EAI Endorsed Transactions
on Industrial Networks and Intelligent Systems
Research Article EAI.EU

# Automated evaluation of Tuberculosis using Deep Neural Networks

Truong-Minh Le[1], Bao-Thien Nguyen-Tat[2,*], Vuong M. Ngo[3]

[1]Faculty of Computer Science, University of Information Technology, Vietnam
[2]Faculty of Information Technology, Vietnam Aviation Academy, Vietnam
[3]Ho Chi Minh City Open University, Vietnam

## Abstract

INTRODUCTION: Tuberculosis (TB) is a chronic, progressive infection that often has a latent period after the initial infection period. Early awareness from those period to have better prevention steps becomes an indispensable part for patients who want to lengthen their lives. Hence, applying cutting-edge technologies to support the medical business domain plays a key role in improving speed and accuracy in methods of diagnosis. Deep Neural Network-based technique (DNN) is one of such methods which offers positive results by leveraging the advantages of analyzing deeply the data, especially image data format via tons of deep layers of the neural networks. Our study was wrapped up by objectively assessing the performance of modern Deep Neural Network approaches and suggesting a model offering good results in terms of the selected metrics as defined later. In order to achieve optimized results, the chosen model must adapt well to the datasets but require less hardware and computational resources.

OBJECTIVES: Our objective is to pick up and train a Deep Neural Network architecture which is highly trusted and flexibly fitted and applied to various datasets with minimum configurations. This will be used to produce good predictions based on the input data which are Chest X-ray images retrieved from the published datasets.

METHODS: We have been approaching this problem by using the recognized datasets which have already been published before, then resizing them to the consistent input data for training purposes. In terms of Deep Neural Networks, we picked up VGG16 as the baseline network architecture, then use other ones which are state-of-the-art networks for comparison purposes. After all, we recommend the neural network architecture offering the most positive results based on accuracy and recall measurements. So that, this network architecture will show flexibility when fitting into diverse datasets representing different areas in the world that suffered from Tuberculosis before.

RESULTS: After conducting the experiments, we observed that the Mobilenet model produced great results based on the predefined metrics for most of the proposed datasets. It shows the versatility which is applicable to all CXR datasets, especially for the Tuberculosis ones.

CONCLUSION: Tuberculosis is still one of the most dangerous illnesses in the world that needs vital methods to prevent and detect soon so that patients are able to keep their lives longer. After this research, we are constantly improving the current accuracy of the models and applying the current results of this research for later problems such as detecting the Tuberculosis areas in real-time and supporting doctors to make decisions based on the current status of patients.

Received on 03 March 2022; accepted on 11 April 2022; published on 14 April 2022

**Keywords:** Automated Evaluation, Convolutional Neural Networks, Classification, Chest X-ray, Deep Neural Networks, Tuberculosis



doi:10.4108/eetinis.v8i30.478

## 1. Introduction

Tuberculosis is now one of the most lethal diseases. According to the latest scientific report published by WHO [22], in 2020, the number of deaths officially attributed to Tuberculosis (1.3 million) was nearly double that of HIV/AIDS (0.68 million). It does spread from person to person without concerning the borders. Tuberculosis has ambiguous clinical symptoms such as chest pain, dyspnea, sweating, hemoptysis, easily making patients confused and underrate the symptoms at the initial stage. They may try to stay at home to suffer from the symptoms until over their limit before going to health facilities. Besides, infectious bacteria from Tuberculosis can also be spread to the community from the infected people through respiratory activities such as coughing, spitting and sneezing. Detecting Tuberculosis earlier plays a vital role in saving time and money for patients as well as avoiding out of control spreading. Manually diagnosing and identifying via Chest X-ray images requires a thorough base of medical knowledge and image processing techniques to understand the context and many time on medical consultation.

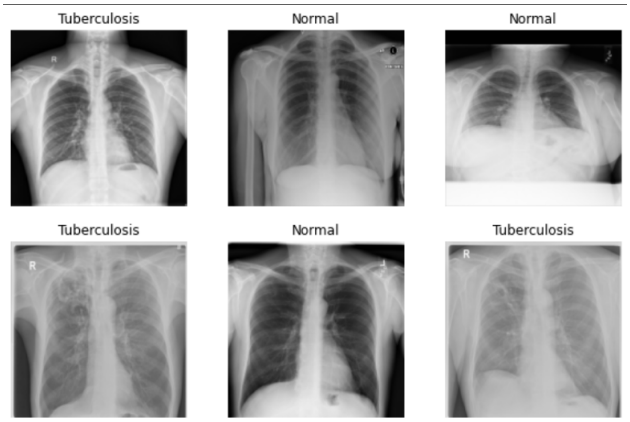


**Figure 1.** Chest X-ray dataset with the labels attached

From the current situation, applying advanced technologies such as Deep Learning with the support of medical health images processing systems to assist doctors in recognizing Tuberculosis and tracking the symptoms and diseases over time is becoming popular. It is replacing traditional techniques since the abilities to help computer scientists develop complex software architectures and build stable models with high accuracy for residents in different areas based on accumulating typical data processes.

In our study, we analyzed the performance of modern deep learning models, namely VGG16. EfficientNetB7, MobileNetV3, DenseNet121, RegNet for Chest X-ray analysis with two main labels: TUBERCULOSIS (those who got Tuberculosis) and NORMAL (normal people). Apart from the baseline model (VGG16), other neural network architectures are newly published. They can help us build good models with high results on many datasets. We comparatively cross-assessed the accuracy of these models on each dataset. Details of how we build models, as well as the detailed dataset are stored in our public repository [1]. Our paper contains three contributions as below:

- We made a specific survey and selected the five latest models to use for the Tuberculosis evaluation problem. Also, we summarized four well-known datasets in the Tuberculosis domain and adjusted them to well-adapt the classification problem. In the upcoming projects, everyone can use these datasets for their specific problem with little updates based on their needs.
- Based on the concluded results, we proposed a model with highly trusted accuracy and recall. In addition, as compared to others, it provides a high level of stability. We can then use it to apply for local residents in a variety of different regions.
- A visually detailed comparison between the chosen models has been conducted to achieve in-depth knowledge so that other researchers can gain more deep insights about this problem such as what has been done, what we will do in the upcoming projects, what is still left. From that, we can have well-prepared topics for the next stage.

The paper will be arranged as follows, section 2 reviews the related works of other researchers. Section 3 provides background theories and methods that we utilized in our research. Section 4 shows the practical outcomes of our study. Section 5 is the further discussion of how we develop these topics further in new problems and the conclusion of the paper.

## 2. Related Works

Stephan Jaeger et al. [8] proposed two standard datasets that should be used for the Tuberculosis classification problem, including the Shenzhen Chest X-ray dataset and Montgomery Chest X-ray dataset. We leverage the final results, two highly trusted datasets, namely Montgomery and Shenzhen Chest X-ray datasets as the fundamental data resources to our research.

Paras Lakhani and Baskaran Sundaram [13] applied Deep Learning to an evaluation of the contemporary algorithms' efficiency on four deidentified HIPAA-compliant datasets, including AlexNet [12] and

*Corresponding author. Email: thienntb@vaa.edu.vn

[1] https://github.com/letruongminhuit/tuberculosis-dnn

GoogLeNet [20]. The experiments have acquired the recall is 97.3%. Our research has a different way to evaluate the result from this.

Sonaal Kant and Muktabh Mayank Srivastava [9] used a deep neural network architecture to analyze if a patient has infected Tuberculosis via microscopy images of sputum. They also suggested a 5-layered Neural Network architecture on *dataset 3* from ZiehlNeelsen Sputum smear Microscopy image DataBase (ZNSM-iDB) - Mohammad Imran Shah et al. [17]. We made use of a variety of dataset resources and dataset targets (frontal CXR images compared to the microscopy images of sputum). In terms of the actual outcomes, the authors received the greatest F1-Score for their best performance which was 74.79 per cent.

Tawansongsang Karnkawinpong et al. [11] reused the architecture of the VGG-16 neural network model combined with affine transformation. They combine three datasets, including Shenzhen, Montgomery and Thai into one larger dataset. Then, they utilize three pre-trained neural network architectures, namely AlexNet, VGG-16, and CapsNet correspondingly to diagnose Tuberculosis infection. Throughout the research, applying affine test with -10 to 10 rotation with VGG-16 obtained the highest accuracy which is 90.79%. Compared to our current accuracy score, as we use the more advanced algorithms our metrics surpass the accuracy score of [11] with 98.35 per cent.

R. Dinesh Jackson Samuel and B. Rajesh Kanna [4] utilized the model which uses the pre-trained model of InceptionV3 paired with SVM to classify the data. To complete their research, the authors used the publicly available dataset [17]. This model has a high accuracy score of 95.05%, which is reliable to assist medical practitioners in making decisions if patients have got Tuberculosis. As per the earlier mentioned research [9], this research has a different dataset target from ours. We think using this dataset to analyze in the future, extending our research scope is also a good development.

T Karnkawinpong and Y Limpiyakorn [10] conducted experiments to categorize CXR images for Tuberculosis on two datasets - one acquired from the National Library of Medicine and one from private Thai datasets. AlexNet, VGG-16 and CapsNet were three neural network architectures used. They used the augmentation technique with shuffle sampling to help overcome overfitting, which is an advantage. The final findings revealed that the shuffle sampling approach used in the VGG16 neural network architecture had the best accuracy score (94.56 per cent). However, this is still inferior to our accuracy when using MobileNetV3 on TB Chest X-rays (98.35 per cent). The study's flaw is that it used the VGG16, a very rudimentary neural network architecture that produces lower results than state-of-the-art techniques like MobileNetV3 and DenseNet121.

[3] presented a study to evaluate the different computational performance and classification results between four convolutional neural network models, namely VGG-16, VGG-19, ResNet50 and GoogLenet. They utilized the aforementioned approaches on two datasets, Montgomery and Shenzhen, as well as our results, too. Data Augmentation was also used as a preprocessing step prior to training and classification tasks to increase the size of these datasets. When it comes to the Montgomery dataset, their proposed models received a 77.14 per cent accuracy score at the conclusion of the study. This study had a lower accuracy score than ours, with a score of 77.81 per cent.

Tawsifur Rahman et al. [16] aggregated pieces of small data into one reliable and bigger dataset with 7000 images in total. Also, they compared three methods: segmentation of X-ray images using two different U-net models, classification using X-ray images, and segmented lung images. In the second experiment using the ChexNet neural network architecture without segmentation, their accuracy and F1-score have both achieved 97.07%. We also use the dataset of this research and apply our techniques to understand thoroughly the dataset and the final results, including accuracy is higher (98.35%) and F1-score (98.32%).

In the previous year, Luyao Ma et al. [14] was to use CT images to analyze if the patients are normal. With regard to data, they used the dataset of 846 patients collected from a large hospital and then U-Net deep learning algorithm was used to analyze. This approach achieved a 96.8% of accuracy score. One advantage of this research is applying the segmentation technique to highlight accurately the field that has a signal of Tuberculosis. We are going to apply this technique in the future to improve the classification efficiency when applying in CXR images.

Linh T. Duong et al [5] made use of the vision transformer architecture and transfer learning to deeply analyze the Tuberculosis disease on four specific datasets, namely Montgomery CXR dataset, Shenzhen dataset, Belarus dataset, a COVID-19 dataset, and the adoption of additional imahes from various sources, including RSNA Pneumonia Detection Challenge dataset, COVID-19 Radiography DB CXR images by merging all of them to each other to have the final dataset, including Montgomery County CXR dataset and Shenzhen dataset which have been used in our study. After that, they have set up 14 environment configurations and verified the final results. As a result, the tenth environment configuration achieved the highest accuracy with 97.72%. When comparing the outcomes obtained in [5] to ours, we can see that our result when applying

MobileNet-V3 on the Tuberculosis (TB) Chest X-ray returns the higher accuracy score of 98.35%.

## 3. Proposed Methods

### 3.1. Classification Problems

Classification problems, as depicted in Figure 2, are predictive modelling problems in which a class label is predicted for given input data. To begin, data is typically divided into two parts: training and testing. The training partition will then be divided into two sub-partitions, including the training set and dev set. We will discuss the role of the dev set and how we will use it in future experiments in our study. Then, using a classification model, each data point in a dataset is assigned to one class and label. The model is built using the labeled datasets. As a result, the main task of the classification problem is to find a standard model after the fitting process so that new data which is inputted into the model can be classified into the correct class. In this paper, building model means finding function $f$ to map data point x to y $\in Y : y = f(x)$.

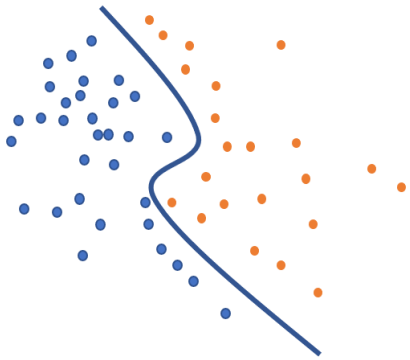

**Figure 2.** Visualization of classification problem

There are four factors needed to consider when we want to resolve a classification problem:

- **Dataset**: this is the core component as our ultimate goal is to find the model that fits the data well and determine the relationship between the data points and its ground truth as well as the relationship between the training set and test set. We will then proceed to perform the classification tasks or any other academic tasks in Machine learning or Deep Learning domains.
- **Model**: to effectively build a strongly stable model, we must first understand how to process the data and combine the layers between deep learning layers in order to achieve good results.
- **Loss Function**: this is the solution for minimizing the gap between the ground truth and predicted labels. This value must be clearly understood in order for us to fully comprehend the model.
- **Algorithms to optimize the loss function**: this is when we apply some common algorithms such as Gradient Descent to minimize the loss function. This algorithm will try as much as possible to update the loss function after each epoch until the loss function cannot be updated anymore. After all, this is the most optimized model that we are finding and further used in the test set via the evaluation stage.

There are many typical types of classification problems, including binary classification, multi-class classification. We mainly apply binary classification to assign an image object to one of two different classes based on a query that if the data point has the same feature as the classifier.

### 3.2. Deep Neural Network Approaches

When it comes to applying new technologies to analyzing large-scale data and processing the given data to solve problems on a regular basis, everyone has an intention to choose between traditional machine learning approaches and deep neural network-based approaches, each with their own set of pros and cons.

While machine learning approaches have some drawbacks, such as requiring too much time to interpret the data possessed and pre-processing techniques to clean the data before implementing more machine learning-based techniques to propose the results, deep learning demonstrates its strength when applying dynamically deep neural network architectures to various types of data formats, requiring less effort in data cleansing phases, and requiring less human intervention. These benefits outperform traditional machine learning ones in most cases. As a result, systematically analyzing current problems, applying the most appropriate models, and identifying ways to improve performance is a good approach, particularly for Tuberculosis detection problems.

Figure 3 depicts our proposed approach, which is an end-to-end Deep Learning approach that employs an efficient combination of our neural networks. Following the input of an image into the networks, we will use the Data Augmentation technique to increase the size of two datasets, India and Montgomery CXR as preprocessing steps. We fed the training images into five neural network models as part of the Data Augmentation step for the India and Montgomery datasets. This is where the Representation Learning approach comes into play, which allows networks to exploit data features and automatically minimize the loss function. Finally, each input image is classified

into two classes after being returned from the Fully Connected Layer: Tuberculosis and Normal.

### 3.3. Deep Learning Architectures

When it comes to classification problems, Deep Learning outperforms traditional machine learning-based techniques, particularly the intelligent systems with a large amount of data a large number of dimensions, such as speech recognition and computer vision. Deep Learning has also integrated optimized data processing techniques such as pre-processing and feature selection. Furthermore, we must thoroughly understand how to apply effectively combinations of Deep Learning layers and parameters calibration. Of all Deep Learning network architectures, Convolutional Neural Network architecture (CNN) is suitable for medical health image processing and bioinformatics as it offers high-performance capability and reduces the learning parameters when compared to basic neural networks.

**a. VGG16.** We used VGG16 [19] as the baseline model and compared it to other well-known published models. VGG16 is a CNN-based neural network architecture first described in the paper by K. Simonyan and A. Zisserman of the University of Oxford in the paper [19]. This proposed model achieves 92.7% top-5 test accuracy in ImageNet, a dataset of over 14 million images belonging to 1000 classes in total. The input image is 224x224 with three channels by default. In terms of VGG16, the significant point that we can see is improving the model's accuracy by using a deeper neural network architecture. It does, however, retain AlexNet's features. To reduce parameter numbers, VGG16 uses a smaller filter with a 3x3 size rather than 11x11 or 5x5 as AlexNet.

Features can be extracted more effectively when compared to the previous models such as AlexNet [18], and the output is returned at the final layer, which is the fully-connected layer used to predict the output label. VGG16 is subdivided into three different parts: Convolution, Pooling, and Fully Connected layers. It begins with two Convolution layers, followed by a Pooling layer, then another two Convolution layers, followed by a Pooling layer, followed by a repetition of three Convolution layers, followed by a Pooling layer, and finally three Fully Connected layers. The detailed architecture of VGG16 can be seen in Figure 4.

On the other hand, two disadvantages have been proved clearly with VGG16 as below:

- The training phase takes far too long to complete, causing the other stages to fall behind schedule.
- The neural network architecture has an excessive number of weights.

VGG16 is too large in size due to its depth and number of fully connected nodes at the later layers, making deployment and integration into the applications complicated. Despite the fact that it can be leveraged to solve a wide range of deep learning challenges, more optimal network topologies are often favored.

**b. EfficientNetB7.** This model belongs to the EfficientNets algorithm family and was first introduced in 2019 in the paper [21] by Mingxing Tan and Quoc V. Le in May 2019. EfficientNets rely on AutoML and compound scaling to achieve superior performance without affecting badly resource efficiency. The AutoML mobile framework has helped develop a mobile-size baseline framework. Currently, we made use of the EfficientNetB7 neural network architecture to implement the feature extraction process. These researchers proposed a model scaling method that carefully balances the depth, width, and multi-dimension sizes of the network structure, strengthening the computational efficiency. The detailed architecture of the EfficientNetB7 neural network architecture is shown in Figure 5.

**c. MobileNetV3.** The advancement of Computer Vision encourages the development of numerous deep learning architectures with various architectures to ameliorate computational performance. However, due to computational constraints, not all of them can be used in all devices. If we want to develop AI applications on various devices such as mobile and IoT, we will need to thoroughly understand how these devices compromise their hardware resources in order to choose the model for them. One such model is the MobileNetV3 [6].

In usual CNN-based neural networks, depth is one of the main reasons increasing strongly the number of parameters of models. So, Depthwise Separable Convolution will figure out how to eliminate the reliance on depth when performing convolution operation while still producing an output shape of the same size as a standard convolution. Each channel will use a unique filter and will not use the shared parameters, allowing the model to improve the computational performance and reduce the number of parameters required. The detailed architecture of the MobileNetV3 neural network architecture is depicted in Figure 6.

**d. DenseNet121.** The DenseNets algorithm family [7] is one of the most powerful neural nets, achieving state-of-the-art performance on a variety of datasets. When the model architecture is too deep, new issues emerge alongside the CNN-based methods. The reasons hidden behind the issues caused by the data path from the first layer (input) through hidden layers to the classification layer (output) becomes so substantial that they may vanish before reaching the other side. DenseNets maximize network capacity by reusing features from

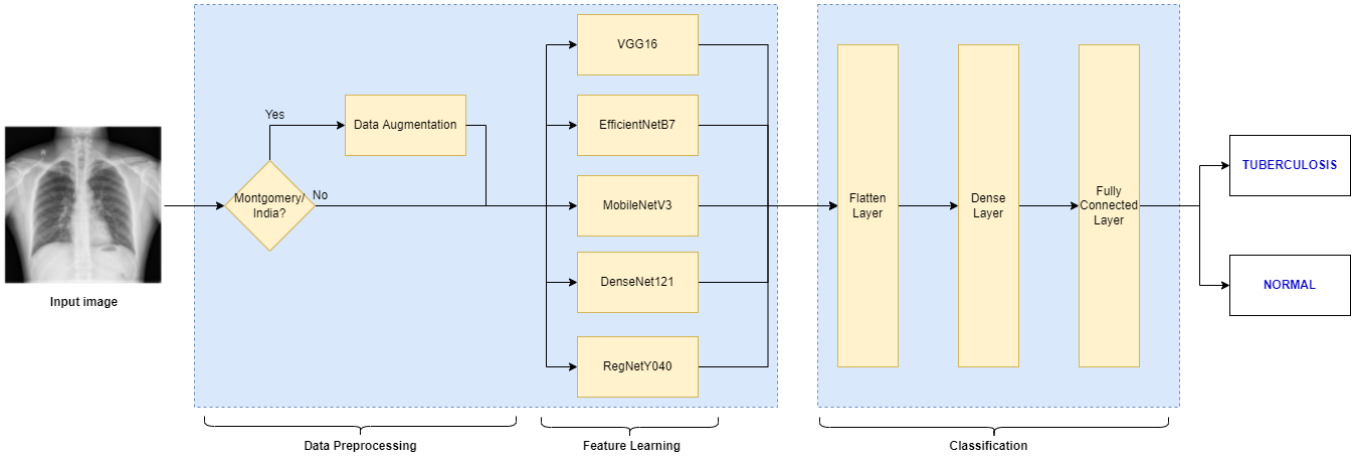


**Figure 3.** End-to-end approach of Chest X-ray classification

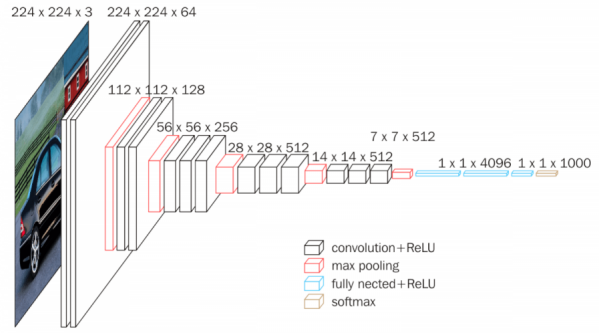


**Figure 4.** Detailed architecture of VGG16 neural network architecture

extremely deep or wide neural network architectures. The feature maps of all preceding layers are treated as independent inputs for each layer, whereas their own feature maps are used as inputs to the next ones. On the channel dimension, DenseNet concatenates inputs and outputs. They create short paths from the early layers to the later ones, as mentioned in the paper [7].

We use DenseNet121, a 121-layer neural network architecture, to solve the Tuberculosis classification problem. It is a DenseNet that has been pre-trained on the ImageNet dataset. The detailed architecture of the DenseNet121 neural network architecture is depicted in Figure 7.

**e. RegNet.** Convolutional Neural Network topologies have traditionally been optimized for a single purpose. For instance, when the ResNet model family was first released, it was designed for maximum accuracy on ImageNet. EfficientNet was created with visual recognition tasks in mind. When it comes to the RegNet [15], they set out to investigate and design a network architecture that was extremely adaptable. One that can be converted to be highly efficient or run on mobile devices, while also being highly accurate when being optimized for classification performance. The width and depth of the network architecture are versatile and flexibly determined by selecting the proper parameters in a quantized linear function.

The parameters are configured differently to produce various RegNets with different purposes:

- A RegNet that has been designed for mobile use
- An efficient RegNet
- A highly precise RegNet

RegNet has a fundamental part called a network design space, being made up of multiple parameters that define a space of possible model architectures, not just different model architectures. Inside the design spaces, there are three fundamental blocks: stem, body, and head. Concerning the stem block, it will take the input images and extract the features within them using the 3x3 convolutional layer, which has a stride value of two. Following that, the body layer will be in charge of carrying out a slew of computational steps as well as handling the previously defined features extracted from the stem block. Finally, the head block will take the implemented computation as input data and process it to determine which outputs belong to which classes. Following the traversal of design space's body, numerous procedures are used to reduce the size of height and width channels while increasing the size of the depth channel. This architecture is visually appealing, but it necessitates a large number of parameters during the training process, making the training phase a strain on a model. In the later section, we'll go through the details of the parameters (both trainable and non-trainable). Such parameters include the network's width, depth, groups, and so on. The authors also defined AnyNet, a space of all possible models, before arriving at the final RegNet design space. AnyNet takes responsibility for investigating the

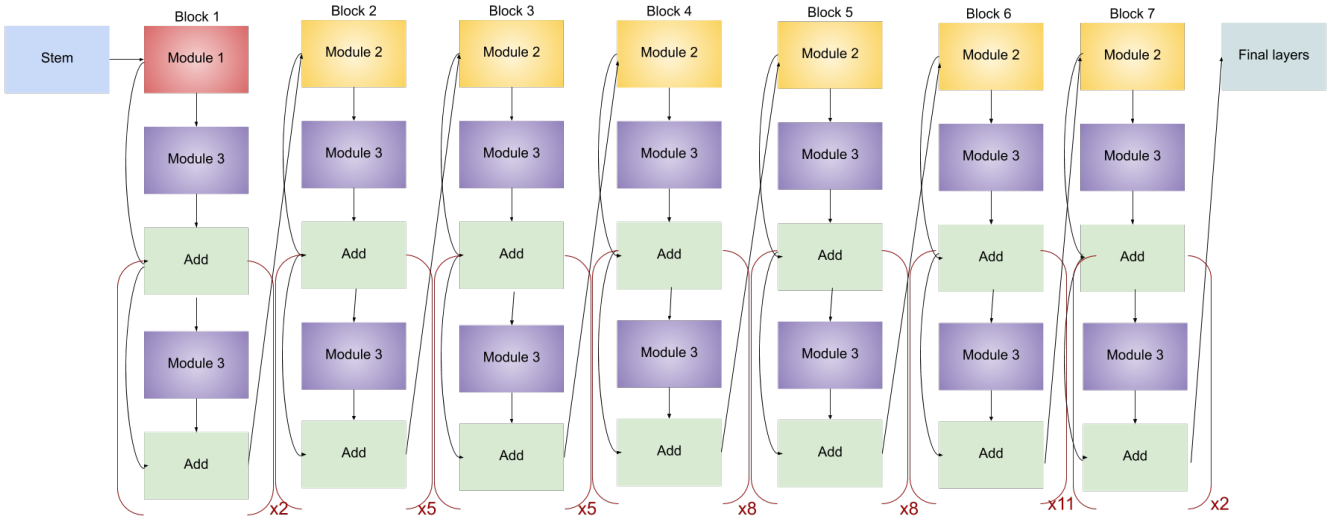


**Figure 5.** Detailed architecture of EfficientNetB7 neural network architecture

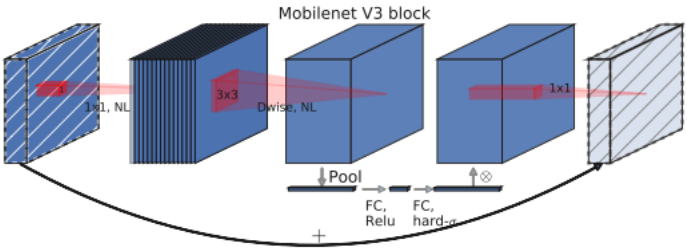


**Figure 6.** Detailed architecture of MobileNetV3 neural network architecture

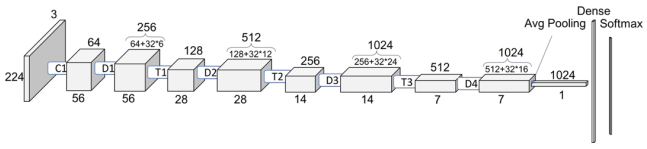


**Figure 7.** Detailed architecture of DenseNet-121 architecture neural network architecture

effective neural network structure. Based on various parameter combinations, this space generates a diverse set of models. Using a standardized training procedure, all of these models are trained and tested on the ImageNet dataset (epochs, optimizer, weight decay, learning rate scheduler). Figure 8 depicts the general structure of the RegNet model and how it performs numerous calculations for prediction steps at the end.

They generate progressively simpler versions of the initial AnyNet design space from this AnyNet space by analyzing which parameters are responsible for the high performance of the best models in the AnyNet design space. They are essentially experimenting with

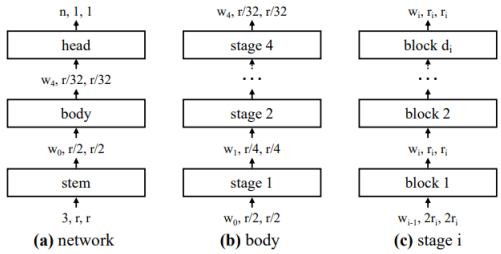


**Figure 8.** Detailed architecture of RegNet architecture neural network architecture

the relative importance of various parameters in order to narrow the design space to only the best models. After all, they acquire the optimized RegNet design space, containing only great models as well as the quantized linear function required to define the models. In our study, we leveraged the RegNetY040 which has been integrated into Tensorflow [1].

## 3.4. Early Stopping

We leverage **Early Stopping** as a way to calibrate the model to get over difficulty of too little training or too much training. If there is too little training, models will not be able to learn all datasets accurately. In case there is too much training, model will be in the overfitting status, which lead to low performance on test set.

In this study, we use the object Early Stopping method [2] and use loss value as the monitor value. After

[1] https://www.tensorflow.org/api_docs/python/tf/keras/applications/regnet/RegNetY040
[2] https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping

5 training epochs, if the loss value does not decrease, the training phases will stop without performing any epoch later.

### 3.5. Neural Network layers

These properties are the fundamental functional building blocks of neural networks. Each layer comprises a tensor for the in and out computational method, as well as some states. After loading the pre-trained models, we fine-tune them by adding layers to the pre-trained models' output, such as Flatten and Fully Connected layers.

Flatten layer plays the role of flattening the input but has no effect on the batch size. It does not learn any characteristics from the models.

We also employ two Fully Connected layers, with the activation function `sigmoid` serving as the classification at the end. The detailed formula of the sigmoid function is shown in the equation 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

## 4. Experimental Results

### 4.1. Datasets

All datasets are collected from highly trusted datasets which are published before. With two datasets - Montgomery County CXR Set and India CXR Set, as it has few images inside which may lead to missing generality, we apply data augmentation to not only increase the size of the datasets but also introduce variability in the datasets, without actually collecting new data. The neural network architecture treats these images as distinct images anyway. Also, data augmentation helps reduce overfitting effectively.

**a. Tuberculosis (TB) Chest X-ray Database.** This dataset [16] consists of two folders containing training images and test images, as well as an excel file containing images information with two labels: Normal (3500 images) and Tuberculosis (3500 images). We divide the dataset into two folders, train and test, so that we can easily use them for phases of training, validation and testing. Tuberculosis and Normal are the corresponding sub-folders in each folder.

We divide the dataset with the specific percentage as below:

- The training set comprises 64% with 2240 images belonging to the Normal class and 2240 images belonging to the Tuberculosis class.
- The validation set comprises 16% with 560 images belonging to the Normal class and 560 images belonging to the Tuberculosis class.
- The test set comprises 20% with 700 images belonging to the Normal class and 700 images belonging to the Tuberculosis class.

**Table 1.** Details about how Tuberculosis (TB) Chest x-ray database is splitted

| | Normal | Tuberculosis |
|---|---|---|
| **Training set** | 2240 | 2240 |
| **Validation set** | 560 | 560 |
| **Test set** | 700 | 700 |

**b. Shenzhen Chest X-ray dataset.** Shenzhen Chest X-ray dataset [8] has been collected from outpatient clinics at hospitals which are Shenzhen No.3 People's Hospital, Guangdong Medical College, Shenzhen, China.

This dataset includes 662 chest x-ray images and a CSV file with two properties: the gender and age of the patients of each image. We modify the dataset by splitting it into three distinct sets – Train set, Validation set and Test set. We also set aside 20 per cent of the dataset for validation purpose. In each set, we also create two sub-folders that correspond to its labels – Tuberculosis and Normal.

**Table 2.** Details about how Shenzhen dataset is splitted

| | Normal | Tuberculosis |
|---|---|---|
| **Training set** | 209 | 217 |
| **Validation set** | 52 | 54 |
| **Test set** | 65 | 65 |

**c. Montgomery Chest X-ray dataset.** Montgomery Chest X-ray dataset [8] has been collected in collaboration with the Department of Health and Human Services, Montgomery County, Maryland, USA. This dataset contains 138 frontal chest x-ray in which there are 80 images having Normal class and 58 images having Tuberculosis class. After reserving 28 images for the Test set and 22 images for Validation set, we allocate the rest of the images to the Training set and start applying Data Augmentation. In particular, we use CLoDSA [1] created by Casado-García et al. The general idea of the method is first to define a list of data augmentation techniques which will be used and add them to the augmentor object. There will be a process that receives the augmentor object as input and returns the list of augmented images. We generate more images to support the Training phase with these techniques: rotation, cropping, shifting, and use the original.

The details of each technique is as below:

- **Rotation**: We do rotate technique the image randomly 5, 10, 15 degree for each image in the dataset.

- **Cropping**: We do cropping technique with percentage 0.9 and start from top-left.
- **Shifting**: We do shifting technique along with X-axis and Y-axis with 10 and 10 respectively.
- **Original**: We also keep the original image as one resource of images because it contains the standard feature of the dataset.

**Table 3.** Details about how Montgomery dataset is splitted and augmented

| | Normal | Tuberculosis |
|---|---|---|
| **Training set** | 385 | 231 |
| **Validation set** | 11 | 11 |
| **Test set** | 14 | 14 |

**d. India Chest X-ray dataset.** India Chest X-ray dataset [2] CXR digital image is taken from X-ray machine available at the National Institute of Tuberculosis and Respiratory Diseases, New Delhi. Dataset is available freely at dataset.

This dataset contains a total of 155 chest x-ray images. For this dataset, we also use the aforementioned tool (CLoDSA) to extend the dataset's size in order to make it more general. We divide the original dataset into three particular parts: training, dev and test sets. We used the same techniques as in the Montgomery dataset for the final one. The structure is similar to the three previously mentioned datasets, as shown below:

**Table 4.** Details about how India dataset is splitted

| | Normal | Tuberculosis |
|---|---|---|
| **Training set** | 287 | 294 |
| **Validation set** | 10 | 10 |
| **Test set** | 26 | 26 |

## 4.2. Data Preparation

Training process using Convolutional Neural Network-based techniques (CNN) require a lot of labeled images as data for necessary phases. Also, input images are required to have a consistent resolution in association with the development of deep learning architectures. All properties belonging to the images should also be coherent with each other for further consideration. Hence, pre-processing data is a vital step for any machine learning system as well as algorithm.

For each dataset described above, we sequentially read image data from training set, validation set and test set to colab via method *image_dataset_from_directory* [2]. For the input images, it is divided into smaller batches with batch_size is 128. We sequentially train on each batch and compute the final value which is mean of all batches.

We resize the image into the resolution 224x224 to fit the model. Also, we use the color mode **RGB** for the images. Besides, the above datasets, especially three datasets which are India Chest X-ray, Montgomery Chest X-ray have quite a few images inside. However, spatial features of images - positions of organs in the body - are needed to be retained to ensure the detection and classification processes are correct. Therefore, we do not use some typical data augmentation techniques such as flipping the images, zooming in and out the images, changing the photo sizes to absolutely keep the correct positions.

Moreover, we shuffle the images in the training set and validation set to ensure each data point create an "independent" change on the model, without being biased by the same points before them. Using this method also prevents models from extracting the rules which can be easily found. This affects positively the model by enhancing the difficulty of detecting the training phase.

## 4.3. Parameters Configuration

Parameters are the coefficients of the model, and they are initialized and updated by the model. During the training processes, parameters are always calibrated in order to minimize the loss function of the model. These parameters can be estimated and learnt from data, then it is still be reused and updated back to the earlier layers. Practitioners have investigated which value of parameters are good and used for various models.

There are two types of parameters when taking Neural Network architectures into account: trainable parameters and non-trainable parameters.

- Trainable parameters: When it comes to pre-trained models, trainable parameters are the parameters that models will need to learn and calibrate on the datasets that the loss function of the model will be minimized.
- Non-trainable parameters: On the other side, it is the pre-trained parameters that have been trained previously and can now be used without the need to retrain them.

Besides, we also have the total parameters which are the sum of trainable and non-trainable parameters. We can observe the parameters that each model has after compiling the model as clearly described in Table 5.

As shown in Table 5, EfficientNet-B7 accounts for most parameters, including both trainable parameters and non-trainable parameters, in order to fully train the datasets. Hence, this consumes a lot of computational and storage resources although it is

[2]https://keras.io/api/preprocessing/image/

**Table 5.** Numbers of parameters for each model

| Model | Trainable Parameters | Non-trainable Parameters | Total Parameters |
|---|---|---|---|
| VGG16 | 3,211,521 | 14,714,688 | 17,926,209 |
| EfficientNetB7 | 16,056,577 | 64,097,687 | 80,154,264 |
| MobileNetV3 | **3,612,929** | **939,120** | **4,552,049** |
| DenseNet121 | 6,422,785 | 7,037,504 | 13,460,289 |
| RegNetY040 | 6,824,193 | 19,619,928 | 26,444,121 |

able to produce good results for some datasets. Next, the RegNetY040 and DenseNet-121 have the approximate numbers of trainable parameters with roughly 6,500,000 parameters, but the difference in non-trainable parameters leads to the difference in the total parameters of the two mentioned models. Overall, MobileNetV3 produces superior results with fewer parameters configured. It means we only need fewer computational resources but achieve better results when integrating MobileNetV3 into the applications and recommendation systems.

### 4.4. Performance Evaluation Metrics

As we are finding solution for the Tuberculosis, it means that we will try to categorize an input image to one class which is Tuberculosis or Normal. There are four values representing for 4 types of predictions:

- **True Positive** (TP): this is the number of data points in which models correctly predict with labels **Positive** and the true labels are also **Positive**.
- **True Negative** (TN): this is the number of data points in which models correctly predict with labels **Negative** and the true labels are also **Negative**.
- **False Positive** (FP): this is the number of data points in which models predict with labels **Positive** but the correct labels are **Negative**.
- **False Negative** (FN): this is the number of data points in which models predict with labels **Negative** but the correct labels are **Positive**.

For the Tuberculosis issue, classifying True Positive and False Negative cases are far more important than two classes left. When patients are not considered as Tuberculosis correctly, they will get sicker or die. Therefore, minimizing the number of predicted cases in two mentioned classes are also the goal of the models created.

We conduct an algorithm to evaluate the models based on four metrics which will be presented later as below (Accuracy, Precision, Recall, and F1-Score). We use 4 lists to store all metrics of all data points. After making prediction for a data point, we assign the newly predicted to a list until there is no data point left in the test set. Then, we evaluate difference between ground truth and the newly predicted labels. Finally, we compute mean for each metric array and return the results:

**Algorithm 1** Evaluation Model algorithm

```
procedure EvaluationModel(model)
    Initialize accuracy list
    Initialize recall list
    Initialize precision list
    Initialize F1-Score list
    for image, label in test set do
        Predictions ← Predict(model)
        Predictions ← Normalize(Predictions)
        accuracy ← accuracy(label, Predictions)
        recall ← recall(label, Predictions)
        precision ← precision(label, Predictions)
        F1 − Score ← F1 − Score(label, Predictions)
        Append accuracy to accuracy list
        Append recall to recall list
        Append precision to precision list
        Append F1-Score to F1-Score list
    end for
    accuracy ← mean(accuracylist)
    recall ← mean(recalllist)
    precision ← mean(precisionlist)
    F1 − Score ← mean(F1 − Scorelist)
end procedure
```

**Accuracy.** Accuracy is a metric to evaluate classification models. It is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

**F1-score.** When it comes to F1-score, we should understand two terms: Precision and Recall.

**Precision** refers to the fraction between True Positive and the sum of True Positive and False Positive. The specific formula of Precision score is described in the

equation (3).

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

**Recall** refers to the fraction between True Positive and the sum of True Positive and False Negative. The specific formula of Recall score is described in the equation (4).

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

When a model has Precision is equal to 1, all detected points are truly positive and there is no negative point inside. But, when a model has Recall is equal to 1, all positive points have been observed. However, this metric cannot evaluate how many negative points inside. A good model is the one has high Precision as well as Recall. To evaluate this measure, we use F1-score with the following formula:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{5}$$

F1-score has the value in (0, 1].

**Confusion Matrix.** For the Tuberculosis classification, a 2x2 table as shown in Figure 9 that summarizes how successful a classification model's predictions were; that is, the correlation between the label and the model's classification. One axis of a confusion matrix is the label that the model predicted, and the other axis is the actual label. The number two denotes how many labels or classes are available.

Predicted Labels

| True Labels | Tuberculosis | Normal |
|---|---|---|
| Tuberculosis | True Positive | False Negative |
| Normal | False Positive | True Negative |

**Figure 9.** Confusion Matrix

## 4.5. Experimental Results

Using pretrained models, we hope to determine whether a given CXR image belongs to the Tuberculosis or Normal class. As a result, five techniques are used to gain insights for furture comparisons.

After applying the fine-tuning process to four datasets, we produced the confusion matrices of all models for labels as shown in Figures 10, 11, 12, 13 sequentially.

The performance of five neural network architectures against four datasets is shown in Table 6. The model producing impressive results when compared to others is highlighted in bold. Accuracy and F1-score are used to evaluate performance with objective results.

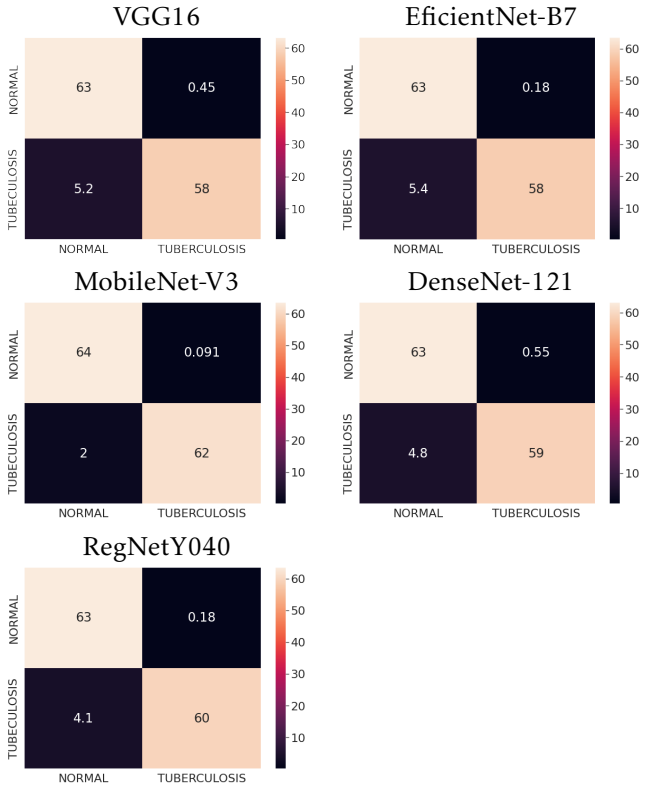


**Figure 10.** Confusion matrices for models applying to Tuberculosis (TB) Chest X-ray database – MobileNet-V3 shows better results with True Positive and False Negative images detected are quite high – 64 and 62 correspondingly

The findings are expressed as a percentage. As a result, we can see that MobileNetV3 delivers the best results in two datasets which are Tuberculosis (TB) Chest X-ray (Accuracy=98.35%, F1-score=98.32%) and Montgomery Chest X-ray (Accuracy=77.81%, F1-score=78.92%). It also produces good performance on two datasets left, including the Shenzhen Chest X-ray dataset (Accuracy=67.19%, F1-score=74.86%), and the India Chest X-ray dataset (Accuracy=86.25%, F1-score=83.75%), with the second-best results. These metrics are roughly comparable to the best.

According to the results of the preceding experiments, MobileNetV3 has the highest level of stability in terms of performance metrics. To be more specific, MobileNets, particularly version 3, uses hyperparameters effectively to trade off latency and accuracy. Based on the perks mentioned, this should be considered when applying for current applications, especially mobile devices.

Differences in human body parts, specifically the thoracic skeleton, between regions of the world result in differences in the spatial characteristic arrangement of X-ray images. In some studies, such as [5], authors combined all datasets into a single larger dataset before

**Table 6.** Performance evaluation between models on the selected datasets

| Feature Extraction Models | Tuberculosis (TB) Chest X-ray | | Shenzhen Chest X-ray | | Montgomery Chest X-ray | | India Chest X-ray | |
|---|---|---|---|---|---|---|---|---|
| | Accuracy | F1-score | Accuracy | F1-score | Accuracy | F1-score | Accuracy | F1-score |
| VGG16 | 95.56% | 95.40% | 64.38% | 72.71% | 64.38% | 68.66% | 81.25% | 80.18% |
| EfficientNetB7 | 95.65% | 95.49% | 61.25% | 67.53% | 49.69% | 65.31% | 86.88% | 88.33% |
| MobileNetV3 | **98.35%** | **98.32%** | **67.19%** | **74.86%** | **77.81%** | **78.92%** | **86.25%** | **83.75%** |
| DenseNet121 | 95.78% | 95.59% | 70.00% | 71.57% | 60.94% | 70.65% | 65.00% | 53.33% |
| RegNetY040 | 96.65% | 96.55% | 62.19% | 68.43% | 71.56% | 75.32% | 84.69% | 84.04% |

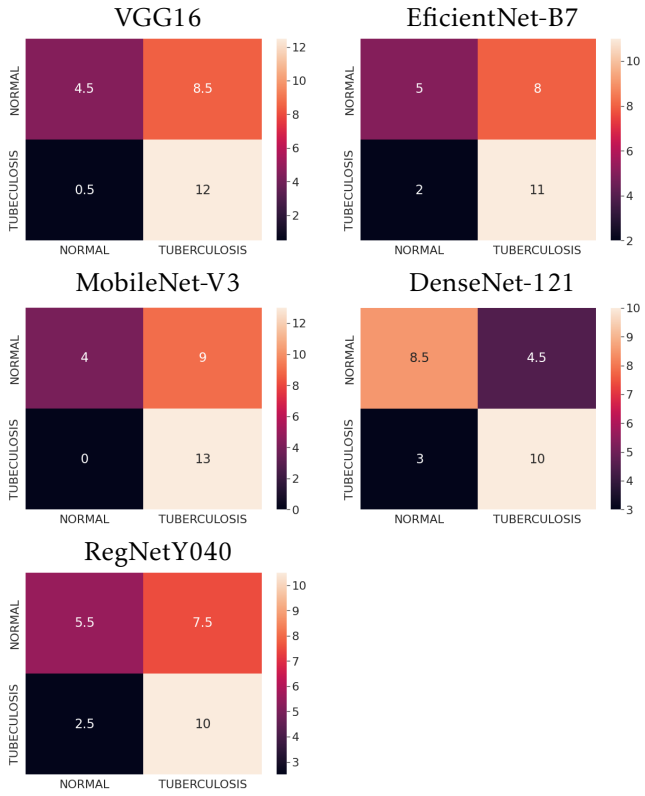


**Figure 11.** Confusion matrices for models applying to Shenzhen Chest X-ray Set - MobileNet-V3 provides the second good results with True Positive and False Negative images detected are high - 4 and 13 correspondingly

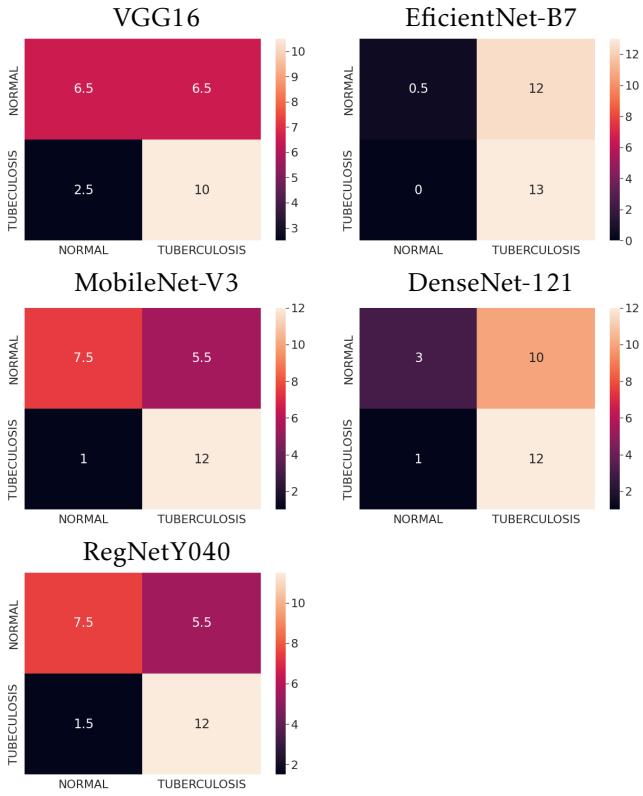


**Figure 12.** Confusion matrices for models applying to Montgomery Chest X-ray Set - MobileNetV3 provides the better results with True Positive and False Negative images detected are high - 7.5 and 12 correspondingly

training, which has a negative impact on the spatial distribution of the object. As a consequence, in our research, we continue to keep the datasets separate in order to preserve the specificity of spatial distribution. This is one of the benefits of our research so far.

VGG16 is an old neural network architecture, as far as we know from CNN-based models. All layers are added as a stack of layers, and the volume of tensors (area of feature maps multiplied by the number of features) is slowly reduced. Although it is simple to understand and apply in real-world situations, the need for too many parameters has a negative impact on inference and test stages, as well as memory limitations. As a result of the aforementioned drawbacks, we use it as a baseline algorithm to primarily benchmark other more modern algorithms.

## 5. Further Discussion and Conclusion

As Tuberculosis is a global disease, it needs to have formal consideration so that patients can have better standard treatments in the near future to extend the life of human-being. For completing this project, we have collected the standard datasets as well as applied the most advanced neural network models to improve the confidence of models. Constantly improving the accuracy of models is the demanding need that we should focus on to support doctors and medical staffs as well as reduce the burden of the medical status.

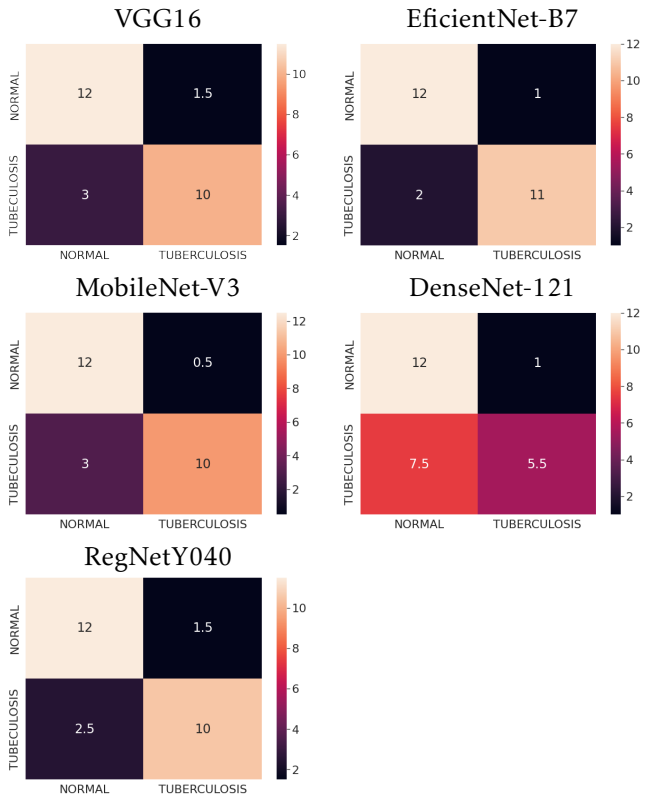


**Figure 13.** Confusion matrices for models applying to India Chest X-ray Set - MobileNet-V3 reveals the good results with True Positive and False Negative images detected are high - 12 and 10 correspondingly

Although DenseNet121 provides the highest results for the Shenzhen Chest X-ray dataset and EfficientNetB7 produces the best results for the India Chest X-ray dataset, MobileNetV3 returns the highest results for Tuberculosis (TB) Chest X-ray set, Montgomery Chest X-ray set in Accuracy and F1-Score metrics. In our study, MobileNetV3 is considered the most stable model which can be used to integrate into applications for further recommendations. Regarding the datasets, all of them are samples that have been collected and they cannot represent all data from the real practices.

Although the proposed models give us good results, this research still needs to be invested more. Many practical experiments under highly strict surveillance of experts should be performed to strongly enhance reliability before releasing into real production. For further research, firstly, we will extend this project by adding more diseases that are urgent such as pneumonia or COVID-19 to make the models stronger. Secondly, we will also extend the research by implementing object detection to understand which part of the images lead to the making decision process of the system. This will help to enhance the reliability of the system.

We finished the research by analyzing the cutting-edge Deep Neural Network-based techniques in terms of computational and operational effectiveness and efficiency. Then, we picked one well-suited model for the Tuberculosis evaluation problem that not only delivered good results in proposed metrics but also struck a balance between stability and computational complexity required during training phases. In the later works, we will try to collaborate with medical facilities to gain more potential data for the Tuberculosis domain and apply deeper techniques to improve as well as integrate new diseases to make the system more diverse based on the spatial body features of Vietnamese people.

## References

[1] Ángela Casado-García et al. "CLoDSA: a tool for augmentation in classification, localization, detection, semantic segmentation and instance segmentation tasks". In: *BMC Bioinformatics* 20.1 (June 2019), p. 323. ISSN: 1471-2105. DOI: 10.1186/s12859-019-2931-1. URL: https://doi.org/10.1186/s12859-019-2931-1.

[2] Arun Chauhan, Devesh Chauhan, and Chittaranjan Rout. "Role of Gist and PHOG Features in Computer-Aided Diagnosis of Tuberculosis without Segmentation". en. In: *PLoS ONE* 9.11 (Nov. 2014). Ed. by Hans A. Kestler, e112980. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0112980. URL: https://dx.plos.org/10.1371/journal.pone.0112980.

[3] "Detection of Pulmonary Tuberculosis Manifestation in Chest X-Rays using Different Convolutional Neural Network (CNN) Models". en. In: *International Journal of Engineering and Advanced Technology* 9.1 (Oct. 2019), pp. 2270–2275. ISSN: 2249-8958. DOI: 10.35940/ijeat.A2632.109119. URL: https://www.ijeat.org/wp-content/uploads/papers/v9i1/A2632109119.pdf.

[4] R. Dinesh Jackson Samuel and B. Rajesh Kanna. "Tuberculosis (TB) detection system using deep neural networks". en. In: *Neural Computing and Applications* 31.5 (May 2019), pp. 1533–1545. ISSN: 0941-0643, 1433-3058. DOI: 10.1007/s00521-018-3564-4. URL: http://link.springer.com/10.1007/s00521-018-3564-4.

[5] Linh T. Duong et al. "Detection of tuberculosis from chest X-ray images: Boosting the performance with vision transformer and transfer learning". en. In: *Expert Systems with Applications* 184 (Dec. 2021), p. 115519. ISSN: 09574174. DOI: 10.1016/j.eswa.2021.115519. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417421009295.

[6] Andrew Howard et al. "Searching for MobileNetV3". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 1314–1324. ISBN: 9781728148038. DOI: 10.1109/ICCV.2019.00140. URL: https://ieeexplore.ieee.org/document/9008835/.

[7] Gao Huang et al. "Densely Connected Convolutional Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 2261–2269. ISBN: 9781538604571. DOI: 10.1109/CVPR.2017.243. URL: https://ieeexplore.ieee.org/document/8099726/.

[8] Stefan Jaeger et al. "Two public chest X-ray datasets for computer-aided screening of pulmonary diseases". In: *Quantitative Imaging in Medicine and Surgery* 4.6 (Dec. 2014), pp. 475–477. ISSN: 2223-4292. DOI: 10.3978/j.issn.2223-4292.2014.11.20. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4256233/.

[9] Sonaal Kant and Muktabh Mayank Srivastava. "Towards Automated Tuberculosis detection using Deep Learning". In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. Bangalore, India: IEEE, Nov. 2018, pp. 1250–1253. ISBN: 9781538692769. DOI: 10.1109/SSCI.2018.8628800. URL: https://ieeexplore.ieee.org/document/8628800/.

[10] T Karnkawinpong and Y Limpiyakorn. "Classification of pulmonary tuberculosis lesion with convolutional neural networks". In: *Journal of Physics: Conference Series* 1195 (Apr. 2019), p. 012007. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1195/1/012007. URL: https://iopscience.iop.org/article/10.1088/1742-6596/1195/1/012007.

[11] Tawansongsang Karnkawinpong and Yachai Limpiyakorn. "Chest X-Ray Analysis of Tuberculosis by Convolutional Neural Networks with Affine Transforms". en. In: *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence - CSAI '18*. Shenzhen, China: ACM Press, 2018, pp. 90–93. ISBN: 9781450366069. DOI: 10.1145/3297156.3297251. URL: http://dl.acm.org/citation.cfm?doid=3297156.3297251.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks". en. In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782, 1557-7317. DOI: 10.1145/3065386. URL: https://dl.acm.org/doi/10.1145/3065386.

[13] Paras Lakhani and Baskaran Sundaram. "Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks". In: *Radiology* 284.2 (Aug. 2017), pp. 574–582. ISSN: 0033-8419. DOI: 10.1148/radiol.2017162326. URL: https://pubs.rsna.org/doi/10.1148/radiol.2017162326.

[14] Luyao Ma et al. "Developing and verifying automatic detection of active pulmonary tuberculosis from multi-slice spiral CT images based on deep learning". In: *Journal of X-Ray Science and Technology* 28.5 (Sept. 2020), pp. 939–951. ISSN: 08953996, 10959114. DOI: 10.3233/XST-200662. URL: https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/XST-200662.

[15] Ilija Radosavovic et al. "Designing Network Design Spaces". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Seattle, WA, USA: IEEE, June 2020, pp. 10425–10433. ISBN: 9781728171685. DOI: 10.1109/CVPR42600.2020.01044. URL: https://ieeexplore.ieee.org/document/9156494/ (visited on 02/20/2022).

[16] Tawsifur Rahman et al. "Reliable Tuberculosis Detection Using Chest X-Ray With Deep Learning, Segmentation and Visualization". In: *IEEE Access* 8 (2020), pp. 191586–191601. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2020.3031384. URL: https://ieeexplore.ieee.org/document/9224622/.

[17] Mohammad Imran Shah et al. "Ziehl–Neelsen sputum smear microscopy image database: a resource to facilitate automated bacilli detection for tuberculosis diagnosis". en. In: *Journal of Medical Imaging* 4.2 (June 2017), p. 027503. ISSN: 2329-4302. DOI: 10.1117/1.JMI.4.2.027503. URL: http://medicalimaging.spiedigitallibrary.org/article.aspx?doi=10.1117/1.JMI.4.2.027503.

[18] Manali Shaha and Meenakshi Pawar. "Transfer Learning for Image Classification". In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. Coimbatore: IEEE, Mar. 2018, pp. 656–660. ISBN: 9781538609651. DOI: 10.1109/ICECA.2018.8474802. URL: https://ieeexplore.ieee.org/document/8474802/.

[19] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations*. 2015.

[20] Christian Szegedy et al. "Going deeper with convolutions". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston, MA, USA: IEEE, June 2015, pp. 1–9. ISBN: 9781467369640. DOI: 10.1109/CVPR.2015.7298594. URL: http://ieeexplore.ieee.org/document/7298594/.

[21] Mingxing Tan and Quoc Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". en. In: *International Conference on Machine Learning*. PMLR, May 2019, pp. 6105–6114. URL: http://proceedings.mlr.press/v97/tan19a.html.

[22] World Health Organization. *Global tuberculosis report 2021*. en. Geneva: World Health Organization, 2021. ISBN: 9789240037021. URL: https://apps.who.int/iris/handle/10665/346387.