Automatic Detection of White Plague using New Modified VGG16 Network with Chest X-Rays

Sweekar Sudhakara

May 5, 2020

Contents

1 Project Title

Automatic Detection of White Plague using New Modified VGG16 Network with Chest X-Rays

2 Abstract

White plague (tuberculosis) is one of the most commonly existing disease in the mankind and typically chest x-rays are used for the radiology examination. However, the existing pattern recognition approaches which includes convolutional neural networks (CNNs) for the tuberculosis disease detection need further improvement to detect the tuberculosis infested chest x-ray accurately and be deployed in real world testing. Therefore, in this project we introduce a new modified VGG16 convolutional neural network that can identify the abnormalities effectively. In addition, we also introduce ladder network, a semi-supervised technique known to perform well on fewer labelled data for the classification task and this network has never been evaluated on bio-medical applications so far. Further, we compare these networks and depict layer wise visualization of the filters involved in the convolutional layers along with the data distribution using t-SNE.

3 Introduction and Problem Description

The white plague (tuberculosis) is one of the most commonly occurring lung disease on this planet. Around 75 million people (i.e. 1% of total population in the world) are affected by the white plague alone and approximately 1.6 million deaths occur each year due to this [1]. Due to its low-cost and easy-access nature, chest radiography, colloquially called chest x-ray (CXR), is one of the most common types of radiology examinations for the diagnosis of thorax diseases. The large number of chest radiographs produced globally are currently analyzed almost entirely through visual inspection on a slice-by-slice basis. This requires a high degree of skill and concentration, and is time-consuming, expensive, prone to operator bias, and unable to exploit the invaluable informatics contained in such large-scale data [2]. Moreover, due to the complexity of chest radiographs, it is challenging even for radiologists to discriminate between normal and disease infested x-ray.

To automatically differentiate between the normal and tuberculosis infested chest x-rays, machine learning learning based models were employed by several researchers. Commonly, convolutional neural networks (CNN) are the most common type employed by these researchers [2–5] for the image based classification task. However, the state-of-the-art performance of these networks for the tuberculosis detection needs further improvement for deployment in real-world testing to aid the doctors. In order to improve on the existing performance we introduce the new modified VGG16 CNN for the classification task inspired from the well known VGG16 network [6] introduced for ILSVRC 2014 competition [7]. In addition, we also implement the ladder networks [8] a branch of machine learning, combine supervised with unsupervised learning and is trained towards reducing the combined error of both these techniques. The ladder networks with the semi-supervised approach trained with minimal labelled data is known to exhibit good performance on unseen data. Further, we summarise the performance of each of

the implemented model with statistical comparison and visualize the working of the new proposed model using the t-SNE algorithm along with the visualization of each convolutional layers in the model.

NOTE:

In this project we solve the **classification problem** with **two different classes**:

1. **Normal** chest x-ray.
2. **Tuberculosis** infested chest x-ray.

In order to solve the binary classification problem, we implement the following models:

1. **Baseline** model based on **supervised learning**.
2. **New Modified VGG16 Network** based on **supervised learning**.
3. **Ladder Network** based on **semi-supervised learning**.

LINK TO THE DATA: OneDrive Download Link (Click here).

4 Related Work

There has been significant contributions to classification of Tuberculosis manifested chest x-rays using several convolutional neural networks. Also, there has been work on ladder networks on different datasets other than bio-medical field. Both the cases are elaborated in the following sections.

4.1 Related work on the dataset area

In this section, we have discussed the state of the art in the area as well as other work complementing this area.

4.1.1 State of the art in this area

Meraj et al. [2], have used Convolutional Neural Networks (CNN) models appended with pre-trained models such as VGG-16, VGG-19, RestNet50, and GoogLeNet implemented in order to identify TB manifested chest x-rays [9]. They find that VGG-16 model generalizes well on the data. They apply data-augmentation techniques such as flipping the chest x-rays to compensate for less data.

The author's have trained their proposed model on the Shenzhen data and Montgomery data individually (Section 5). They obtain an accuracy (performance) of 86.74% on Shenzhen data and 77% on Montgomery data which is currently the best with this data. Additionally, Lopes et al. [5] has used an ensemble training algorithm and achieved a accuracy of 85% on the combination of the two data similar to how we plan to use the two combined data for our work.

4.1.2 Other works in this area

Pasa et al. [3] built a convolutional neural network optimized for the problem with architecture similar to AlexNet with augmented chest x-ray images. The authors employed a 5 step multi-convolutional layers along with global average pooling with He-weight initializer.

However, the post augmented data size used by all of these approaches are not specified. Also, the network architectures and hyper-parameters used by the these researchers are not clearly mentioned, hence making this problem open to pruning to achieve better results.

4.2 Related work on the Pattern Recognition Approach

VGG16 convolutional neural network model was introduced by Simonyan et al. [6] which is a well known model submitted to ILVSR 2014 competition [7]. It was trained and tested on ImageNet data with over 1000 classes across 14 million images. The VGG16 model pre-trained weights from ImageNet training was employed by Meraj et al. [2] for the tuberculosis task as mentioned in the previous section.

Ladder network, a semi-supervised technique was first employed by Rasmus et al. [8]. They combined supervised learning with unsupervised learning in deep neural networks and trained to simultaneously minimize the sum of supervised and unsupervised cost functions by back-propagation. They used it for classification task with minimal labelled data. It was trained and tested on MNIST database of hand-written digits [10].

However, ladder networks have been trained and tested with a few datasets only and its capability on real-world large minimal labelled data has not been evaluated. Hence, it is intriguing to evaluate ladder networks on real-world data particularly bio-medical images which are suffering from large unlabelled data problem.

4.3 Part of the project which differs from what has already been done

The model used in the baseline is a pre-trained network appended with global average pooling and fully-connected layers. The proposed new modified VGG16 network is a CNN model with new architecture inspired from the existing VGG16 network. The changes in the proposed network includes inclusion of global average pooling (GAP) instead of the flattening layer, change in number of convolutional layers, inclusion of batch normalization and change number of neurons in the fully connected layers.

5 The Tuberculosis Dataset

The experiments mentioned in Section 6 will be based on two publicly available datasets, Shenzhen & Montgomery that contain human chest frontal x-ray images with two different classes, i.e. chest x-ray images infected with tuberculosis (white plague) disease and chest x-ray images without the disease, respectively. The complete details of the dataset is mentioned in the brief report by Jaegar et al. [9].

	Normal	Tuberculosis	Total
Shenzhen Data	336	326	**662**
Montgomery Data	80	58	**138**
Overall Data	**416**	**384**	**800**

Table 1: Distribution of the data.

The dataset, Montgomery contains 138 chest frontal x-ray images which includes 80 x-ray images of lungs without tuberculosis disease, and 58 x-ray images of lungs infected with the disease. The Department of Health of the Montgomery County (Maryland, USA) collected all the x-ray images present in the Montgomery dataset. The x-ray images either have resolution of 4892×4020 (or) 4020×4892 pixels. The Montgomery dataset also additionally contains manually generated masks for lung segmentation of each and every sample present in the dataset. However, in this work, we will not be using the segmentation masks in any of the experiments. Figure 2 corresponds to the samples of the Montgomery dataset.

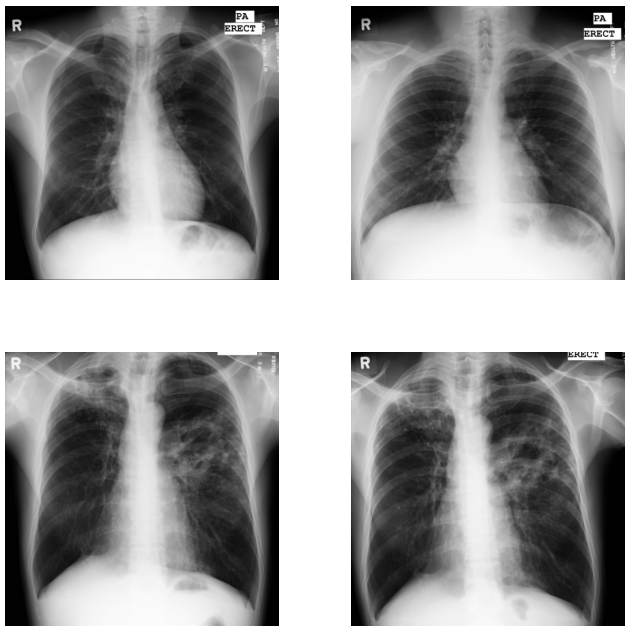

Figure 2: Examples of frontal chest x-rays from Montgomery dataset. The first row shows the image of normal lungs and the second row shows the image of tuberculosis infected lungs.

The dataset, Shenzhen contains 662 chest frontal x-ray images which includes 336 x-ray images of lungs without tuberculosis disease, and 326 x-ray images of lungs infected with the disease. The Guangdong Medical College (Shenzhen, China) collected all the x-ray images present in the Shenzhen dataset. The x-ray images have resolution of around 3000×3000 pixels. Figure 4 corresponds to the samples of the Shenzhen dataset. The overall distribution of the data is summarized and depicted in Table 1.

Link to the Data: OneDrive Download Link (Click here).

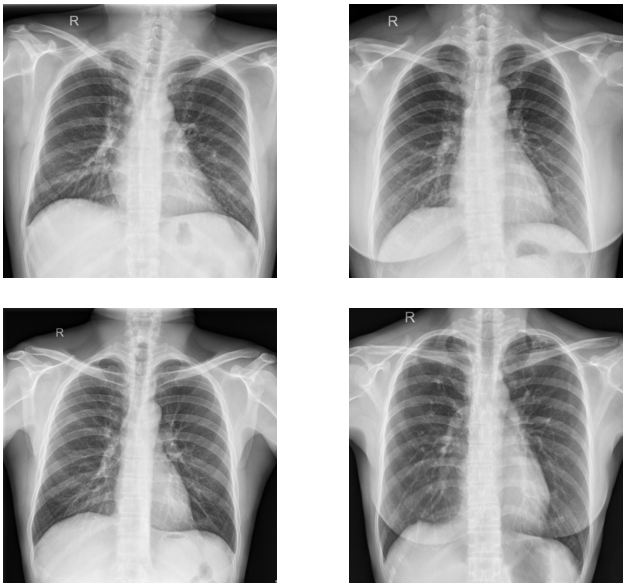

Figure 4: Examples of frontal chest x-rays from Shenzen dataset. The first row shows the image of normal lungs and the second row shows the image of tuberculosis infected lungs.

5.1 Why the dataset size is sufficient for my machine learning task?

This is a very common question that rises while solving a problem using machine learning. But the answer majorly depends on the problem statement. But to generalize, it can be said that the following parameters can be taken into consideration before choosing the size of the dataset i.e. 1) Number of class 2) Number of features 3) Inter-class and Intra-class variance 4) Model for classification.

The general rule of thumb is to use 10 times the number of parameters in the learning model. Let us take a look at the aspect of this project. In this project, we proposed to use new modified VGG16 network to accomplish the goal with classification problem of two classes; normal and tuberculosis, respectively. The new proposed network comprises of a total of 5.8M parameters. To satisfy the rule we need huge amount of data but since the proposed network comprises of regularization component in terms of batch normalization and dropout which prevents over-fitting. Thus, the total number of parameters is not much relevant to the data size. Additionally, the tuberculosis data is well studied from previous works and it was understood that the inter-class and intra-class variance was high enough. Besides these observations, resource constraints were also taken into consideration. With the available time, the memory and the other mentioned aspects we decided to augment the data for training and use the original data for testing. Although there is no quantitative justification to prove the choice is right, the chosen data is giving promising results.

6 Methods

The following methodology has been employed in this project:

1. Pre-processing of chest x-rays by cropping the black border, resizing the image to $224 \times 224 \times 3$ and lastly, whitening & centering of the image.

2. Data augmentation by scaling, rotation and translation of the original images to generate a larger set of images for training.

3. Implementation of the baseline model by Meraj et al. [2] with transfer learning using VGG16 and ResNet50 pre-trained weights on the tuberculosis data.

4. Implementation of ladder networks trained and tested on on the tuberculosis data.

5. Implementation of new modified VGG16 convolutional neural network trained and tested on the tuberculosis data.

6. Visualization of each layer of convolutional filter activations for the new proposed model. In addition, t-SNE on each layer of convolutional and fully connected layers of the proposed network and comparison of the final fully connected layer of the proposed network with the baseline model.

6.1 Pre-processing

The pre-processing of the input data containing the images includes the following steps:

- Any black border or black band existing in the edges is cropped from the image.

- The image is resized so that the length of the image is 224 pixels long and width is 224 pixels wide. The length and width has been chosen such that the images can be trained with pre-trained weights which generally consider (224×224) size images as the input.

- For each image, the mean over all pixels in the whole tuberculosis data is subtracted and the pixel values are divided by their standard deviation.

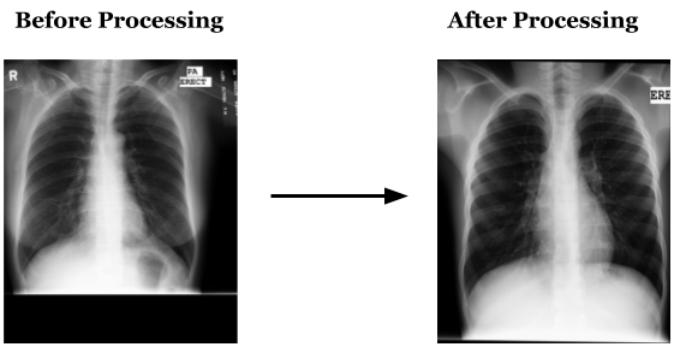

Figure 5: Example of the Pre-processing.

6.2 Data Augmentation

Types	Rotation	Width Shift	Length Shift	Scaling	Flip
Range	0 - 10	0 - 0.1	0 - 0.1	0 - 0.1	Horizontal

Table 2: Data Augmentation Types and Range.

To increase the data size for training we perform data augmentation. The reason for the data augmentation is that the existing open-source TB data contains only 800 samples which is not sufficient enough for training a neural network. Thus, instead of following the labelling of data which is computationally expensive we generate labelled data by the process of data-augmentation. It is a process where labelled data is generated by inflating the training set with labels preserving transformations artificially. The different data augmentation methods done in this project is depicted in Table 2 i.e. 1) scaling 2) rotation and 3) translation. In addition, data augmentation avoids over-fitting of data because of the presence of more number of samples to generalize on. In this work, we firstly split the complete TB data containing 800 samples into training and testing in the ratio 70 : 30 i.e. 560 (294 - Normal, 266 - Tuberculosis) and 240 (122 - Normal, 118 - Tuberculosis) samples, respectively. We further augment only the training data approximately 13 times to generate 7776 training samples (3812 - Normal, 3964 - Tuberculosis). The testing samples is not augmented to retain the originality. We perform the data augmentation using Keras Data Generator [11].

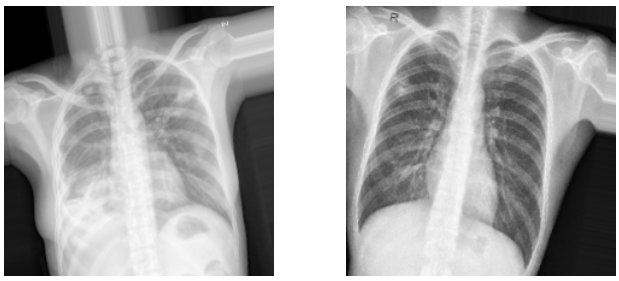

Figure 6: Examples of the augmented data.

6.3 Transfer Learning using Pre-trained Weights (Baseline)

In this project, we have used the network presented by Meraj et al. [2] as the baseline model for tuberculosis detection task in chest x-rays. The reason for choosing this model as the baseline is that it the most recent work in the tuberculosis detection task and it employs pre-trained weights. Hence, any model equivalent or better performance than this model is a significant contribution for this task. The architecture employed in the task is depicted in Figure 7 which gives the output '0' for Normal and '1' for tuberculosis infested image.

As depicted in the figure, the tuberculosis data is firstly split into training and testing data. The train data is further augmented to create more number of samples using Keras data augmentation. Further, the pre-trained model weights which has been previously trained on ImageNet data is loaded. To confine the weights from the pre-trained model the global average pooling layer is employed along with dropout of 0.5 and a dense layer of

1024 neurons is used again with a similar dropout. In the end, a sigmoid layer activation in a fully connected layer/dense layer is employed to predict the label of the chest x-ray. For the pre-trained weights we have considered VGG16 and ResNet50 models which performs the best as depicted by the authors for this task. The optimization function for the network is Adam [12] optimizer along with binary cross-entropy as the loss function.

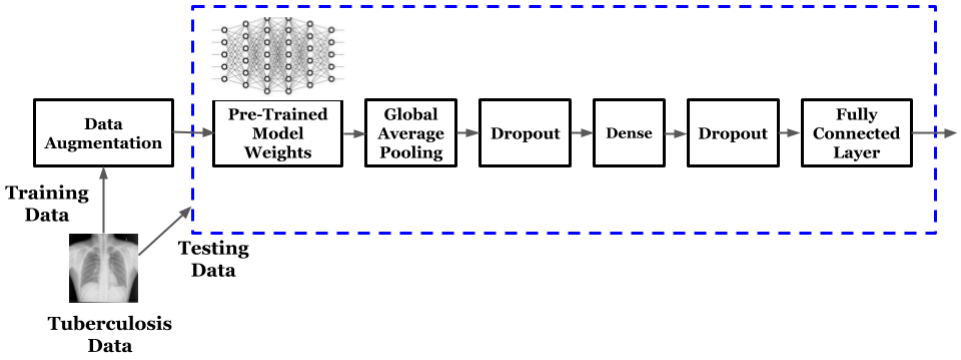

Figure 7: Illustration of the Baseline Approach.

What changes we have incorporated in the Baseline's Implementation?
The following are the changes we incorporated in comparison to the original baseline implementation:

1. The author's did not consider any form of pre-processing in the classification task whereas we pre-processed the images (6.1) before feeding it to the data-augmenter.
2. Originally, only data augmentation considering horizontal flip was considered and the post augmented data size was not mentioned. However, we consider 5 different augmentations combined mentioned in Table 2 and generated 7776 training samples after augmentation.
3. The value of the dropout and the dense layer size was not mentioned by the author's and we found the best possible values iteratively through experimentation. Additionally, the best possible learning rate was found iteratively.
4. The author's originally conducted experiments separately on the two parts of the tuberculosis data depicted in Section 5 (data-split, 75 : 25) and obtain a maximum accuracy (performance) of 86% & 77% for Shenzhen & Montgomery data, respectively. However, we combine both the dataset's (data-split, 70 : 30) & iteratively find the hyper-parameters and obtain a performance better than the original work with approximately 87% accuracy.

6.3.1 VGG16

Figure 8 depicts the architecture of VGG16 convolutional neural network [6]. The weights correspond to the VGG16 model trained and tested on ImageNet data with 1000 classes and 14 million images. In this work, the last fully connected layer depicted in the above figure has been chopped off and replaced by a GAP layer along with dense layers and the whole network which is trained for a total of 10 epochs.

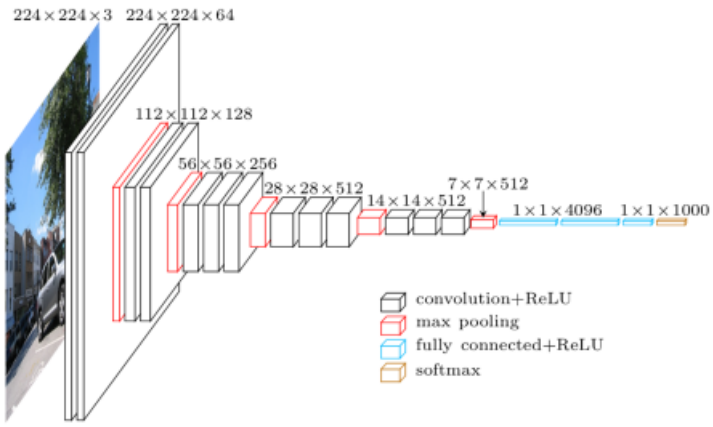

Figure 8: VGG-16 Architecture.

6.3.2 ResNet50

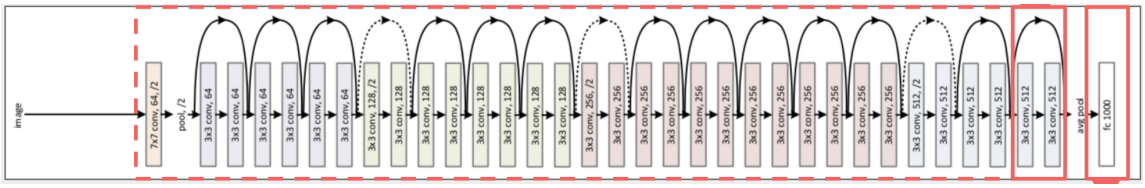

Figure 9: ResNet50 Architecture.

Figure 9 depicts the architecture of ResNet50 convolutional neural network [13]. The weights correspond to the ResNet50 model trained and tested on ImageNet data with 1000 classes and 14 million images. In this work, the last fully connected layer depicted in the above figure has been chopped off and replaced by a GAP layer along with dense layers and the whole network which is trained for a total of 10 epochs.

6.4 Ladder Networks (Proposed Method - I)

The ladder network is an application of supervised and unsupervised learning together. The unsupervised learning task is associated with the denoising of the layer representations at each level of the network. The structure of the network is similar to the auto-encoder with skip connections from the encoder to the decoder. The network's task is similar to of the denoising auto-encoders except that it is not only applied to the inputs but also to all the layers. The skip connections in the network aids in representing the details in the higher layers such that the decoder can recover any details discarded by the encoder.

Figure 10, illustrates the ladder network. The rightmost feed-forward network is known as the "clean" part and the remaining network is known as the "noisy" part. The noisy part comprises of an encoder-decoder trained to denoise the each layer and the clean part is feed forward network with several fully connected layers trains on clean data to predict the label and provides the feedback for the denoising encoder-decoder. The

clean input is simultaneously provided to both the feed-forward network and the encoder-decoder which contains layers injected with zero mean Gaussian noise, this noise induces a regularizing effect. The denoising network employs skip connections to keep track of the information even in the higher layers and the cost function associated with the predicted label $\tilde{y}$ is given by $(-\log\{\mathcal{P}(\tilde{y} \mid x)\})$ (categorical cross-entropy loss). In addition, the enocoder-decoder network learns the denoising parameters by comparing the denoising terms in each layer with the clean network along with the cost function, $\|z^{(l)} - \hat{z}^{(l)}\|^2$ (reconstruction cost). The autoencoder then propagates backwards and recreates each layer's z values using the denoising function.

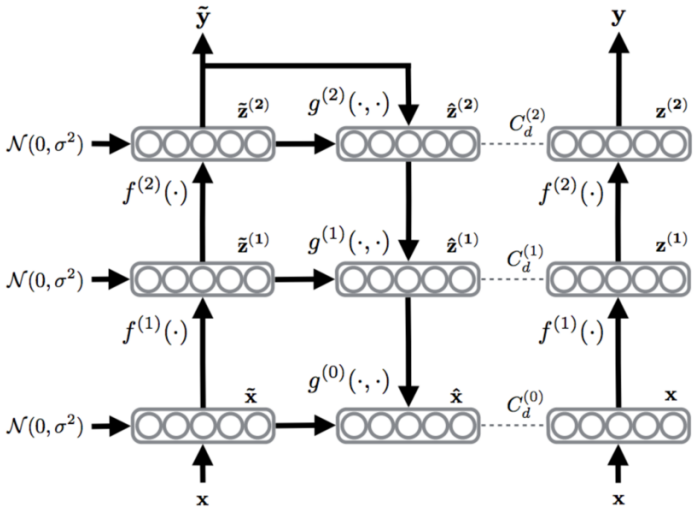

Figure 10: Illustration of the Ladder Network.

6.4.1 Cost function

The overall loss function of the ladder network is the combination of the classification loss i.e. the cross entropy loss from the encoder-decoder part and the reconstruction cost of all the layers i.e. the difference between the denoised and clean z value which can be depicted as follows:

$$\text{Cost} = (-\log\{\mathcal{P}(\tilde{y} \mid x)\}) + \sum_{l=0}^{L} \lambda_l \times \|z^{(l)} - \hat{z}^{(l)}\|^2 \tag{1}$$

6.5 New Modified VGG16 Convolutional Neural Network (Proposed Method - II)

In this project, in addition to the ladder network (section 6.4) and the baseline network by Meraj et al. (section 6.3) implementation, we also introduce a new modified VGG16 network for the tuberculosis detection task. The proposed new modified VGG16 network performs better in comparison to the baseline network and the semi-supervised ladder network and this is further illustrated in the sections to follow. The new network is a convolutional neural network (CNN) which takes the input chest x-ray image and outputs

Figure 11: Original VGG16 Architecture.

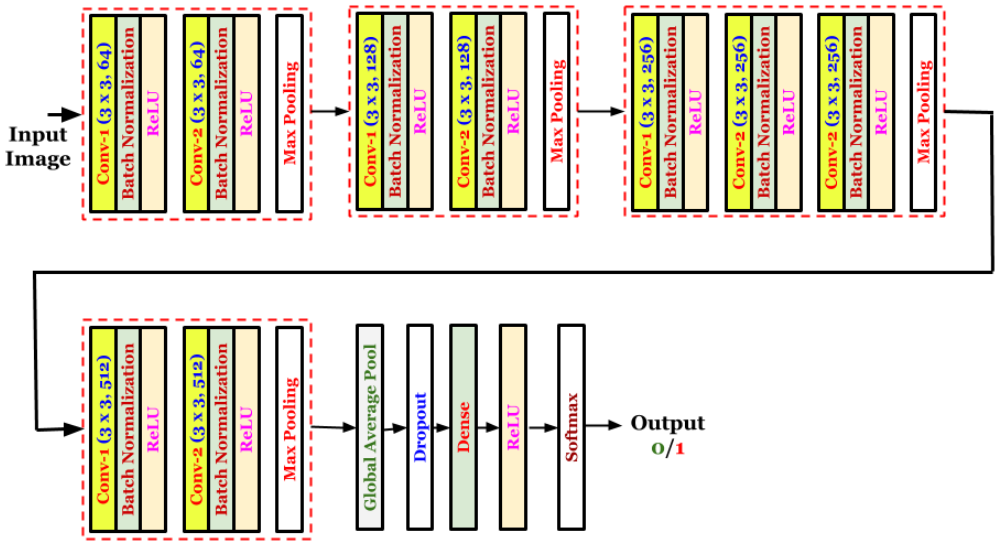

Figure 12: Proposed New Modified VGG16 Architecture.

whether the x-ray is TB infested or not. The proposed new network is inspired from the VGG16 convolutional neural network [6] illustrated in figure 8. The VGG16 network was used to win ILSVR (ImageNet) competition in 2014 and is considered to be one of the excellent computer vision based model architecture till date. The interesting thing about the VGG16 is that instead of containing large number of hyper-parameters it instead has convolution layers of 3x3 filter with a stride 1 and contains same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture with a 2 fully connected layers and a softmax activation in the end.

Our proposed network is a modification over the original VGG16 network described in figure 11 and the new modified VGG16 network is illustrated in figure 12. We have included the following modifications to the existing VGG16 architecture i.e. 1) batch normalization has been included for each convolutional layer 2) the total number of convolutional layers have been reduced, particularly the last set of convolutional layer pack with 512 filters of size (3×3) has been dropped. 3) instead of a flattening layer a global average pooling layer has been employed along with dropout. 4) lastly the number of neurons in the last fully connected layer has been reduced to 1024 as opposed to the original 4096 neurons. The optimization function for the network is Adam [12] optimizer along with categorical cross-entropy since the labels for the images are one-hot encoded.

The network achieves performance better than all the network mentioned in this work and the functioning of the same with layer wise data and filter visualization is depicted in further sections. The description of the parameters resulting in each layer of the network is depicted in Figure 13.

6.6 t-SNE Algorithm

The t-SNE algorithm, known as t-Distributed Stochastic Neighbor Embedding is a machine learning algorithm employed for dimensionality reduction that is particularly well suited for high-dimensional data. In other words, it allocates each high-dimensional object by a two or three-dimensional point in such a way the distribution remains same i.e. similar objects are associated by nearby points and distant points are associated with dissimilar objects. The algorithm functions in two separate steps, i.e. firstly, it constructs a probability distribution over pairs of high dimensional objects similar objects have a high probability and dissimilar objects have a low probability of being picked. In the next step, it defines a probability distribution similar to the previous one, over the points in the low-dimensional map with the aim to minimize the KL-Divergence between the two distributions based on the locations of the points in the map.

6.7 Training Device Description

For this project, we have used the CyberLAMP GPU resources. The configuration was single shared GPU with 4 nodes and 8GB memory. We employed Python 3.7 on CyberLAMP for our experiments as this version aligned with the starter code setup provided.

7 Results and Discussion

The following sections contain a detailed overview of the results and the performance corresponding to each of the network architectures trained with the TB data. We also present the performance, filter visualization, saliency map and the t-SNE application.

7.1 Transfer Learning using Pre-trained Weights (Baseline)

As mentioned earlier, the network architecture described in Meraj et al. 7 has been employed as the baseline for the classification task. As depicted we use two different state-of-the-art pre-defined weights i.e. VGG16 and ResNet50 in the baseline model. We present the results and discussion for both of these networks together. The overall results/performance of both the pre-defined weights have been depicted in Table 3.

Classifier	Tuberculosis (Sensitivity)		Non-Tuberculosis (Specificity)		Overall	
	Training	Testing	Training	Testing	Training	Testing
VGG16	93.14	82.20	97.76	92.62	95.41	87.08
ResNet50	93.79	79.66	95.01	95.08	95.67	87.50

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
conv2d_1 (Conv2D)	(None, 224, 224, 64)	1792
batch_normalization_1 (Batch	(None, 224, 224, 64)	256
activation_1 (Activation)	(None, 224, 224, 64)	0
conv2d_2 (Conv2D)	(None, 224, 224, 64)	36928
batch_normalization_2 (Batch	(None, 224, 224, 64)	256
activation_2 (Activation)	(None, 224, 224, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 112, 112, 64)	0
conv2d_3 (Conv2D)	(None, 112, 112, 128)	73856
batch_normalization_3 (Batch	(None, 112, 112, 128)	512
activation_3 (Activation)	(None, 112, 112, 128)	0
conv2d_4 (Conv2D)	(None, 112, 112, 128)	147584
batch_normalization_4 (Batch	(None, 112, 112, 128)	512
activation_4 (Activation)	(None, 112, 112, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 56, 56, 128)	0
conv2d_5 (Conv2D)	(None, 56, 56, 256)	295168
batch_normalization_5 (Batch	(None, 56, 56, 256)	1024
activation_5 (Activation)	(None, 56, 56, 256)	0
conv2d_6 (Conv2D)	(None, 56, 56, 256)	590080
batch_normalization_6 (Batch	(None, 56, 56, 256)	1024
activation_6 (Activation)	(None, 56, 56, 256)	0
conv2d_7 (Conv2D)	(None, 56, 56, 256)	590080
batch_normalization_7 (Batch	(None, 56, 56, 256)	1024
activation_7 (Activation)	(None, 56, 56, 256)	0
max_pooling2d_3 (MaxPooling2	(None, 28, 28, 256)	0
conv2d_8 (Conv2D)	(None, 28, 28, 512)	1180160
batch_normalization_8 (Batch	(None, 28, 28, 512)	2048
activation_8 (Activation)	(None, 28, 28, 512)	0
conv2d_9 (Conv2D)	(None, 28, 28, 512)	2359808
global_average_pooling2d_1 (	(None, 512)	0
dropout_1 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 1024)	525312
dense_2 (Dense)	(None, 2)	2050

Total params: 5,811,522
Trainable params: 5,807,170
Non-trainable params: 4,352

Figure 13: Parameters description of the new modified VGG16 network.

Table 3: Performance of Transfer Learning using Pre-trained Models.

The results were obtained after training the network for 10 epochs with a learning rate of

8×10^{-5} and 1×10^{-4} for VGG16 and ResNet50, respectively. Additionally, a batch size of 8 was used for training with a duration of 1.5 hrs. The baseline was optimized using the Adam optimizer [12] with the learning rates mentioned previously along with the binary cross-entropy loss function since the final classification layer is sigmoid activation function.

Figure 14, 15, 16 & 17 represent the confusion and classification matrices of training and testing respectively. Figure 18 & 19 represents the accuracy and loss over epochs. In the confusion matrices, the rows correspond to the true class and the columns correspond to the predicted class. Diagonal and off-diagonal cells correspond to correctly and incorrectly classified observations, respectively. The column on the far right of the classification matrices shows the percentages of all the examples predicted to belong to each unit lattice group that are correctly and incorrectly classified. These metrics are employed to compute the sensitivity which measures the proportion of actual positives that are correctly identified and specificity is similar but it is related to actual negatives.

In Table 3, it can be observed that the sensitivity of VGG16 pre-trained model instilled in the architecture gives a better sensitivity than the ResNet50 instilled model which is desired for a classifier employed in bio-medical testing as it detect tuberculosis more accurately than the normal person detection which is required, since there is no harm if a normal person is classified as tuberculosis patient and it is dangerous if a TB patient is labeled as normal. In addition, as mentioned by the authors VGG16 performs significantly better than the ResNet50 model in terms of the overall performance comprising across training and testing.

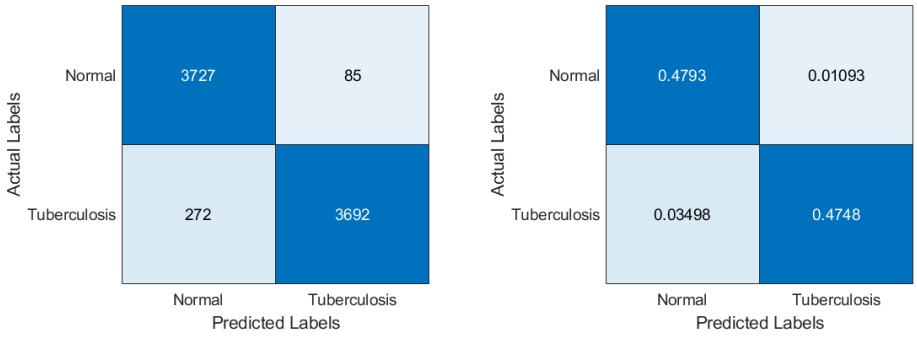

Figure 14: Training confusion and classification matrices for VGG16 pre-trained weights.

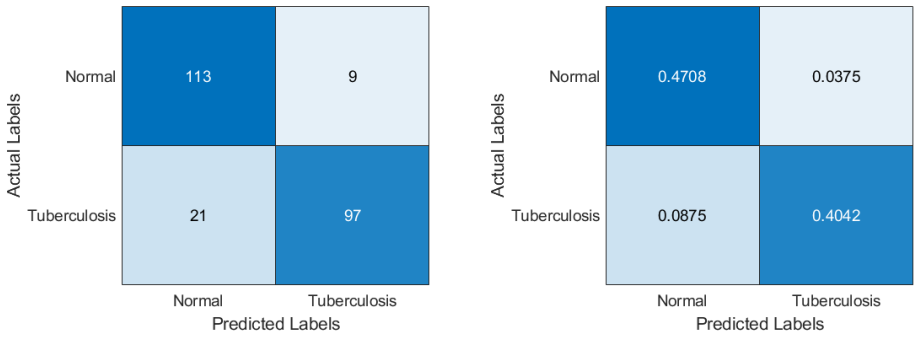

Figure 15: Testing confusion and classification matrices for VGG16 pre-trained weights.

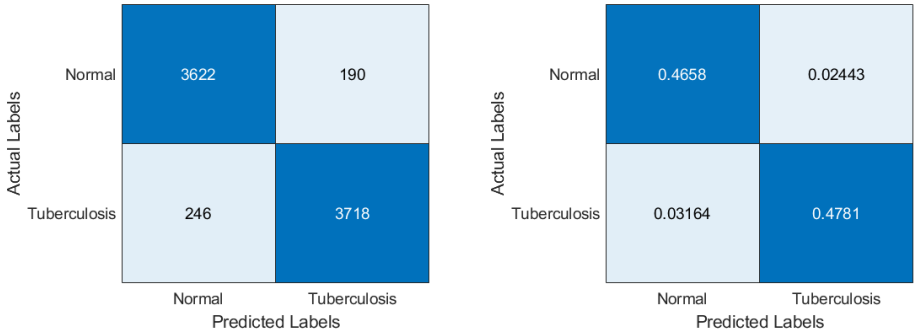

Figure 16: The confusion and classification matrices of training TB data set using ResNet50 pre-trained weights.

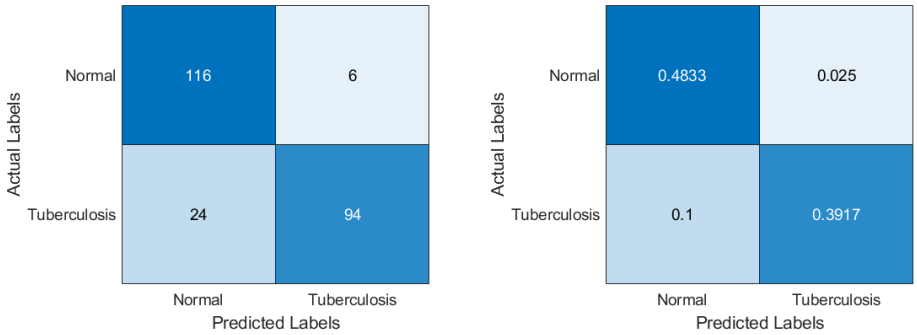

Figure 17: Training confusion and classification matrices for ResNet50 pre-trained weights.

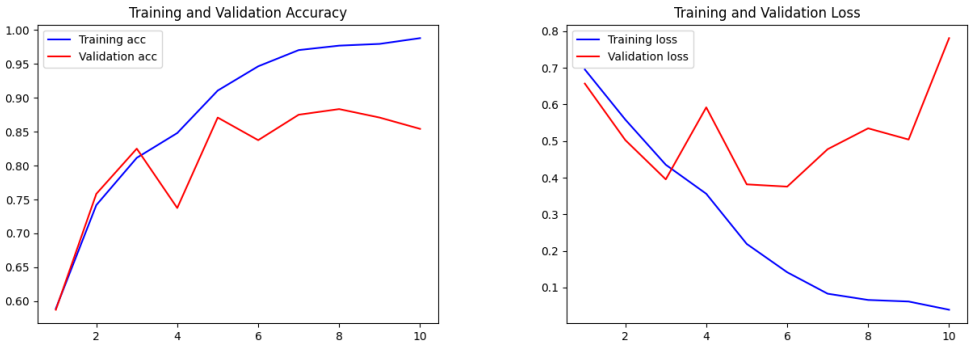

Figure 18: The training and validation accuracy & loss using VGG16 pre-trained weights.

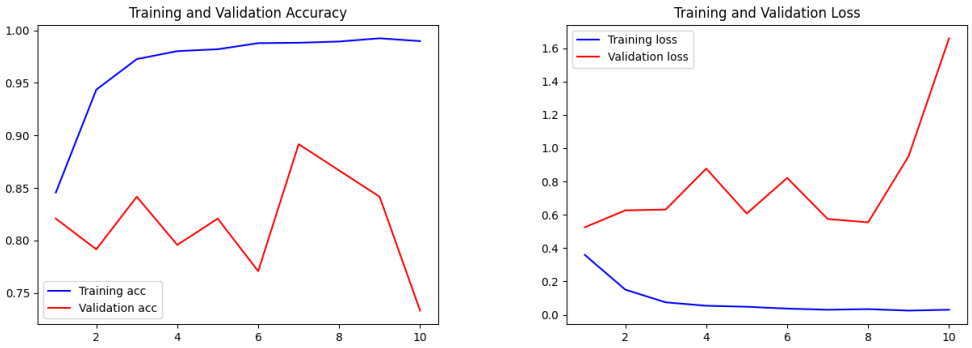

Figure 19: The training and validation accuracy & loss using ResNet50 pre-trained weights.

7.1.1 Saliency Map/Heat Map

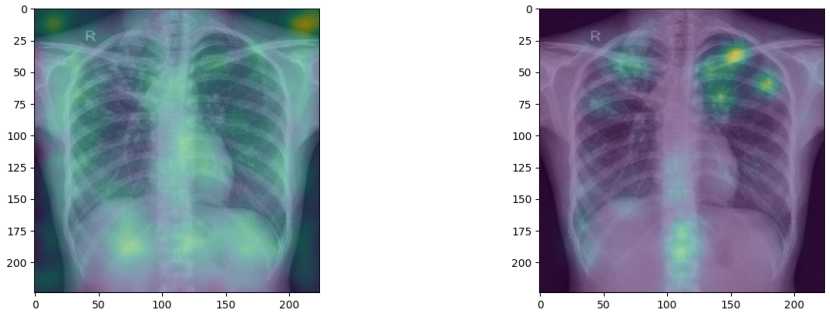

Figure 20: Saliency Maps of VGG16 and ResNet50 Pre-trained Models.

Once the network was trained for classification, we also generated saliency maps. The saliency map is a visualization technique which help us understand the network and may also be useful as an approximate visual diagnosis for presentation to radiologists.

Saliency maps generate a heatmap that shows which region of the image weights more for the classification. The principle these visualizations are based is the following: the derivative of the output class score w.r.t. to an activation in a feature map indicates the impact this activation has on the class score. If the derivative is small, then a change in the activation will have a negligible impact on the output score, therefore the activation is unimportant for the classification. On the contrary, a big derivative indicates that the activation is important for the class score. From Figure 20, it can be observed that the VGG16 and ResNet50 both have their weights concentrated at foggy part existing in the lungs which is the main distinguishing factor between normal and TB infested person.

7.1.2 Filter Visualization

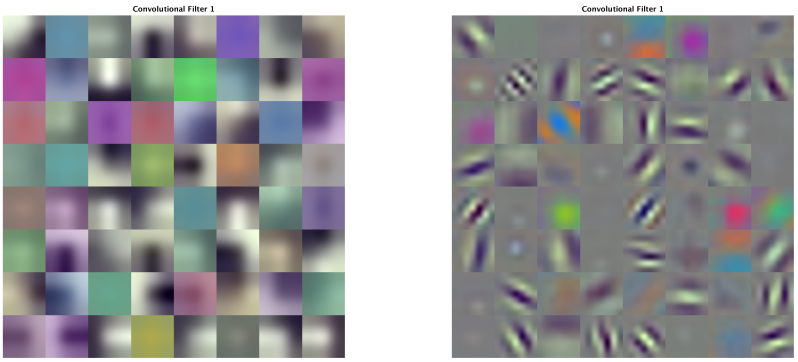

Figure 21: The Convolutional Layer 1 Filter Visualization for VGG16 and ResNet50 Pre-trained Models.

In this section, we visualize each filter present in the first convolution layer for each of the pre-trained models employed in the architecture experimented in this project. The visualization for each corresponding filter in the convolutional layer have been depicted using a single image. From the figure, it can be observed that the filters in the first convolutional layer learn fundamental features from the images such as edges.

7.2 Ladder Networks (Proposed Method - I)

The ladder network employed in this work contains totally seven fully connected layers with number of neurons: $128 \times 128, 1000, 500, 250, 250, 250, 2$, respectively and the input to the network is an image of size (128×128). For each layer in the noisy part of the network (discussed in Section 6.4) we add a Gaussian noise with variance of 0.3 and a batch normalization term. We have employed Adam optimizer [12] for optimizing the network with learning rate 0.002. The network is trained for a total of 100 epochs with a duration of 1.5 hr.

Figure 22 and 23 represent the confusion and classification matrices of training and testing respectively. Figure 24 represents the accuracy and loss over epochs. In the confusion matrices, the rows correspond to the true class and the columns correspond to the predicted class. Diagonal and off-diagonal cells correspond to correctly and incorrectly classified observations, respectively. The column on the far right of the classification matrices shows the percentages of all the examples predicted to belong to each unit lattice group that are correctly and incorrectly classified. These metrics are employed to compute the sensitivity which measures the proportion of actual positives that are correctly identified and specificity is similar but it is related to actual negatives.

Classifier	Tuberculosis (Sensitivity)		Non-Tuberculosis (Specificity)		Overall	
	Training	Testing	Training	Testing	Training	Testing
Ladder Network	92.23	81.36	92.92	86.07	92.56	83.75

Table 4: Performance of Ladder Networks.

In Table 4, it can be observed that the sensitivity of the ladder network is lower than its specificity and it is harmful in bio-medical testing as it suggests that TB samples are being mis-classified as normal instead of TB. However, the overall performance of the ladder network is relatively lower in comparison to the baseline model's performance depicted in section 7.1. This could be because of the employment of only fully-connected layers in the feed-forward path which fail to recognise the important features underlying in the x-ray image unlike the convolutional layers. This suggests the need for further fine-tuning in order to improve the ladder network's performance over the baseline model's performance and to be deployed in real world testing. From table 5 it can be observed that the model's performance is similar across different random splits of test/train depicting the proposed network's consistency in performance.

Type	Test/Train Split	Total Splits	Mean	Maximum/ Minimum	Variance	Standard Deviation
Testing	70 : 30	5	83.80	84.45/82.91	0.28	0.53
Training			93.51	95.82/91.08	2.89	1.70

Table 5: Summary of the performance of Ladder Network based on 5 random test/train split of ratio 70 : 30.

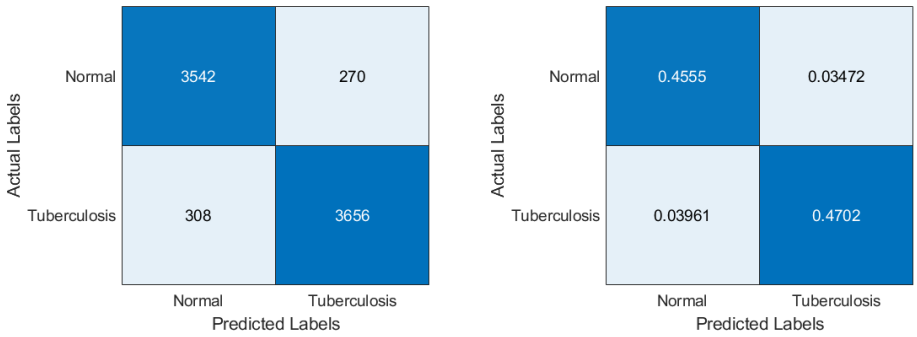

Figure 22: Training confusion and classification matrices for Ladder Network.

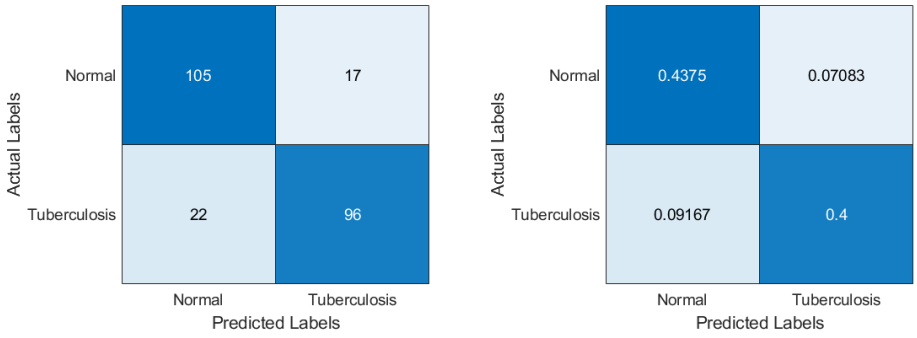

Figure 23: Testing confusion and classification matrices for Ladder Network.

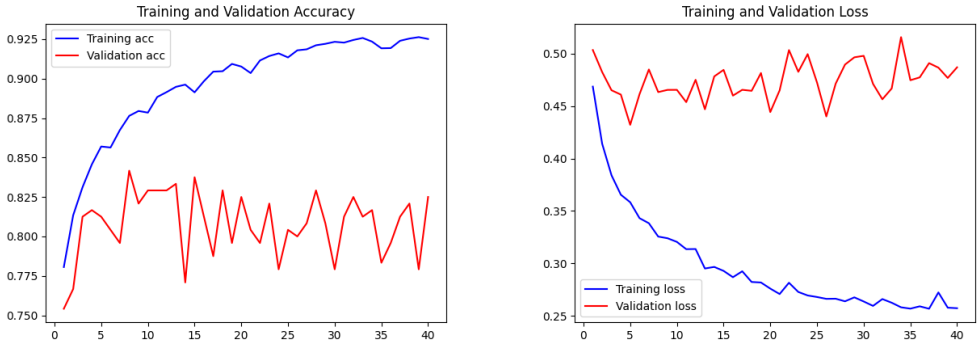

Figure 24: The training and validation accuracy & loss using Ladder Network.

7.3 New Modified VGG16 Convolutional Neural Network (Proposed Method - II)

The architecture of the new modified VGG16 network is depicted in figure 6.5. The network contains totally 9 convolutional layer along with batch normalization and relu activation function, which is succeeded by a global average pooling along with a softmax classification layer. The proposed network was trained for a total of 200 epochs over a duration of 5 hrs. The input image size for the network was $224 \times 224 \times 3$. Training was performed using categorical cross-entropy as the error function and with mini-batches of 4 samples. The samples were shuffled after each epoch before forming the mini-batches, in order to randomize the whole learning procedure and reduce over-fitting. The model was optimized considering the Adam optimizer [12] with a learning rate of 1×10^{-5}.

Figure 25 and 26 represent the confusion and classification matrices of training and testing respectively. Figure 27 represents the accuracy and loss over epochs. Table 6 summarizes the overall performance of the new proposed network along with the sensitivity and specificity measure of the model. Additionally, Table 7 illustrates the performance of the proposed network in terms of accuracy for 5 different random test/train splits of ratio 70 : 30.

Classifier	Tuberculosis (Sensitivity)		Non-Tuberculosis (Specificity)		Overall	
	Training	Testing	Training	Testing	Training	Testing
Proposed Network	96.01	86.44	94.70	90.16	95.37	88.33

Table 6: Performance of the New Modified VGG16 Network.

In Table 6, it can be observed that the sensitivity of the new proposed network is slightly better than its specificity and it is preferable in bio-medical testing as it suggests that TB samples are correctly being labeled as TB instead of normal. In addition, the overall performance of the new proposed network is better in comparison to the baseline model's performance depicted in section 7.1 and the semi-supervised technique based ladder network (section 7.2). The overall better performance of the new modified VGG16 network could be because it has relatively lower number of parameters along with normalization term which avoids over-fitting and improves the performance in comparison to the baseline model. Also, the proposed network is trained from scratch on the TB data unlike the baseline model which integrates weights trained on some other data. In addition, the proposed network unlike the ladder network employs deep convolutional layers to capture information from the image accurately. Further, from table 7 it can be observed that the model's performance is similar across different random splits of test/train depicting the proposed network's consistency in performance.

Type	Test/Train Split	Total Splits	Mean	Maximum/ Minimum	Variance	Standard Deviation
Testing	70 : 30	5	88.05	88.75/87.08	0.32	0.56
Training			95.76	97.08/94.92	0.55	0.74

Table 7: Summary of the performance of New Modified VGG16 Network based on 5 random test/train split of ratio 70 : 30.

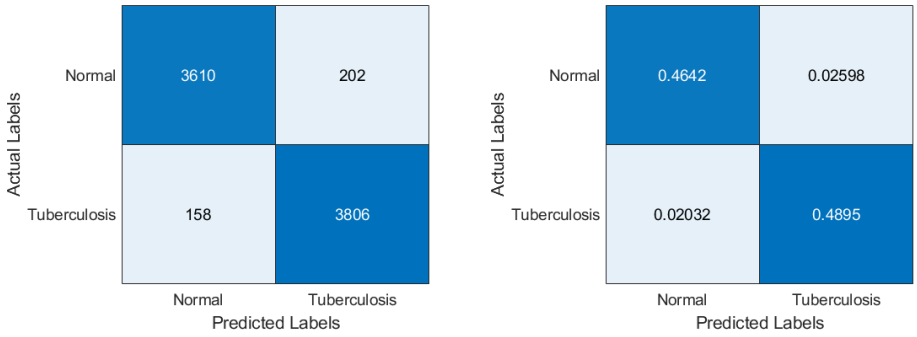

Figure 25: Training confusion and classification matrices for Modified VGG16 Network.

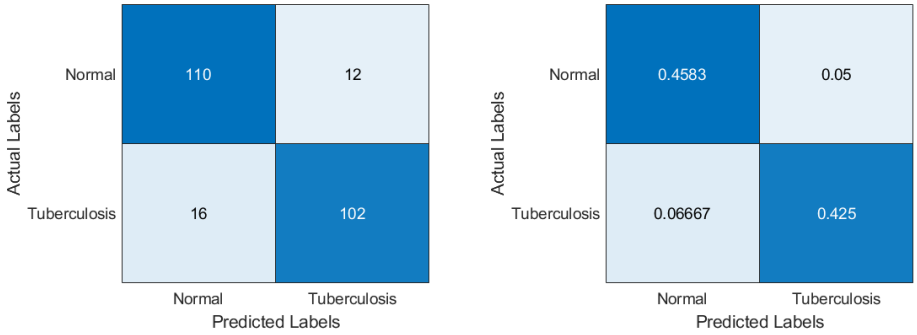

Figure 26: Testing confusion and classification matrices for Modified VGG16 Network.

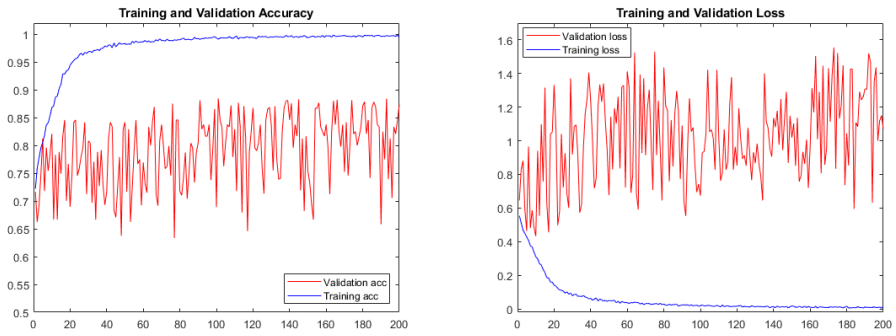

Figure 27: The training and validation accuracy & loss using Modified VGG16 Network.

7.3.1 ROC and Precision Recall Curve

ROC (Receiver Operating Characteristic) curve is an illustration of how any predictive model can distinguish between the true positives and negatives. In order to do this, a model needs to not only correctly predict a positive as a positive, but also a negative as a negative. The ROC curve does this by plotting sensitivity, the probability of predicting a real positive will be a positive, against 1-specificity, the probability of predicting a real negative will be a positive. Figure 28(a) depicts the ROC curve for the proposed network on the testing data. Here, it can be observed that the network achieves a slightly higher false positive rate than the true positive rate, suggesting the need for further improvement.

Precision is a ratio of the number of true positives divided by the sum of the true positives and false positives. It describes how good a model is at predicting the positive class. Precision is referred to as the positive predictive value. Recall is calculated as the ratio of the number of true positives divided by the sum of the true positives and the false negatives. Recall is the same as sensitivity. Reviewing both precision and recall is useful in cases where there is an imbalance in the observations between the two classes. Figure 28(b) depicts the precision curve. Here, it can be observed that the network achieves a similar precision and recall suggesting the network's overall good performance.

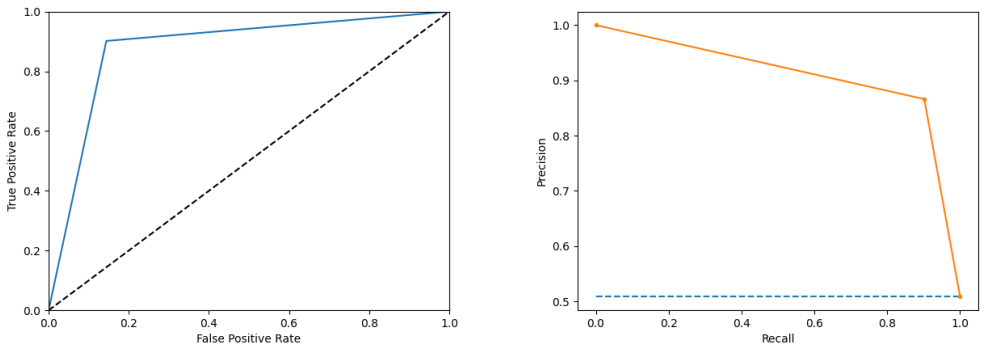

Figure 28: ROC & Precision-Recall Curve for Modified VGG16 Network on testing data.

7.3.2 Saliency Map/Heat Map

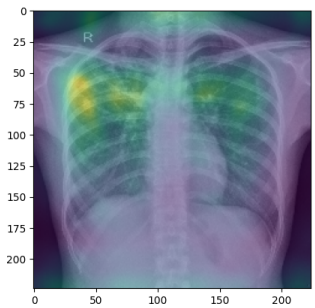

Figure 29: Saliency Map of New Modified VGG16 Network.

In this section, we present the saliency map, a form of visualization technique that helps us understand the network and is also useful as an approximate visual diagnosis for presentation to radiologists. Saliency maps generate a heatmap that shows which region of the image weights more for the classification. Figure 29 illustrates the saliency map of the new proposed network.

7.3.3 Filter Visualization

In this section, we visualize the filters present in each of the nine convolutional layer for the new modified VGG16 model depicted in the architecture experimented in this project. The visualization for each filter corresponding to a convolutional layer have been depicted using a single image. From the figure, it can be observed that the filters in each of the convolutional layers learn specific features from the chest x-ray images relevant to the classification task. Figures 30, 31, 32, 33 and 34 depicts the filter visualization for each of the convolutional layers.

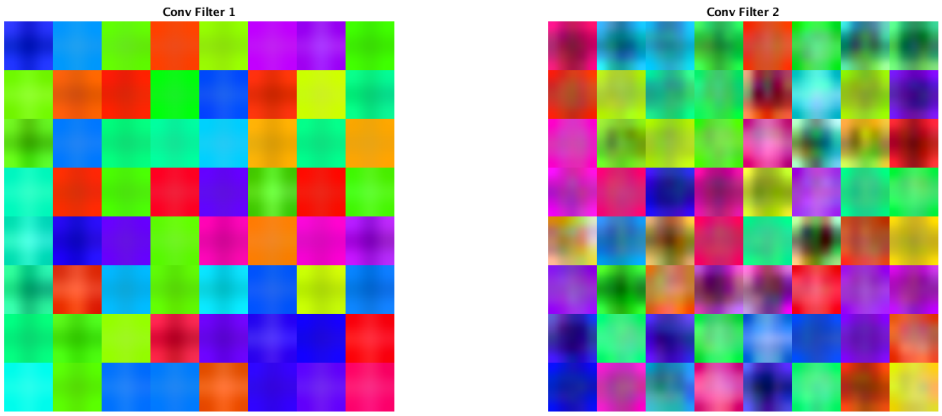

Figure 30: Filters corresponding to convolutional layer's 1 and 2.

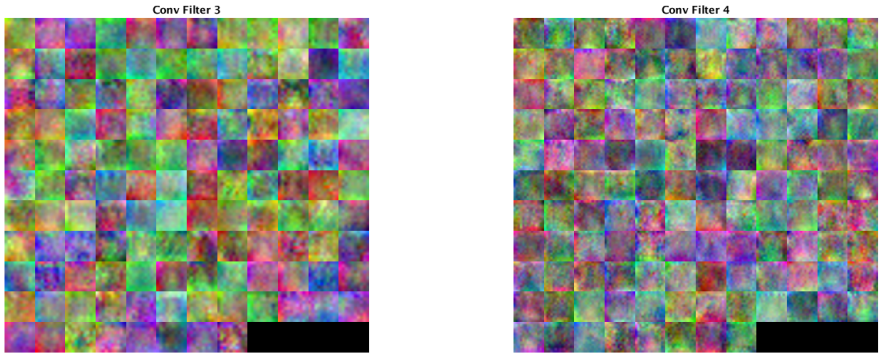

Figure 31: Filters corresponding to convolutional layer's 3 and 4.

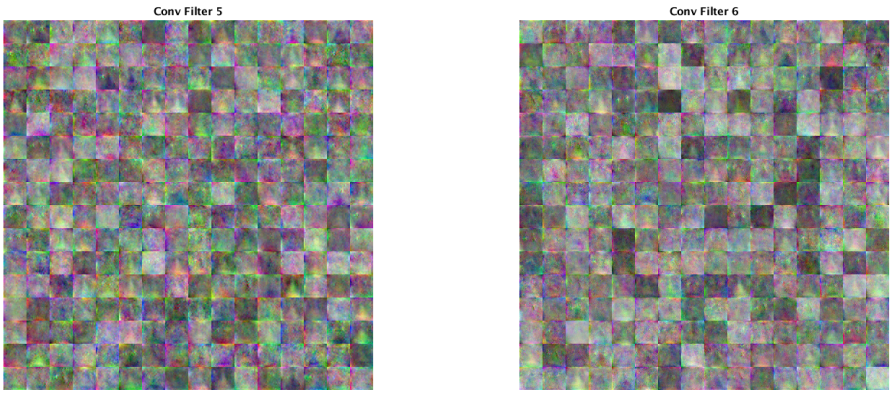

Figure 32: Filters corresponding to convolutional layer's 5 and 6.

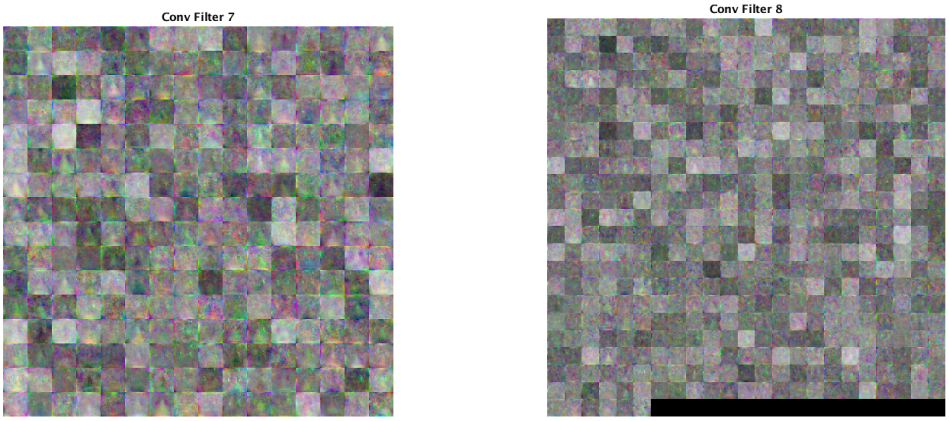

Figure 33: Filters corresponding to convolutional layer's 7 and 8.

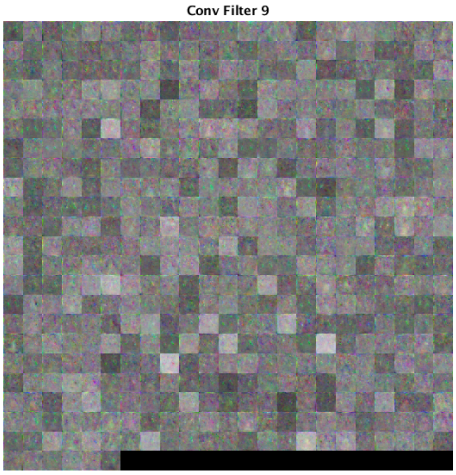

Figure 34: Filters corresponding to convolutional layer 9.

7.3.4 t-SNE Visualization

T-distributed Stochastic Neighbor Embedding (t-SNE) is a machine learning feature reduction algorithm for visualization applied on the convolutional layer and fully connected layer activations of new modified VGG16 network trained on the TB data. We present the results of t-SNE for testing activations of the proposed network. From the figures, we can observe the modeling of a high-dimensional object by a two-dimensional point in such a way that similar objects are modeled by nearby points and dis-similar objects are modeled by distant points with high probability. Figures 35, 36, 37, 38, 39, 40 and 41 represents the t-SNE from each of the layers of the modified VGG16 network considering testing data. From the figure 40 which illustrates the last fully-connected layer, it can be observed that the inter-class variance is high and, as a result, the separation between the groups is quite distinct. This justifies the high classification accuracy of the system.

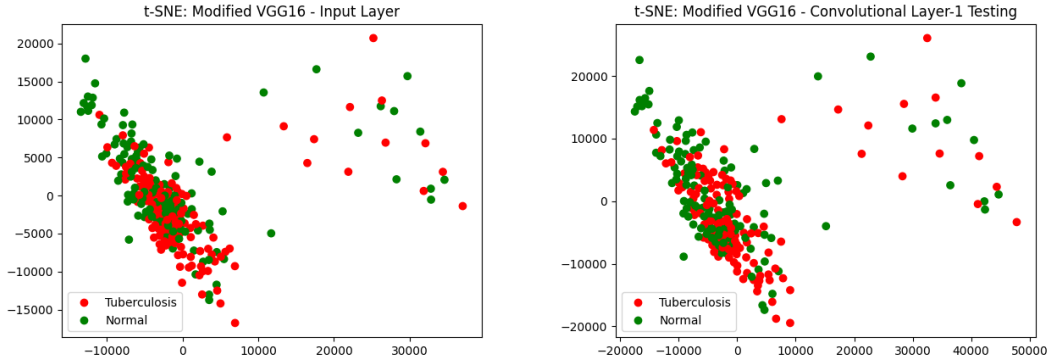

Figure 35: t-SNE visualization of new modified VGG16 network testing for input layer and convolutional layer 1.

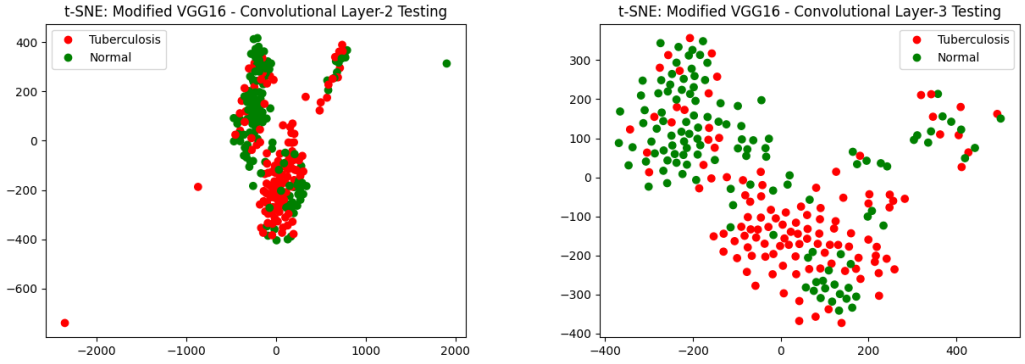

Figure 36: t-SNE visualization of new modified VGG16 network testing for convolutional layer's 2 and 3.

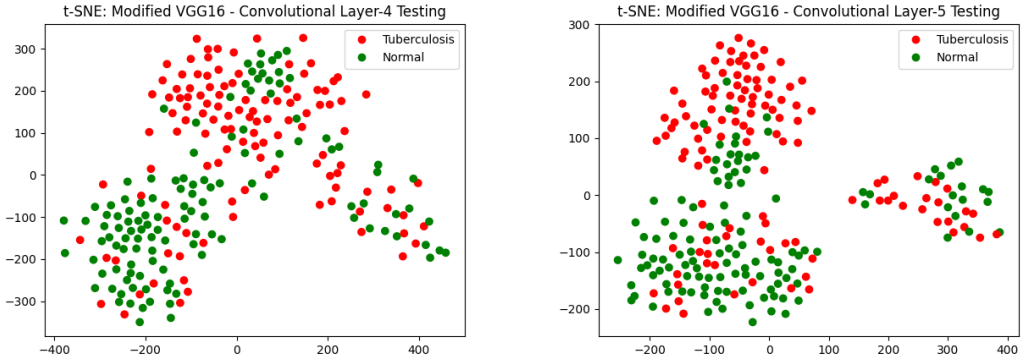

Figure 37: t-SNE visualization of new modified VGG16 network testing for convolutional layer's 4 and 5.

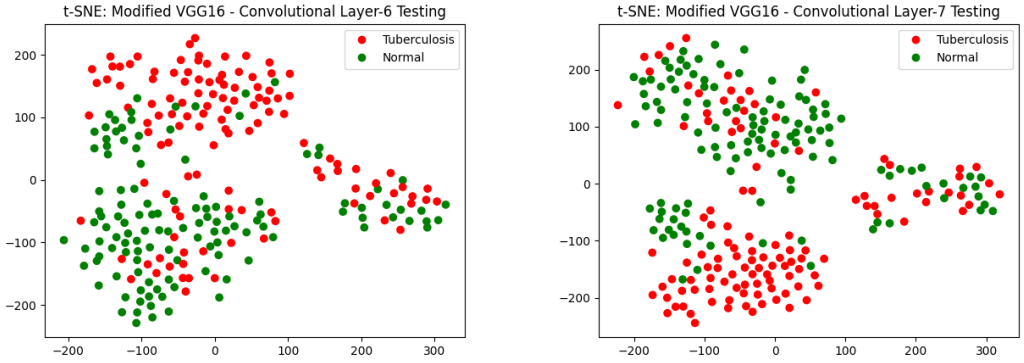

Figure 38: t-SNE visualization of new modified VGG16 network testing for convolutional layer's 6 and 7.

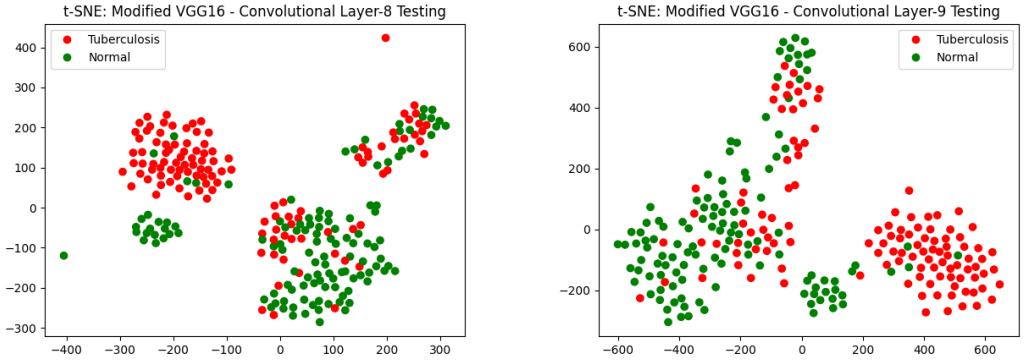

Figure 39: t-SNE visualization of new modified VGG16 network testing for convolutional layer's 8 and 9.

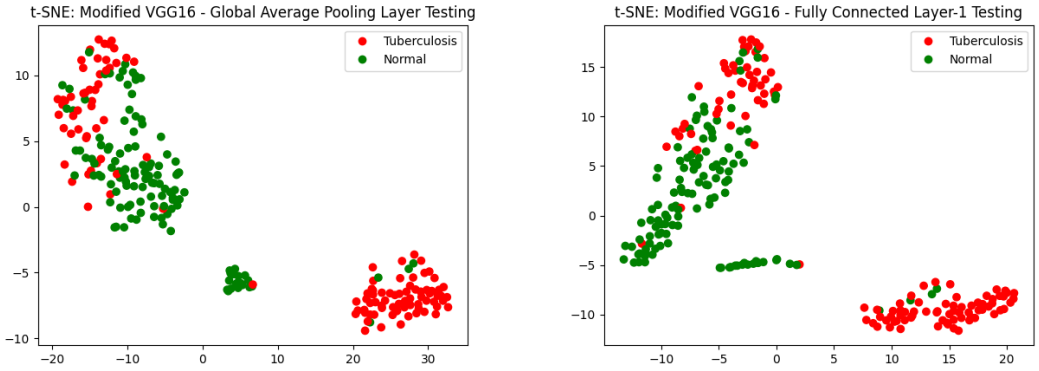

Figure 40: t-SNE visualization of new modified VGG16 network testing for global average pooling layer and fully-connected layer.

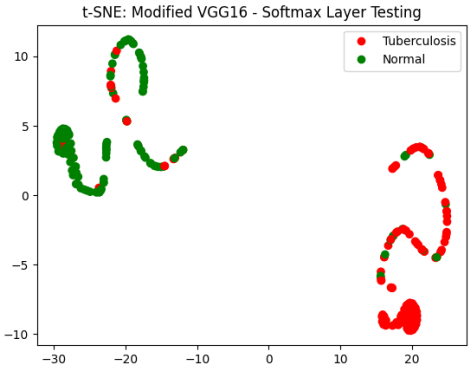

Figure 41: t-SNE visualization of new modified VGG16 network testing for classification layer.

7.3.5 Comparison between baseline, ladder network and new modified VGG16 network based on t-SNE

Figure 42 represents the t-SNE from the last fully connected layer of the baseline model considering VGG16 and ResNet 50 pre-trained weights. On comparision of data separation by the modified VGG16 network in the last fully connected layer (figure 40) with the baseline model (figure 42) it can be observed that the inter-class variance of the proposed network is higher in comparison to the baseline model and, as a result, the separation between the groups is quite distinct and better. This justifies the high classification accuracy of the proposed network than the baseline model.

In addition from figure 43, we can observe the data separation by the ladder network in the last fully connected layer. It can be perceived that the inter-class variance of the ladder network is poor in comparison to both the baseline model and the new proposed network and, as a result, the separation between the groups is quite poor.

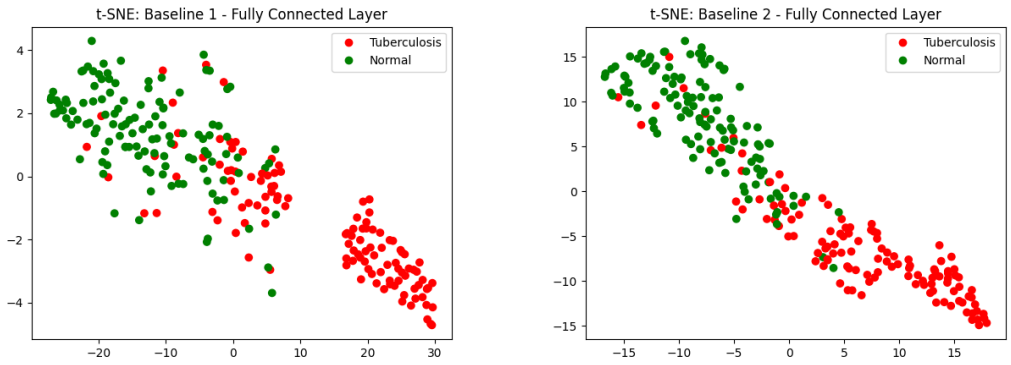

Figure 42: t-SNE visualization of the baseline model with pre-trained weights: VGG16 and ResNet50 considering fully-connected layer for testing data.

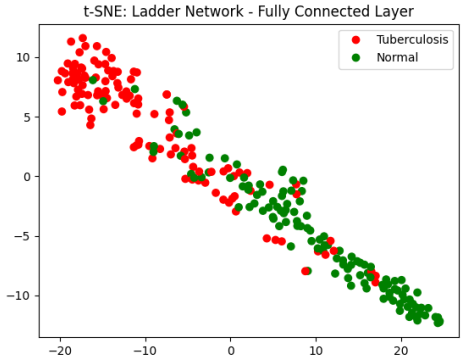

Figure 43: t-SNE visualization of ladder network considering fully-connected layer for testing data.

8 Surprises and what have we learnt from this project

The main surprise in this project was concerning the ladder network, i.e. it was expected to perform well on the binary classification task, but it resulted in poor performance. However, the current results are promising to improve on.

There are several aspects that we learnt from the project, which includes:

1. Data augmentation improves the ability of the network to generalize better.
2. Transfer learning is an effective tool that converges quickly and mostly gives accurate results.
3. Visualization of convolutional neural network functioning with t-SNE and filter visualization.
4. Networks as simple as a few convolutional layers can capture good amount of information, it is all about whether the right features are being picked from the filter or not.

5. Network hyper-parameters pruning to achieve better performance.

9 Conclusion and Future Work

9.1 Conclusion

The following conclusions can be drawn from this work:

1. Tuberculosis (TB) is one of the most commonly existing diseases in mankind and chest x-rays are generally used for examination. In this project, we solve the problem of TB detection with chest x-rays using a new convolutional neural network (CNN) architecture based on supervised learning with two classes normal and TB-infested chest x-rays.

2. The new convolutional neural network (CNN) is based on the existing VGG16 CNN model known for its good performance with modifications to the number of convolutional layers, inclusion of: batch-normalization, dropout and global average pooling (GAP). Hence, we call the new proposed network as the new modified VGG16 CNN model.

3. However, since the CNNs require large amount of data for training. Hence, we employ data-augmentation technique to generate more labelled data by flipping, rotating, etc, along with image pre-processing like black border removal, centering & whitening, etc, to refine the input images.

4. In addition, we implement the ladder networks which is based on a semi-supervised technique for the classification task. The ladder networks combine the feed-forward networks with the encoder-decoder structure and train on the data with a aim to reduce the overall cost (cross-entropy: supervised technique, RMS error: unsupervised technique).

5. In comparison to the new modified VGG16 network outperforms the baseline model with pre-trained weights and the ladder network in terms of overall accuracy and sensitivity. The new model also consistency in performance across multiple random splits of data. However, the ladder network fails to outperform the baseline model suggesting need for improvement.

6. To understand the performance of the new proposed model we investigate the t-SNE pre-processing tool and visualize the data separation for each layer of the network using this tool. In addition, we present the filter visualization to illustrate the features captured by each layer. Lastly, we also present the ROC and Precision-Recall curve to understand the sensitivity and specificity of the trained model.

9.2 Future Work

1. Fine-tune the ladder networks further to achieve better results such that it can be deployed into real world testing.

2. Replace fully-connected layers in the ladder network (feed-forward path) with convolutional layers to extract relevant features from images.

3. Fine-tune the proposed new modified VGG16 convolutional network further to improve its overall performancce, particularly its sensitivity.

4. Data Augmentation with noise injection other than the usual flipping, rotation, translation, etc.

5. Implement bio-medical image pre-processing techniques to further enhance performance.

10 The timeline

The following describes the timeline taken to complete the project.

Week	Info
Week 1 (3/23-3/27):	Data preparation, augmentation and pre-processing of the dataset.
Week 2 (3/30–4/03):	Ready the baseline models and check the results.
Week 3 (4/06–4/10):	Implement ladder networks and fine tune it.
Week 4 (4/13–4/17):	Implement new modified VGG16 network and fine tune it.
Week 4 (4/13–4/17):	Visualize the filters and data distribution in each layers using t-SNE.
Week 5 (4/20–4/24):	Consolidation of all experiments and report.
Week 6 (4/27–4/30):	Project Due!

References

[1] W. H. Organization, *Global Tuberculosis Control: Epidemiology, Strategy, Financing: WHO Report 2009*. World Health Organization, 2009. 3

[2] S. S. Meraj, R. Yaakob, A. Azman, S. N. M. Rum, A. Shahrel, A. Nazri, and N. F. Zakaria, "Detection of Pulmonary Tuberculosis Manifestation in Chest X-rays Using Different Convolutional Neural network (CNN) Models." 3, 4, 5, 8, 9

[3] F. Pasa, V. Golkov, F. Pfeiffer, D. Cremers, and D. Pfeiffer, "Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization," *Scientific reports*, vol. 9, no. 1, pp. 1–9, 2019. 5

[4] E. Tian and P. Ocampo, "Exploring the efficacy of using a neural network trained on non-tuberculosis chest x-rays for detecting tuberculosis," *Stanford Deep Learning Course (CS230) Project*, 2019.

[5] S. Lopez-Garnier, P. Sheen, and M. Zimic, "Automatic diagnostics of tuberculosis using convolutional neural networks analysis of MODS digital images," *PloS one*, vol. 14, no. 2, 2019. 3, 4

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 3, 5, 10, 13

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015. 3, 5

[8] A. Rasmus, M. Berglund, M. Honkala, H. Valpola, and T. Raiko, "Semi-supervised learning with ladder networks," in *Advances in neural information processing systems*, 2015, pp. 3546–3554. 3, 5

[9] S. Jaeger, S. Candemir, S. Antani, Y.-X. J. Wáng, P.-X. Lu, and G. Thoma, "Two public chest x-ray datasets for computer-aided screening of pulmonary diseases," *Quantitative Imaging in Medicine and Surgery*, vol. 4, no. 6, pp. 475–477, 2014. 4, 5

[10] Y. LeCun, C. Cortes, and C. J. Burges, "The MNIST database of handwritten digits, 1998," *URL http://yann.lecun.com/exdb/mnist*, vol. 10, p. 34, 1998. 5

[11] F. Chollet *et al.*, "Keras," https://keras.io, 2015. 9

[12] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 10, 13, 16, 20, 22

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 11

[14] G. Ye, "Sign language translation using ladder networks," *Stanford Deep Learning Course (CS230) Project*, 2019.

[15] U. Lopes and J. F. Valiati, "Pre-trained convolutional neural networks as feature extractors for tuberculosis detection," *Computers in biology and medicine*, vol. 89, pp. 135–143, 2017.