd update, the database has been expanded to 3616 COVID-19 positive cases along with 10,192 Normal, 6012 Lung Opacity (Non-COVID lung infection), and 1345 Viral Pneumonia images.

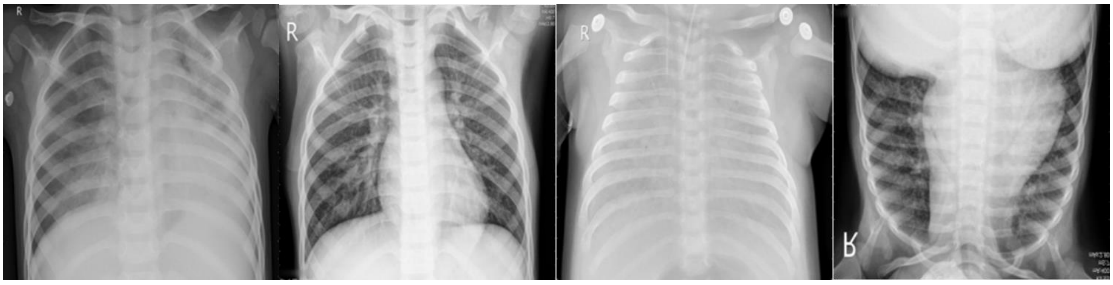
In our experiments, we considered COVID-19, Normal, and Viral Pneumonia classes, with 1345 chest X-ray images in each: 1076 for training and 269 for testing. Since resizing is one of the vital steps in data preprocessing, all images were resized to 224x224 pixel size in the datasets. In Figure 1, representative chest X-ray images of Normal, COVID-19, Viral Pneumonia patients are given, respectively.



(a) Chest X-ray images of COVID-19 patients



(b) Chest X-ray images of Normal patients



(c) Chest X-ray images of Viral Pneumonia patients

Figure 1: *Samples of used chest X-ray images dataset. a) COVID-19, b) Normal, c) Viral Pneumonia*

2. Data augmentation

As the model's network increases, the parameters to be learned will also increase, which will quickly lead to overfitting. In this case, to solve the overfitting problem caused by the small number of training images, we have implemented data augmentation. Data augmentation helps in creating artificially newer examples when the dataset is small by applying different transformations to the available training images. Data augmentation is done by using shear range of 0.2, zooming throughout 0.2, and horizontal flipping.

3. Proposed Multi-Classification Network

In this study, we built deep CNN-based MobileNet, InceptionV3, ResNet50, EfficientNetB0, Inception-ResNetV2, and VGG16 models for the classification of Chest X-ray images to three classes : COVID-19, Normal and Viral Pneumonia. In addition, we applied the transfer learning method that was realized by using ImageNet data to overcome the limited data and training time. The schematic representation of six pre-trained CNN models to predict COVID-19, Normal and Viral Pneumonia patients is shown in Figure 2.

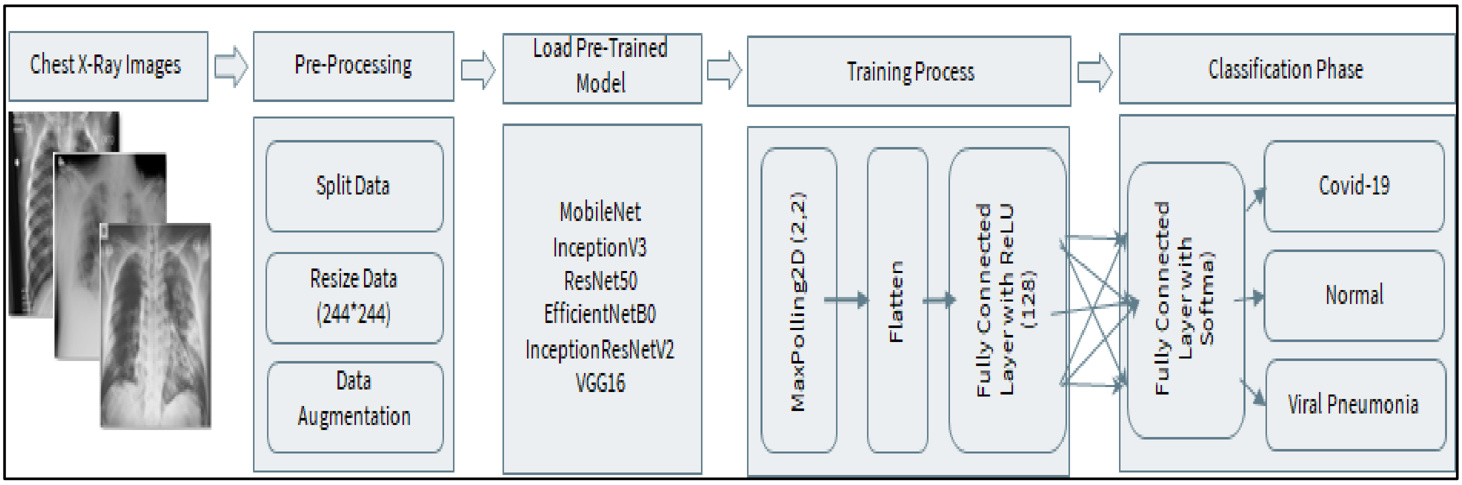


Figure 2: *Transfer Learning global architecture used for 3-class classification*

a) Input and Pre-Processing Steps

Since the properties of the image (width and height) vary from one image to another, therefore, the study has used a fixed size of 224 x 224 pixels. Following that, 80% of the data are used as the training dataset, and 20% are used to evaluate the trained model. Finally, we apply three data augmentation techniques to obtain more training data: shearing, zooming, and flipping.

b) Pre-trained Models

Pre-trained models are trained on a large benchmark dataset as a starting point to solve different problems. In this study, eight different pre-trained models (e.g. MobileNet, InceptionV3, ResNet50, EfficientNetB0, InceptionResNetV2, and VGG16) have been used for multi-classification (3-class). All the models have different convolution, pooling layers that extract the features from images and classify the images from extracted features.

c) Training and Classification Process

In the final step, we feed the pre-trained models with a balanced dataset of COVID-19, Normal, and Viral Pneumonia chest X-ray images. In the training and classification phase, MaxPooling2D has been used in all the models to calculate the average for each patch of the feature map with pool size (2, 2). Afterward, we flattened the activations to create a vectorized feature map and connected two fully connected layers; one layer contained 128 nodes, and the other consisted of 3 for 3-class classification. Subsequently, the second fully connected layer activations were fed into a softmax layer, which provided the probability for each of normal, COVID-19, and viral pneumonia.

4. Experimental Setup

Python programming language was utilized for training the deep transfer learning models. All experiments were executed on Google Colaboratory (Colab) as an editor for executing the codes, which was connected to local resources. The background running environment is built up using deep learning framework TensorFlow (2.6.0) and Keras package [18] on a GPU. The machine that has been used in the experiments has a CPU Intel Core i7-8700 with 12 Threads - (16 GB/1 TB HDD/256GB SSD/Windows 10 Professionnel 64-bit/8 GB Graphics) and is equipped with a GPU NVIDIA GeForce RTX 2070 SUPER (Table 1).

The models were trained with ImageNet initialization weights using the ADAM optimizer with default parameters. The batch size and learning rate are experimentally set to 256, 0.001, and the number of epochs is set to 30 to avoid overfitting for all experiments.

Machine Name	Supermicro X11
Operating System	Windows 10 Professionnel 64-bit
Processor	Intel(R) Core(TM) i7-8700
Cores	6
Threads	12
RAM	16 GB
Graphic Card	NVIDIA GeForce RTX 2070 SUPER
VRAM	8 GB
Disk	1000 GB (HDD) 256 GB (SSD)

Table 1*: Machine Configuration*

5. Performance Metrics

To test the classification performance of pre-trained models that we have used in this study, the following metrics have been implemented to show the classified or misclassified cases. The performance metrics are calculated based on True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values.

- **Accuracy**

It measures the ratio of correctly classified cases compared to the whole dataset. If the accuracy is higher, that means the models perform better. The accuracy is a portion of the predicted or classified value to its actual value. It is represented as follows:

$$\mathrm{Acc} = \frac{TN + TP}{TN + TP + FN + FP}$$

- **Precision**

It measures the percentage of correctly classified as positive out of all positive cases. It is defined as follows:

$$\text{Pre} = \frac{TP}{TP + FP}$$

- **Recall**

The recall is computed as the ratio of positives that were correctly predicted as true positives divided by the number of actual positives. It is calculated as follows:

$$\text{Rec} = \frac{TP}{TP + FN}$$

- **F1-Score**

F1-Score is calculated based on the scores of precision and recall. It provides the classification capability of the model. F1-Score measures the test's accuracy. If the F1 score presents the best value, that means perfect precision and recall. It is calculated as follows:

$$\text{F1} - \text{S} = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall}\right)$$

- **Specificity**

Specificity is also called True Positive Rate (TPR), which measures the ratio of actual negatives that are correctly labeled. It is represented as follows:

$$\text{Spe} = \frac{TN}{TN + FP}$$

TP is the proportion of positive cases that are correctly classified as positive ; FP is the proportion of negative cases that are misclassified as positive ; TN is the proportion of negative cases that is correctly classified as negative, and FN is the proportion of positive cases that is misclassified as negative by the proposed model.

6. Conclusion

In this chapter we present the methodology used in this project. First we describe data and different data augmentation techniques that have been used. The third section give details of the global architecture of this project. The machine configuration used is given in the fourth section. Finally, the different performance metrics to evaluate the models is represented in section five.

CHAPTER III : RESULTS AND DISCUSSION

1. Class Classification Training, and Validation Accuracy and Loss

Figure 3 shows the training and validation accuracy with their loss values for the 3-class classification based on the six pre-trained models (ResNet50, VGG16, MobileNet, EfficientB0, InceptionResNetV2, and InceptionV3). The running time intervals for all pre-trained models were set into the 30th epoch to avoid overfitting of the models. Considering the overall classification accuracy, the MobileNet model (Figure 3A) showed the highest validation accuracy (96.05 %) compared to the VGG16 (95.29 %) (Figure 3F), ResNet50 (95.04 %) (Figure 3C), EfficientNetB0 (94.92 %) (Figure 3D), InceptionResNetV2 (89.59 %) (Figure 3B), and InceptionV3 (89.34 %) (Figure 3E). As a result, the MobileNet model provides an advantage over the other pre-trained models used in multi-classification networks in both training and validation stages. A comparison of training and testing accuracy and loss of the six models for 3-class classification is shown in Figure 4. While a lot of oscillation is observed in test data in some models, some models are more stable. The MobileNet and EfficientB0 models appear to have less oscillation.

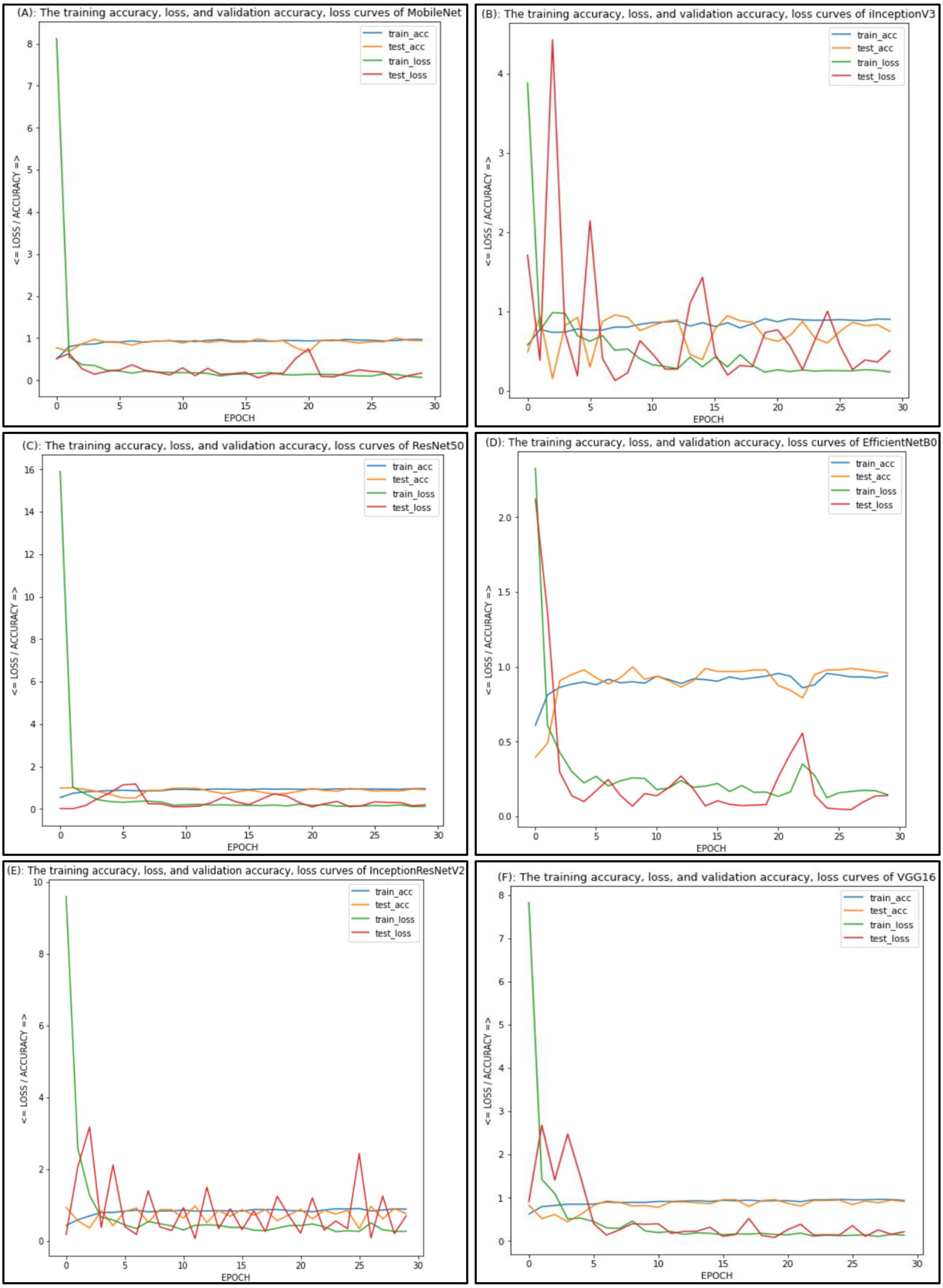


Figure 3: *The training accuracy (train_acc), loss (train_loss), and validation accuracy (val_acc), loss (val_loss) curves of all pre-trained models for 3 classes: (A) MobileNet, (B) InceptionV3, (C) ResNet50, (D) EfficientNetB0, (E) InceptionResNetV2, and (F) VGG16*

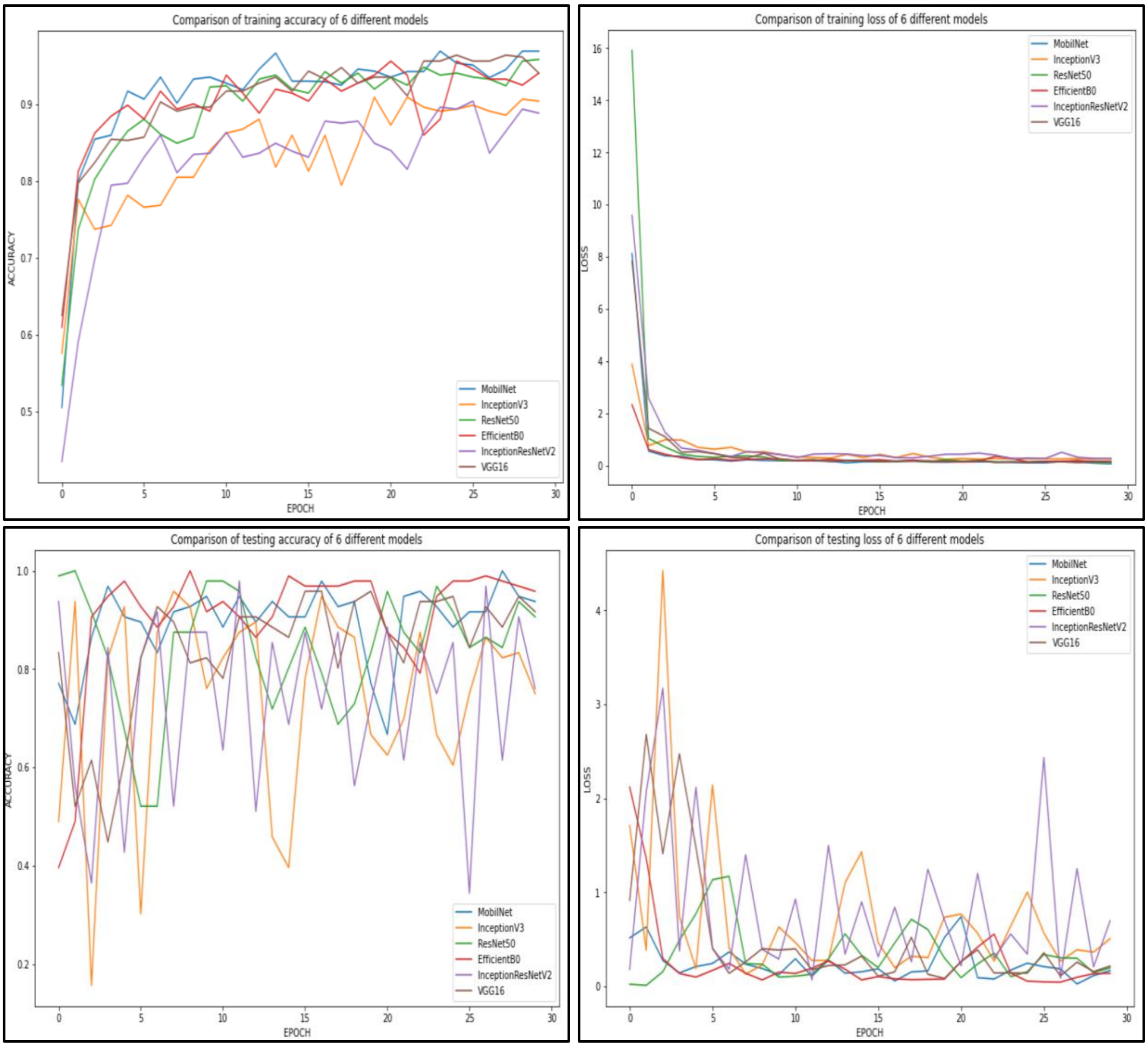


Figure 4*: Comparison of training and testing accuracy and loss values of 6 different models for 3-class classification*

2. Class Classification Confusion Matrix

Experimentally, we presented the performance of our classifiers using a confusion matrix, which is a common evaluation metric for classification. Rows of the confusion matrix correspond to an actual class while columns represent the predicted class and the color intensity specifies the probability of each element in a row. It can be observed from the confusion matrix of 3-class classification (Figure 5) that the MobileNet and EfficientNetB0 pre-trained models classified COVID-19 cases better than other models. They were able to classify 255 cases out of 269 cases, only 14 cases were misclassified. However MobileNet outperforms EfficientNet in predicting the Normal class.The worst model in predicting Covid-19 is InceptionResV2, it lost 64 cases.

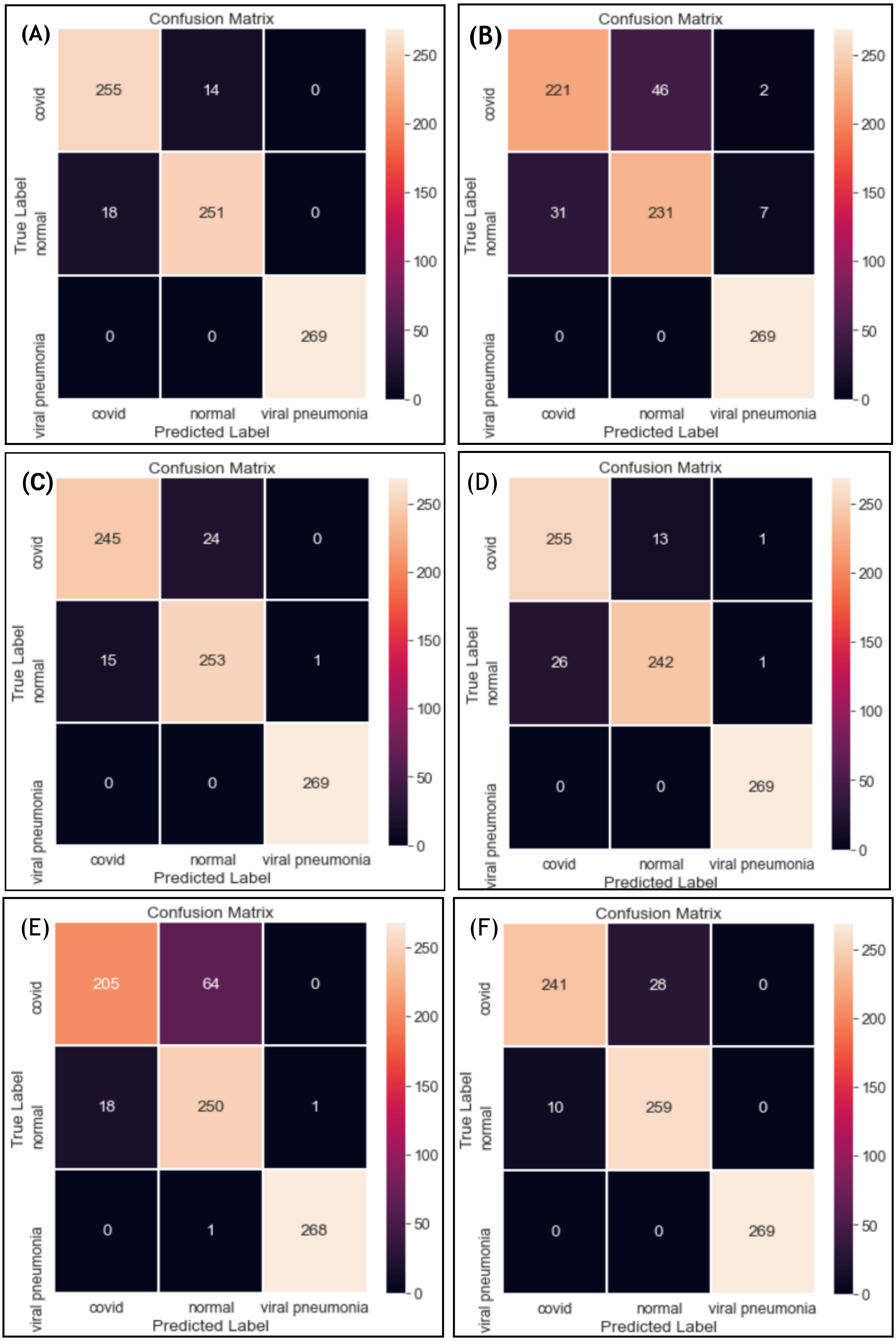


Figure 5: *The confusion matrix results of 3-class classification obtained using pre-trained models (A) MobileNet, (B) InceptionV3, (C) ResNet50, (D) EfficientNetB0, (E) InceptionResNetV2, and (F) VGG16*

3. Class Comparative Performance Metrics of Pre-Trained Models

Table 2 demonstrates in detail the performance metrics of the six pre-trained models used in the proposed network for 3-class classification. It can be noticed from the performance metrics of pre-trained models that the best classification performance was recorded by the MobileNet and VGG16 model. Most models reached high values in detecting viral pneumonia.

Classification Model	Classes	TP	TN	FP	FN	Prec	Rec	F1-S	Spe
MobileNet	Covid-19	255	520	18	14	93%	95%	94%	97%
	Normal	251	524	14	18	95%	93%	94%	97%
	Viral Pneumonia	269	538	0	0	100%	100%	100%	100%
InceptionV3	Covid-19	221	507	31	48	88%	82%	85%	94%
	Normal	231	492	46	38	83%	86%	85%	92%
	Viral Pneumonia	269	529	9	0	97%	100%	98%	98%
ResNet50	Covid-19	245	523	15	24	94%	91%	93%	97%
	Normal	253	514	24	16	91%	94%	93%	96%
	Viral Pneumonia	269	537	1	0	100%	100%	100%	100%
EfficientNetB0	Covid-19	255	512	26	14	91%	95%	93%	95%
	Normal	242	525	13	27	95%	90%	92%	98%
	Viral Pneumonia	269	536	2	0	99%	100%	100%	100%
InceptionResNet V2	Covid-19	205	520	18	64	92%	76%	83%	97%
	Normal	250	473	65	19	79%	93%	86%	88%
	Viral Pneumonia	268	537	1	1	100%	100%	100%	100%
VGG16	Covid-19	241	528	10	28	96%	90%	93%	99%
	Normal	259	510	28	10	90%	96%	93%	95%
	Viral Pneumonia	269	538	0	0	100%	100%	100%	100%

Table 2: *All performances of 6 different models for 3-class classification. True Positive (TN), True Negative (TN), False Positive (FP), False Negative (FN), Accuracy (ACC), Recall (REC), Specificity (SPE), Precision (PRE), F1-Score (F1)*

The results (Table 3) show that the MobileNet pre-trained model showed an accuracy of 97.02% in detecting COVID-19 for 3-class and the achieved precision, recall, F1-score, and specificity values were 96%, 96%, 96%, and 98.02%, respectively. In addition, less training time was achieved by MobileNet, followed by VGG16, EfficientNetB0, and ResNet50. Inception family models were the slowest.

Classification Model	Performance Metrics					
	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Specificity (%)	Training Time (s)
MobileNet	**96.05%**	**96%**	**96%**	**96%**	**98.02%**	**120.83**
InceptionV3	89.34%	89.33%	89.33%	89.33%	94.67%	131.87
ResNet50	95.04%	95%	95%	95.33%	97.52%	126.98
EfficientNetB0	94.92%	95%	95%	95%	97.46%	125.33
InceptionResNetV2	89.59%	90,33%	89.67%	89.67%	94.78%	142.89
VGG16	95.29%	95.33%	95.33%	95.66%	97.64%	124.64

Table 3: *Accuracy (Acc), Precision (Pre), Recall (Rec), F1-Score (F1-S), and Specificity (Spe) results for 3-class classification (COVID-19 vs. Normal vs. Viral Pneumonia) of 6 pre-trained model*

5. Conclusion and Discussion

As seen from the results, the MobileNet performances are significantly higher than the other models. MobileNet is a small, low-latency, low-power model parameterized to meet the resource constraints of various use cases. It has fewer parameters compared to the other models, which allow it to prevent overfitting [27] and to be used in real-time applications, so it was able to produce good results in a short time and less number of epochs. Unlike the other models, architectures are complex and need more time to be treated, such as ResNet50, InceptionV3, and InceptionResNetV2. As we can see in the training curve of InceptionV3 and InceptionResNetV2, their accuracy is getting increased, and it can be inferred that the accuracy will certainly be enhanced if the training is runned for more epochs. MobileNet is more accurate than the original Inception architecture (GoogleNet) while being smaller and more than 2.5 times less computation [27]. VGG16 is holding second place, and it is a simple and deep architecture with very small (3×3) convolution filters that manage to decrease the number of parameters [20] and the complexity of the network then the training time. It is nearly as accurate as MobileNet while being 32 times deeper and 27 times more compute-intensive [27]. EfficientNets is a family of eight models which achieve much better accuracy and efficiency than previous ConvNets. The simplest one, EfficientNetB0, gives already satisfying results thanks to the compound scaling method [24].

GENERAL CONCLUSION

This project is an end-of-study project to obtain the master's degree. The main objective of this project is to compare some CNN architectures in diagnosing COVID-19 using chest X-ray images. Adjusting the model and getting good results can take days, weeks, or even months. So, throughout this project, and after doing an in-depth bibliographic research on the field of machine learning in general and convolutional neural network architectures in particular, we tried to apply a set of modifications on six CNN pre-trained models in order to bring the most significant performance gains.

As the number of COVID-19 cases rises every day, it is necessary to detect every positive COVID-19 case quickly, although the resources' lack in many countries. The fast diagnosis of COVID-19 infection is essential and plays a vital role in limiting the spread of diseases to other individuals. Performance results show that 96.05% of the accuracy of the three classes (COVID-19, Normal, and Viral Pneumonia) was obtained using the MobileNet model, followed by VGG16 with 95.29%. These results are considered a real achievement despite the time constraint. It should also be noted that the selection of an algorithm depends, of course, on the problem to be solved and the storage and computation resources available.

In the light of our findings, we believe that our project would be helpful for medical diagnosis to make decisions and reduce the direct contact that happens between the patients and the specialists during the RT-PCR test. Also, the study still needs scientific testing, but with higher performance, it can pave a way towards a modern and efficient diagnosis of the COVID-19.

This project was the first step to get to know about CNN and some of its architectures. In the future works, we aim to implement the methodology used in this project with other models. Also, we will test the designed multi-classification network with Computed Tomography (CT) images for COVID-19 detection and compare the achieved results with the results obtained using chest X-ray images, and test the performance of these models in detecting other diseases. On the other hand, in every deep learning-based classification model, there are two parts inside, namely the feature extraction parts (stacking of convolutional and pooling layers) and classification parts (fully connected layers)[29]; we intend to replace fully connected layers with SVM classifier. Moreover, because of the long time deep learning networks took in general during the training process, we will try to use the architecture proposed in [28], improve it, and make it adaptable with different use cases.

BIBLIOGRAPHY

[1] J. Zhang *et al.*, « Viral Pneumonia Screening on Chest X-ray Images Using Confidence-Aware Anomaly Detection », *ArXiv200312338 Cs Eess*, déc. 2020, Consulté le: sept. 04, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/2003.12338

[2] A. Narin, C. Kaya, et Z. Pamuk, « Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks », *Pattern Anal. Appl.*, vol. 24, nº 3, p. 1207-1220, août 2021, doi: 10.1007/s10044-021-00984-y.

[3] T. Majeed, R. Rashid, D. Ali, et A. Asaad, « Covid-19 detection using CNN transfer learning from X-ray Images », p. 10.

[4] I. D. Apostolopoulos et T. A. Mpesiana, « Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks », *Phys. Eng. Sci. Med.*, vol. 43, nº 2, p. 635-640, juin 2020, doi: 10.1007/s13246-020-00865-4.

[5] A. Narin, C. Kaya, et Z. Pamuk, « Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks », *Pattern Anal. Appl.*, vol. 24, nº 3, p. 1207-1220, août 2021, doi: 10.1007/s10044-021-00984-y.

[6] A. H. Elmokadem *et al.*, « Mimickers of novel coronavirus disease 2019 (COVID-19) on chest CT: spectrum of CT and clinical features », *Insights Imaging*, vol. 12, nº 1, p. 12, déc. 2021, doi: 10.1186/s13244-020-00956-6.

[7] J. P. Cohen, P. Morrison, et L. Dao, « COVID-19 Image Data Collection », *ArXiv200311597 Cs Eess Q-Bio*, mars 2020, Consulté le: sept. 05, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/2003.11597

[8] L. Wang et A. Wong, « COVID-Net: A Tailored Deep Convolutional Neural Network Design for Detection of COVID-19 Cases from Chest X-Ray Images », *ArXiv200309871 Cs Eess*, mai 2020, Consulté le: sept. 04, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/2003.09871

[9] E. E.-D. Hemdan, M. A. Shouman, et M. E. Karar, « COVIDX-Net: A Framework of Deep Learning Classifiers to Diagnose COVID-19 in X-Ray Images », p. 14.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, et L. Fei-Fei, « ImageNet: A Large-Scale Hierarchical Image Database », p. 2.

[11] K. Sahinbas et F. O. Catak, « Transfer learning-based convolutional neural network for COVID-19 detection with X-ray images », in *Data Science for COVID-19*, Elsevier, 2021, p. 451-466. doi: 10.1016/B978-0-12-824536-1.00003-4.

[12] M. Farooq et A. Hafeez, « COVID-ResNet: A Deep Learning Framework for Screening of COVID19 from Radiographs », p. 5.

[13] A. M. Alqudah, S. Qazan, et A. Alqudah, « Automated Systems for Detection of COVID-19 Using Chest X-ray Images and Lightweight Convolutional Neural Networks ». sept. 05, 2021. doi: 10.21203/rs.3.rs-24305/v1.

[14] M. Barstugan, U. Ozkaya, et S. Ozturk, « Coronavirus (COVID-19) Classification using CT Images by Machine Learning Methods », p. 10.

[15] Y. LeCun, Y. Bengio, et G. Hinton, « Deep Learning », *Nature*, vol. 521, p. 436-44, mai 2015, doi: 10.1038/nature14539.

[16] A. Krizhevsky, I. Sutskever, et G. E. Hinton, « ImageNet Classification with Deep Convolutional Neural Networks », in *Advances in Neural Information Processing Systems*, 2012, vol. 25. Consulté le: sept. 19, 2021. [En ligne]. Disponible sur: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[17] « Common architectures in convolutional neural networks. », *Jeremy Jordan*, avr. 20, 2018. https://www.jeremyjordan.me/convnet-architectures/ (consulté le sept. 19, 2021).

[18] *Keras: Deep Learning for humans*. Keras, 2021. Consulté le: sept. 19, 2021. [En ligne].

Disponible sur: https://github.com/keras-team/keras
[19] K. He, X. Zhang, S. Ren, et J. Sun, « Deep Residual Learning for Image Recognition », in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, juin 2016, p. 770-778. doi: 10.1109/CVPR.2016.90.
[20] K. Simonyan et A. Zisserman, « Very Deep Convolutional Networks for Large-Scale Image Recognition », *ArXiv14091556 Cs*, avr. 2015, Consulté le: sept. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1409.1556
[21] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, et Z. Wojna, « Rethinking the Inception Architecture for Computer Vision », *ArXiv151200567 Cs*, déc. 2015, Consulté le: sept. 03, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1512.00567
[22] « A Guide to ResNet, Inception v3, and SqueezeNet », *Paperspace Blog*, juin 05, 2020. https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/ (consulté le sept. 06, 2021).
[23] C. Szegedy *et al.*, « Going Deeper with Convolutions », *ArXiv14094842 Cs*, sept. 2014, Consulté le: sept. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1409.4842
[24] M. Tan et Q. V. Le, « EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks », *ArXiv190511946 Cs Stat*, sept. 2020, Consulté le: sept. 06, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1905.11946
[25] « EfficientNet: Summary and Implementation », *HackMD*. https://hackmd.io/@bouteille/HkH1jUArI (consulté le sept. 06, 2021).
[26] « COVID-19 Radiography Database ». https://kaggle.com/tawsifurrahman/covid19-radiography-database (consulté le sept. 07, 2021).
[27] A. G. Howard *et al.*, « MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications », *ArXiv170404861 Cs*, avr. 2017, Consulté le: sept. 18, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1704.04861
[28] A. Krizhevsky, « One weird trick for parallelizing convolutional neural networks », *ArXiv14045997 Cs*, avr. 2014, Consulté le: sept. 19, 2021. [En ligne]. Disponible sur: http://arxiv.org/abs/1404.5997
[29] W. Zhu, B. Braun, L. H. Chiang, et J. A. Romagnoli, « Investigation of transfer learning for image classification and impact on training sample size », *Chemometrics and Intelligent Laboratory Systems*, vol. 211, p. 104269, avr. 2021, doi: 10.1016/j.chemolab.2021.104269.