Stacking a Model: Leveraging a Vision Transformer and CNNs for an Accurate Tuberculosis Classification

Student Details

Tae yong Kwon

Student Number: **U867693**

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE IN COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE DEPARTMENT OF COGNITIVE SCIENCE & ARTIFICIAL INTELLIGENCE SCHOOL OF HUMANITIES AND DIGITAL SCIENCES TILBURG UNIVERSITY

STUDENT NUMBER

U867693

COMMITTEE

Prof. dr. Eric Postma

Dr. Phillip Brown

LOCATION

Tilburg University

School of Humanities and Digital Sciences

Department of Cognitive Science & Artificial Intelligence

Tilburg, The Netherlands

DATE

June 24th, 2024

WORD COUNT

6579

Acknowledgement

I would like to express my gratitude to Prof. Dr. Eric Postma for his support and insightful guidance throughout this thesis semester. His emphasis on key points and pragmatic advice helped me a lot to keep me focused and be on track.

I extend my thanks to my parents and my little brother for their unwavering support over the years. They stood by me when during the uncertain times after the high school. They watched over me during my military service. They encouraged me when I decided to change my field of study out of the blue. Their constant presence and encouragement always have been a source of energy for me, for which I am deeply appreciative.

I also thank all my close friends for giving me inspiration. I am lucky to have such intelligent and talented people around.

Contents

Stacking a Model: Leveraging a Vision transformer and CNNs for an Accurate Tuberculosis Classification

Tae Yong Kwon

Abstract

This thesis investigates the performance of ensemble deep learning Convolutional Neural Network (CNN) models and vision transformers in the early detection of Tuberculosis (TB), a leading cause of death in developing countries. Despite advances in diagnostic technologies, the specificity of traditional methods such as chest X-rays remains a challenge (Cudahy & Shenoi, 2016). This study aims to enhance diagnostic performance through an ensemble model combining pre-trained networks: MobileNetV3, VGG-16, and ViT-B16, chosen for their diversity in architecture and robust performance.

The distinctive approach of this study lies in analyzing the increase in performance metric by stacking visual transformer on top of a CNN ensemble. Previous studies on ensemble learning for TB detection have typically focused either on CNN ensembles or CNNs combined with transformers. This research goes a little further, examining the difference in performance metrics when a vision transformer is added to a CNN ensemble, from a standalone CNN ensemble.

Preliminary findings indicate that the both ensemble model achieves superior accuracy from the baseline models, except from the VGG-16. Vision transformer + CNN ensemble has achieved a significant performance metrics improvement from the CNN ensemble. Further work could include application of the developed ensemble model framework to other diseases with similar diagnostic challenges such as pneumonia, to evaluate its versatility and potential for broader medical applications.

DATA SOURCE, ETHICS, CODE AND TECHNOLOGY STATEMENT

As per the data owner's request, I include this statement: "Data were obtained from the TB Portals (https://tbportals.niaid.nih.gov), which is an open-access TB data resource supported by the National Institute of Allergy and Infectious Diseases (NIAID) Office of Cyber Infrastructure and Computational Biology (OCICB) in Bethesda, MD. These data were collected and submitted by members of the TB Portals Consortium (https://tbportals.niaid.nih.gov/Partners). Investigators and other data contributors that originally submitted the data to the TB Portals did not participate in the design or analysis of this study." The obtained data is anonymized. Work on this thesis did not involve collecting data from human participants or animals. The original owner of the data used in this thesis retains ownership of the data during and after the completion of this thesis. However, the NIAID (Contactperson: Eric Boiter) was informed about the use of this data for this thesis and potential research publications. All the figures belong to the author. The thesis code can be accessed through the GitHub repository following the link (rnjsxodyd90/ensemble-model-for-tuberculosisVGG-Mobile-Vit (github.com). Part of the

CODE has been adapted by the author from (Balance Data with Augment Images F1score=88% (kaggle.com), Image classification with Vision Transformer (keras.io) , https://www.kaggle.com/code/ebrahimelgazar/vision-transformer-vit-keras-pretrained-models licensed under a Apache 2.0 open source). The reused/ adapted code fragments are clearly indicated in the notebook. In terms of writing, the author used assistance with the language of the paper. A language model Grammarly(https://www.grammarly.com) was used to improve the author 's original content, for paraphrasing, spell checking and grammar. No other typesetting tools or services were used.

1. Introduction

This thesis investigates whether an ensemble model combining CNNs and Vision Transformers (ViTs) enhances performance metrics compared to ensembles solely based on CNNs for the classification of TB images. It applies stacking ensemble technique to these baseline architectures to determine their effectiveness in the given context.

Traditionally, CNNs have dominated image classification tasks, including TB and COVID-19 detection from X-rays and CT scans (Sarvamangala & Kulkarni, 2021). However, the newly imported and adapted transformer architecture from NLP has been demonstrating supreme proficiency, even comparable and surpassing the accuracy of CNN models from the start of 2020s (Zhang et al., 2023). ViTs have been extensively studied in numerous journals. However, it's still unclear if an ensemble model combining ViTs and CNNs can outperform a CNN-only ensemble in performance metrics.

From a practical perspective, using CNN and ensemble methods can significantly speed up TB diagnosis by reducing the reliance on radiologists to manually review each image.

From a scientific perspective, comparing the performance of ensemble models that combine CNNs and ViTs against those using only CNNs provides valuable insights into the correlation between the combination of model architecture and model performance. Such comparisons can demonstrate the benefits of integrating different model architectures, potentially leading to better generalization to unseen data. This approach may establish new benchmarks for ensemble techniques in radiographic imaging, helping to identify best practices and paving the way for future advancements in radiography diagnostics.

This is the research question based on this context:

> *Does an stacked ensemble model combining CNNs(mobilenetv3, VGG-16) and ViTB-16 outperform an ensemble model comprised solely of CNNs in terms of performance metrics for the classification of TB?*

The main findings are as follows:

- CNN+ViT ensemble showed a major improvement in accuracy and other performance metrics compared to CNN ensemble.
- Stacking resulted in a minor improvement in performance metrics for both ensembles, compared to the baseline models.
- CNN+ViT ensemble achieved a well-round performance, suitable for the medical image analysis context.

2. Related work

The related work section will first explore the work done to classify lung images with individual CNNs and transformers in the realm of radiography images. The works using ensemble model will be then explored.

Research on classifying lung images with individual Convolutional Neural Networks (CNNs) and vision transformers has shown significant promise in the field. For instance, (Almezhghwi et al., 2021) utilized a support vector machine on top of AlexNet and VGG16, resulting in an impressive area under the curve (AUC) value of 98% and 97%, respectively. Their proposed method surpassed the performance of both individual classifiers, highlighting the potential of combining machine learning techniques with CNNs for improved diagnostic accuracy. However, this approach may be limited by its reliance on handcrafted features, which may not fully capture the complexity of medical images.

Similarly, (Pradhan et al., 2022) focused on optimizing CNN performance through a hill climbing algorithm, achieving an accuracy of 86.66% from a small dataset of just 120 images. This work underscores the importance of optimization techniques in enhancing model performance, though the small sample size raises questions about the model's generalizability to larger, more diverse datasets.

Furthermore, (Nahiduzzaman et al., 2023) developed a framework that successfully classified 17 different lung disorders using an extreme machine learning algorithm (ELM) on top of a CNN, achieving over 90% accuracy. This study demonstrated the capability of CNNs to handle multi-class classification tasks effectively.

2.1 Ensemble learning

The concept of ensemble learning was first proposed by (Nilsson,1965). The process involves combining various baseline models to build a stronger single output. Ensemble learning can be divided into two distinct fusion methods: weighted methods and meta-learning methods. Weighted methods involve allowing different weights for base learners, while meta-learning methods use the meta-knowledge learnt from the base learners (Yang et al., 2022). Stacking involved in this paper is a main example of meta-learning ensemble method. There are also other methods than the two methods mentioned.

Ensemble learning has been extensively utilized in the context of medical image analysis, particularly with chest X-rays (CXR). In one study, the authors fused Swin Transformer and convolutional layers at the architectural level to create a network that demonstrated improved efficiency in lung segmentation of the CXR (Liu et al., 2022). Like this study, another study combined the ViT with MobileNetV2 and DenseNet169 at the functional layer level, feeding the output into a global pooling layer to predict pneumonia from CXR images, achieving an accuracy of 93.91% (Mabrouk et al., 2022). Only difference with this study was that they have utilized a MobileNetV2, a previous version of MobileNetV3Large that does not incorporate the adaptation for large datasets.

As for the weighted voting, it is commonly used in image classification applications (Ennouni et al., 2021). In a majority voting-based ensemble for instance, each base learner throws a vote for the class based on its prediction, and the majority class is chosen as the collective prediction of the ensemble. However, majority voting ensembles have significant drawbacks, including bias

towards weaker baseline learners, issues with generalization, and difficulties in training. I am not using the idea of voting for this reason.

As for meta learning, (Ali et al., 2023) explores the breast cancer classification utilizing meta-learning approach. They gathered the output of prediction from three different machine learning models, and ensembles them with a logistic regression to create a meta learner – which has successfully countered the problem of issue of overfitting shown in other studies and achieved an overall accuracy of 87.9% in classifying breast cancer. This is also the reason why a stacking is utilized in this study. Although the referenced study used InceptionV3, densenet121 and Resnet50, the other pretrained models famous for their performance will be used – since there is no significant variation in the architecture. Also, would be explored later, but variation in architecture determines the success of the ensemble – since the homogeneous ensembles might miss a pattern in the data (Dutschmann et al., 2023)

The use of ensemble methods is not limited to the meta-learning and weighted methods. There are different methods. For instance, there have been efforts to employ fuzzy integral-based ensemble methods purely on CNN classifiers, including DenseNet121, VGG19, and ResNet50, to classify tuberculosis (Dey et al., 2022). On the other paper, (Kalaivani & Seetharaman, 2022) proposed a three-stage ensemble model, where in the first stage the CXR images are segmented with the conventional UNET model, CNN used then to extract features from the second stage and combined by voting in the third stage. This method achieved the accuracy of 99.35%. These diverse approaches highlight the versatility and effectiveness of ensemble learning in enhancing model performance in medical image analysis.

EfficientNet variants have become very popular in the current scene of ensemble learning. For instance, Ravi et al. (2022) utilized features from EfficientNet models (B0, B1, B2) and combined them using a stacked ensemble classifier for COVID-19 detection, achieving an overall accuracy of 98%. Similarly, Oloko-Oba and Viriri (2021) used UNET segmentation to preprocess CXR images and extracted features with different EfficientNet variants (B2, B3, B4). These features were then combined using the bagging method, resulting in a final accuracy of 97.44%. EfficientNet ensembles have also been applied successfully in other fields, such as the identification of Ethiopian plant species by Kiflie et al. (2024), demonstrating the broad applicability and success of EfficientNet-based ensembles.

3. Methods

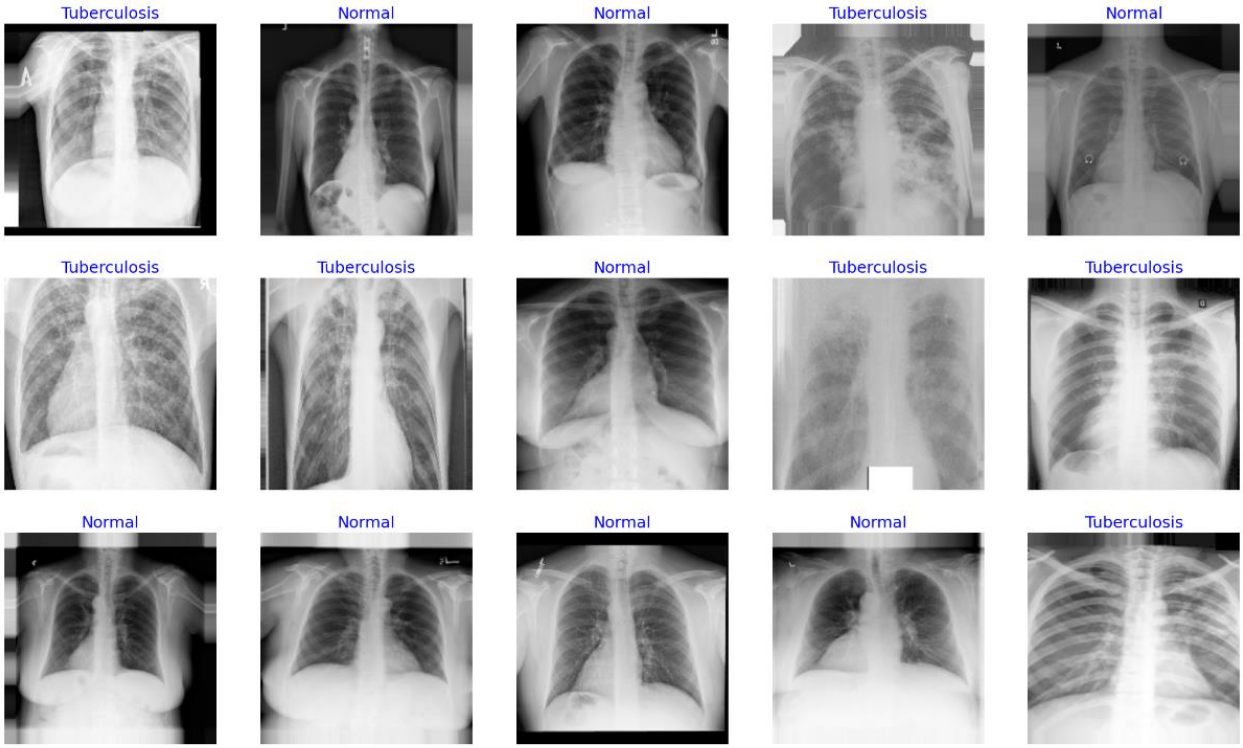
3.1 Dataset

The TB image dataset used in this study is sourced from TB Portals, a program bolstered by the US National Institute of Allergy and Infectious Diseases. As of May 2024, TB Portals provides a total of 13,025 CT and CXR images of TB-infected lungs, including 8,000 CXR images exclusively depicting TB infection (TB Portals Program, 2017; Rosenthal et al., 2017). All images are in DICOM format and include both color and greyscale versions. Additionally, the dataset includes a comprehensive CSV file containing patient metadata such as sex, age of onset, drugs used, outcomes, diagnosis codes, and more. The data originates from various sources, including clinical trials, research studies, and routine treatments from countries such as India, Belarus, and Romania.

For normal CXR images, a separate dataset consisting of 3,500 images is sourced from Kaggle (Chowdhury et al., 2020). These images were originally provided in PNG (Portable Network Graphics) format. Some of these images were originally part of the RSNA CXR dataset, which was introduced for the pneumonia detection challenge.

Figure 1

Image Sample Showing Normal And TB CXRs Used In This Study



3.2 Preprocessing

The original images in our tuberculosis (TB) dataset were in DICOM format, which posed a challenge due to their large size and bulky metadata, making them impractical for processing within system RAM. To address this, I preprocessed and converted these images to PNG format, ensuring consistency with the normal chest X-ray (CXR) images in our dataset.

Balancing the Dataset

To prevent issues associated with an imbalanced dataset, I downsampled the TB images to match the number of normal CXR images. This approach ensured that the model does not overfit to the majority class of "Normal" and avoid skewed model training (Wendler & Gröttrup, 2021). Subsequently, the entire dataset was randomly shuffled to ensure a uniform distribution across the training, validation, and test sets – to help avoid bias from inherent ordering from the data (Xu & Goodacre, 2018). I adopted a 60/20/20 split for these sets, as mentioned in the previous literature. This method was implemented to allocate a robust training set, while the validation and test sets remained large enough for meaningful performance assessment.

Normalization and Augmentation

The pixel values of all images, ranging from 0 to 255, were normalized to the range from 0 to 1. This step is crucial as neural networks generally perform better with normalized inputs, enhancing convergence rates (Bjorck et al., 2018).

I applied the following data augmentation techniques to further generalize on unseen dataset(Z. Yang et al., 2023).

Shear Transformation

With a shear intensity of 0.2, images were distorted along the horizontal or vertical axis. This augmentation aimed to improve the model's invariance to geometric transformations, addressing slight distortions present in many CXR images. Studies have shown that a moderate level of shear transformation can significantly improve model robustness without introducing excessive noise (Shorten & Khoshgoftaar, 2019).

Zoom Augmentation

A zoom factor of 0.2 was applied randomly to the training images, introducing variations in magnification. This technique enhances the model's ability to identify objects presented at various scales, which is particularly important given the varied zoom levels in CXR images provided by the NIAID, some focusing on the lungs and others showing the entire chest area (He et al., 2016).

Horizontal Flipping

This augmentation involved flipping images horizontally to teach the model that the orientation of the X-ray image does not affect its classification. By exposing the model to both original and flipped images, I improved its ability to accurately classify images regardless of their orientation (Elgendi et al., 2021).

3.3 Performance metrics

In the domain of classifier evaluation, several performance metrics are commonly used to assess the effectiveness of a model. According to (Yang & Pedersen, 1997), these metrics include accuracy, precision, recall, F1 score, and the confusion matrix. Each metric provides different insight into the model's performance, and together, they offer a comprehensive evaluation – which will be explored in the discussion section.

Accuracy: The ratio of correctly classified instances over the overall number of instances.

Precision: How many times the positive class is incorrectly predicted as being negative, or vice versa.

Recall: Measures how well the model predicts True positives.

F1 Score: Harmonic mean of recall and precision

Equation 1

Performance Metrics

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

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

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

$$F1\ Score = \frac{2*TP}{TP+FN+2*TP}$$

TP – True positive

FP – False positive

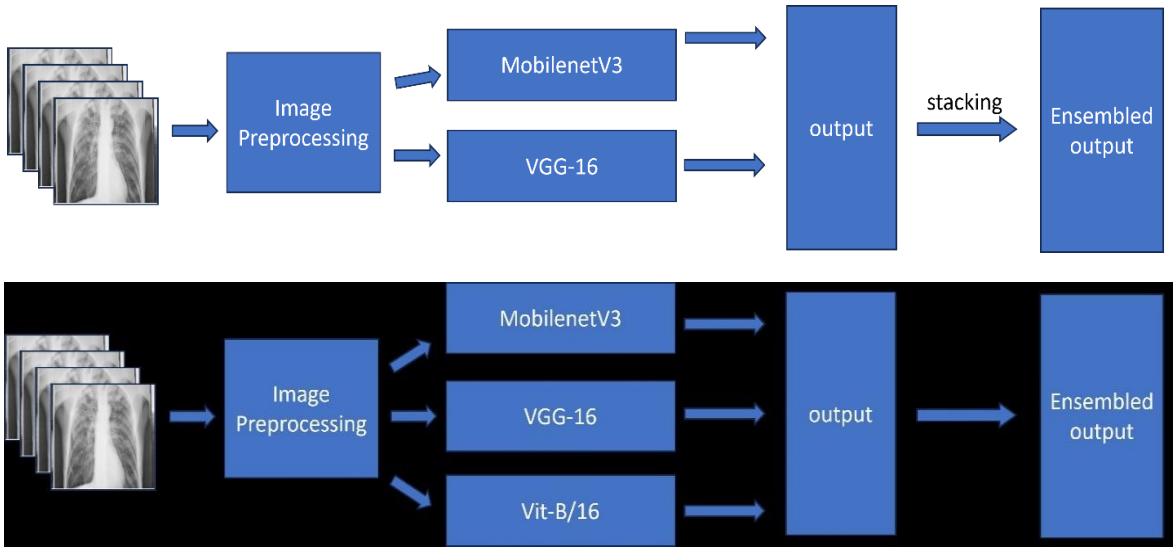
FN – False negative

TN – True negative

3.3 Methodology

Figure 2

Constructing the ensemble model - Methodology



Software used:

Python was used as a main software. For numerical operations and data manipulation, Numpy and pandas were utilized. Numpy supports large, multi-dimensional arrays and matrices (Harris et al., 2020), while Pandas offers powerful data analysis tools (McKinney, 2010). Visualization was facilitated by matplotlib and seaborn. Matplotlib was also used for graphing (Hunter, 2007), and Seaborn was used to visualize a heatmap (Waskom et al., 2021). Computer vision tasks were handled by cv2 (OpenCV), a comprehensive library for computer vision and machine learning (Brad ski, 2000).

Machine learning and evaluation metrics were supported by sklearn, which offers tools for building and validating machine learning models (Pedregosa et al., 2011). For building models, tensorflow and keras were utilized. TensorFlow is an open-source library for machine learning (Abadi et al., 2015), and Keras is an API running on top of TensorFlow, simplifying the creation and training of neural networks (Chollet, 2018).

The transformers library was also utilized. (Wolf et al., 2020).

Hardware used : GTX 1660 Super with 12 GB memory for training and testing the models on local, GPU on Google Collab was also utilized (NVIDIA Tesla K80 12GB memory), Google Drive with 200GB extra storage space was used.

Method:

I focused on creating two ensembles to compare their performance: one combining MobileNetV3 and VGG-16, and the other adding Vision Transformer (ViT) to the mix. These ensembles were evaluated against baseline models using a stacking technique, where a logistic regression model served as the meta-learner.

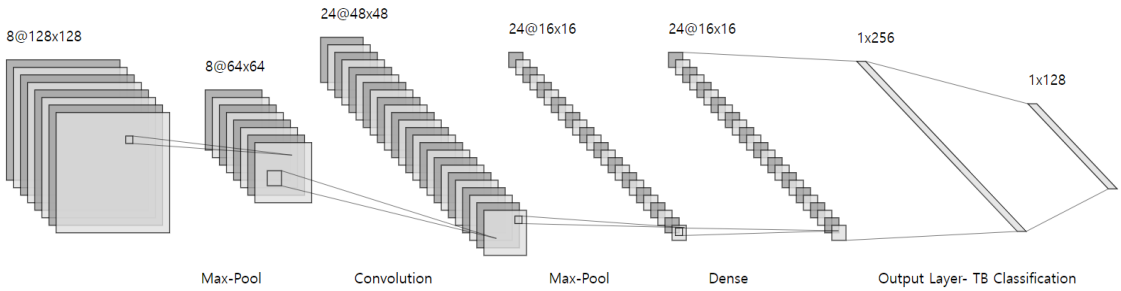
First, I downsampled and balanced the dataset, and split it into training, validation, and test sets. Next, I normalized the data to speed up model convergence and applied data augmentation techniques to enhance model robustness. The ensembles were then trained, and their performance metrics were compared.

In the following subsections, I dive into the specifics of each baseline model, explaining their architecture and their roles within the experiment.

3.4 Elements of CNN

Figure 3

Basic CNN Structure



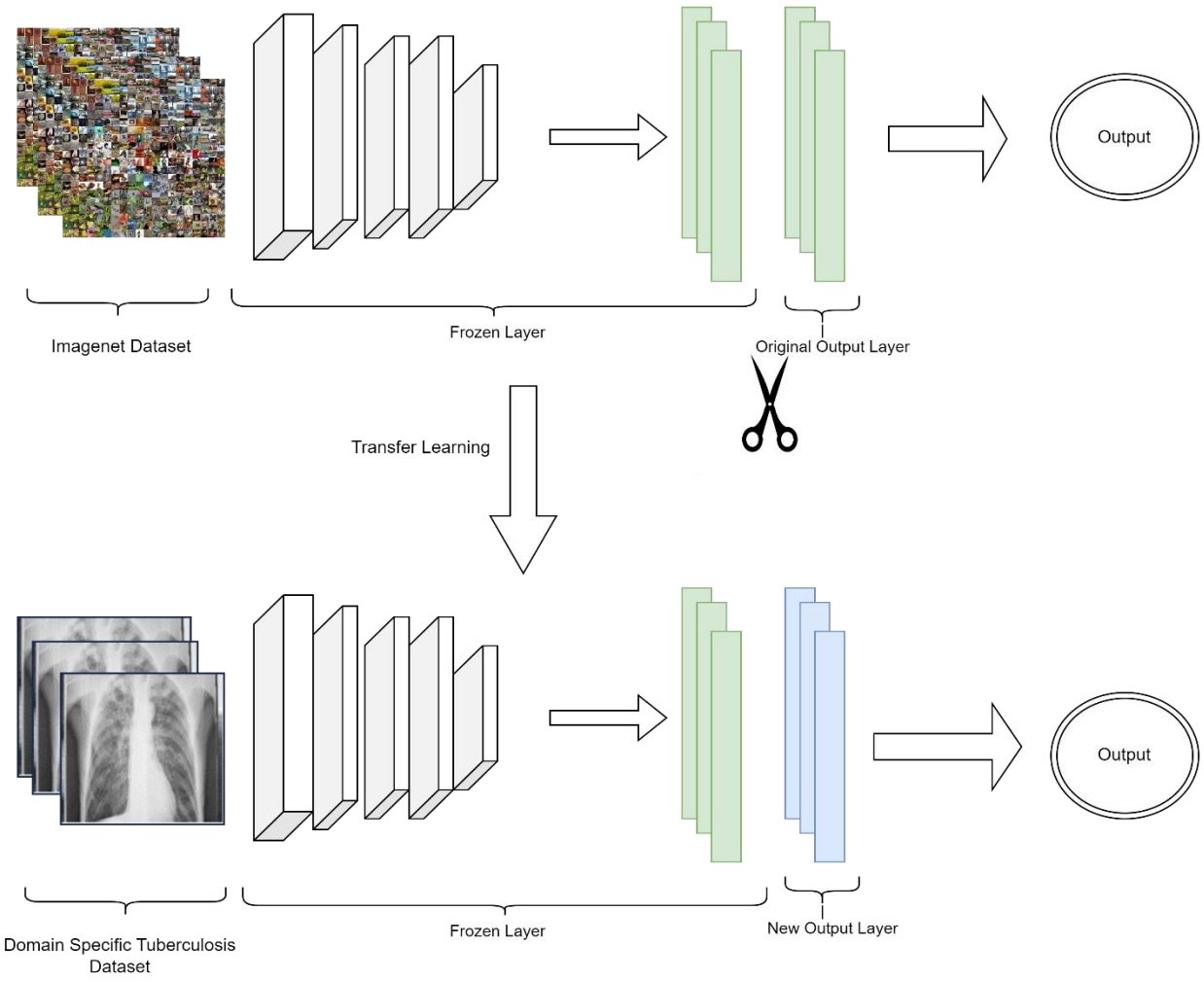
CNNs was first proposed by Alex Krizhevsky. It consists of a substantial neural network architecture - with 5 different convolutional layers followed by max pooling layers and three fully connected layers. The initial two layers of a CNN, the convolution and pooling layers, are dedicated to feature extraction. The subsequent fully connected layer utilizes these extracted features to produce final outcomes, such as classifications (Zeiler & Fergus, 2014).

Central to the CNN architecture is the convolution layer. Convolution is a mathematical method of combining two signals to form a third signal. In the context of digital images, which are essentially stored as two-dimensional (2D) grids of pixel values, a small matrix of parameters known as a kernel is employed across each position of the image (Capobianco et al., 2021). This kernel functions as a tunable feature extractor, enabling CNNs to efficiently process images by detecting features regardless of their position within the image. As each layer outputs to the next, the features being extracted progressively increase in complexity through a hierarchical structure (Yamashita et al., 2018).

3.5 Transfer Learning on CNNs

Figure 4

Transfer Learning of CNN



CNNs trained on large datasets such as ImageNet-21k (Ridnik et al., 2021) learn rich feature representations transferable to other tasks. Pre-trained models such as VGG16 and MobileNetV3, having been trained on millions of images, provide robust feature extraction layers (Oquab et al., 2014).

The fine-tuning process in this study involved taking a pre-trained model of MobilenetV3, VGG-16, ViTB16 and updating its weights on a new dataset – as described in Figure 4. Initially, the general feature-capturing layers (edges, textures) were frozen to retain their pre-learned weights (Weiss et al., 2016). The later layers which capture task-specific features are then fine-tuned on the CXR dataset, allowing the model to adapt to the specifics of the new CXR dataset. These configuration of output layers will be further addressed in the following baseline model sections.

3.6 VGG-16

VGG16, introduced by Karen Simonyan and the Google DeepMind team, marked a significant improvement over the original ConvNet architecture by emphasizing depth. VGG16 comprises 16 layers with trainable parameters, hence the name "VGG16." It utilizes small receptive fields (3x3) and small pixel strides (1), a strategic departure from earlier models that employed larger receptive fields. These smaller convolutional filters help reduce the network's tendency to overfit (Saabni & Schclar, 2020).

In this study, I used a pre-trained VGG16 model as a baseline for tuberculosis classification. To preserve the pre-trained features and avoid overfitting, the base model layers were frozen (Albashish, 2022). The output layer added mirrors that of MobileNetV3 Large. Specifically, this enhancement involved flattening the 3D tensor output from VGG16 into a 1D vector (Jeczmionek & Kowalski, 2021). This vector is then passed through a dense layer with 1024 neurons using ReLU activation and L2 regularization to prevent overfitting. Batch normalization was incorporated to expedite training, followed by a dropout layer with a 50% dropout rate to further mitigate overfitting (Srivastava et al., 2014). This configuration was repeated with a second dense layer of 512 neurons, again followed by batch normalization and dropout. Finally, a SoftMax activation layer was applied to provide class probabilities for classification. These hyperparameters were all tested, as seen in appendix A, and was chosen to have the best performance. Other than downsampling, the categorical cross entropy was used as a loss function to mitigate the effect of the original class imbalance. However, it later turned out to be excessive and giving less optimal training performance – thus changed to binary cross entropy (Aurelio et al., 2019). This will be further examined in the results section.

3.7 MobilenetV3Large

Also developed by Google, MobileNetV3 is designed for efficiency and is particularly suitable for mobile and edge devices with limited computational resources. MobileNetV3 comes in two versions: Small and Large (Howard et al., 2019). This study employs MobileNetV3 Large, which is optimized for high-resource scenarios like tuberculosis classification, involving the processing of over 6000 images.

MobileNetV3 Large was chosen over EfficientNets, which are commonly used in related studies (Oloko-Oba & Viriri, 2021; Marques et al., 2022). The decision is driven by MobileNetV3 Large's unique architecture that includes depthwise separable convolutions, a feature absent in VGG16 and distinct in its implementation from EfficientNet. Studies indicate that combining models with different architectures often leads to improved performance (Dietterich, 2000).

Depthwise separable convolution is a two-step method that starts with a depthwise convolution (spatial convolution applied independently to each input channel), followed by a pointwise convolution (1x1 convolution) (Kaiser et al., 2017). This method significantly reduces the number of parameters and computational cost compared to standard convolutions used in VGG16.

To illustrate, consider a convolutional kernel with a shape of $(k \times k)$ with the input channel c_{in} and output channel c_{out}, respectively.

For a standard convolution, the filter tensor has the following dimension:

$$F \in R^{k \times k \times c_{in} \times c_{out}} \quad (1)$$

Therefore, the number of parameters in the filter tensor is:

$$k \times k \times c_{in} \times c_{out} \quad (2)$$

For the depthwise separable convolution, it splits into depthwise and pointwise convolution as previously mentioned.

Number of parameter in the depthwise filter is:

$$k \times k \times c_{in} \quad (3)$$

Number of parameter in the pointwise filter is:

$$1 \times 1 \times c_{in} \times c_{out} = c_{in} \times c_{out} \quad (4)$$

The total number of parameter in depthwise seperable convolution is the sum of parameters in the pointwise and depthwise convolutions:

$$k \times k \times c_{in} + c_{in} \times c_{out} \quad (5)$$

Therefore, the parameter ratio between depthwise separable convolution and standard convolution is:

$$Ratio = \frac{Number\ of\ parameters\ (depthwise\ seperable\ convolution)}{Number\ of\ parameters\ (VGG/normal\ convolution)} \quad (6)$$

$$= \frac{k \times k \times c_{in} + c_{out} \times c_{in}}{k \times k \times c_{in} \times c_{out}}$$

$$= \frac{k^2 + c_{out}}{k^2 \times c_{out}}$$

$$= \frac{1}{c_{out}} + \frac{1}{k^2}$$

From (6), when an output of convolution is large, we can neglect the $\frac{1}{c_{out}}$ so the ratio could be seen as $\frac{1}{k^2}$.

This implies that for large c_{out}, the parameter number in a standard convolutional layer in VGG16 is approximately k^2 times the parameter number in a depthwise separable convolutional layer in MobileNet. This architectural difference allows for the selection of MobileNetV3 Large to enhance the performance of our ensemble model, which includes computationally intensive networks such as VGG-16 and ViT-B16.

As for the general structure of the output layer in the MobileNet that will be implemented in this study, the input layer is taken from the pre-trained MobileNetV3Large model. The output of MobileNet is passed through a GlobalAveragePooling2D layer to reduce the spatial dimensions, followed by a dense layer with 1024 neurons using ReLU activation (Patel & Wang, 2022).

$$\text{ReLU}(x) = \max(0, x)$$

ReLU activation is chosen because it helps to overcome the vanishing gradient problem, allowing for even faster training (Chen et al., 2020). Since the computationally/time intensive VGG-16 and ViT-B16 was already being used in the experiment, it was chosen to mitigate the time taken.

To further enhance regularization, a dropout layer with a 50% dropout rate is applied, followed by another dense layer with 512 neurons - also using ReLU activation. This is paired with an additional dropout layer to prevent overfitting. The final dense layer aligns with the number of target classes (2) and utilizes a sigmoid activation function (Dubey et al., 2022), optimal for binary classification tasks like one in this study. While this study does not investigate further the mathematical intricacies of the activation functions and regularization methods, the selection of these hyperparameters is aimed at maximizing model performance. The hyperparameters used for tuning can also be found in appendix A.

3.8 ViT-B16

Following its dominant performance in natural language processing, Vision Transformers (ViTs) were introduced by Dosovitskiy et al. (2020) for image classification tasks. ViTs represent a significant shift from CNNs by processing images as sequences of patches, akin to words in a sentence. The base model used in this study is the ViT-B16, which processes images by dividing them into fixed-size patches and treating these patches as sequences of tokens.

Pre-trained on the ImageNet-21k dataset, ViT-B16 provides a robust starting point for transfer learning. Images are preprocessed to a size of 224x224 pixels, resulting in 14x14 patches, or 196

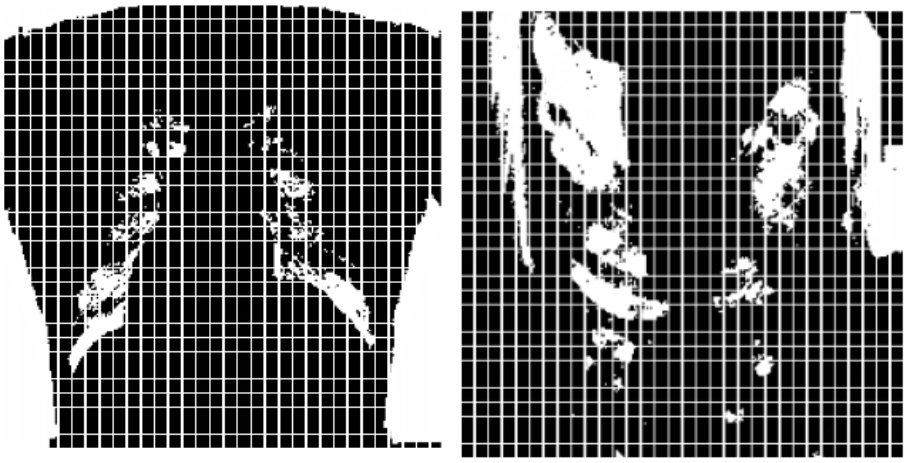
patches in total. A custom patch extraction layer was developed to handle these patches, converting them into a format suitable for the transformer. These tokenized patches are then fed into the transformer model.

Figure 5

CXR Images Before Patching and Augmentation

Figure 6

CXR Images after patching and augmentation



The ViT model in this study was built using the ViT-B16 as the base, followed by additional dense layers to refine feature extraction. Batch normalization layers were included to accelerate training, and dropout layers were used to prevent overfitting. The model was compiled with an Adam optimizer and a categorical cross-entropy function. Adam optimizer used to accelerate the training, as explained by (Arya & Sastry, 2023). Also, despite binary cross-entropy being theoretically more suitable for binary classification, it showed overfitting results initially. Therefore, a categorical cross-entropy was used for the loss function -which is opposite to the binary cross entropy used for CNN models.

Training involved early stopping to avoid overfitting and an interactive stopping callback for manual intervention. Data augmentation techniques, such as vertical and horizontal flipping,

rotation, and brightness adjustment, were applied to the ViT model's preprocessing to increase generalization capabilities in the TB classification context (Hao et al., 2021).

The structure of the ViT model begins with flattened patches undergoing linear projection to transform each 1D vector into a lower-dimensional representation, preserving relationships while reducing dimensionality. Positional embeddings provide positional information to the model. The self-attention mechanism allows the model to focus on various image regions, capturing relationships between different parts (Jiang et al., 2022). The encoder's first layer, a self-attention layer, enables each patch to gather information from others and capture dependencies. The final MLP head maps the transformer's output into the desired format, such as image classification probabilities.

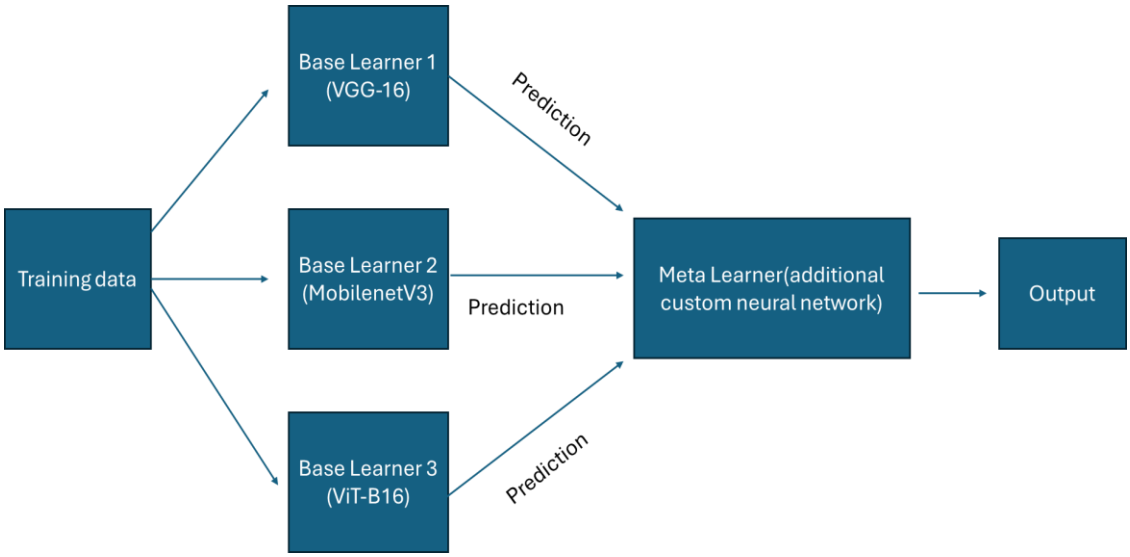
There have been many variants of ViT invented after the first proposal of ViT. In this study, the first visual transformer is used as a baseline – the ViT-B16 introduced in 16x16 study by Dosovitskiy et al. To maintain the scope of this study, the variants of ViT will not be investigated.

3.9 Stacking

Stacking is an ensemble learning technique that synergizes the outputs of several heterogeneous models to yield a more accurate and robust final prediction. This technique can be employed either at the architectural level or the output level – as explored in the literature review. In this study, we focus on the output level, utilizing the predictions of the base learners to create a meta-learner.

Figure 7

Illustration of Stacking models used in the Study



Initially, two distinct Convolutional Neural Network (CNN) architectures, VGG16 and MobileNetV3, are trained separately on the same dataset. These base models, after training, generate predictions on the validation set, capturing different facets of the underlying data patterns. The predictions from both models are then concatenated to form a new feature set, thereby creating a combined representation of the data. This new feature matrix integrates the strengths of each base model, with each row representing the combined predictions for a CXR

sample (Roy et al., 2023). This process allows the subsequent meta-model to leverage the diverse information extracted by the individual base models.

For the meta-model, instead of opting for a complex neural network, I employ logistic regression. Logistic regression is particularly suitable for this binary tuberculosis (TB) classification context due to its simplicity and fewer hyperparameters, which reduces the complexity of model selection (Q. Liu et al., 2022). Given that the combined predictions from VGG16, MobileNetV3 and ViTB-16 already capture complex patterns, logistic regression can effectively integrate these predictions without overfitting.

Moreover, logistic regression provides interpretable results, allowing better understanding and explanation of feature contributions to the final predictions. This transparency is invaluable in medical applications, such as TB case analysis, where understanding the decision-making process is crucial for radiologists who may not be familiar with machine learning. Thus, logistic regression is a practical, efficient, and interpretable choice for the meta-model in this ensemble learning setup.

3.10 Hyperparameter Tuning

For the baseline models, metrics were tuned to find the best possible accuracy for each model. During the tuning process, I experimented with different combinations of hyperparameters such as learning rate, batch size, number of epochs, and architecture-specific parameters. The goal was to identify the optimal configuration that maximizes model performance while avoiding overfitting or underfitting.

Hyperparameters used are listed in the table from Appendix A: Hyperparameters.

4. Results

On this section, the training and performances of each baseline models will be reported. Each of the models' performances will be further discussed in the Discussion section.

4.1 VGG- 16

The VGG-16 model initially exhibited a significant disparity between validation loss and validation accuracy when employing the categorical cross entropy loss function (Figure 8). This discrepancy indicated that the model was struggling to generalize well from the training data to the validation data. To address this issue, the loss function was changed to binary cross entropy.

Figure 8

Training VGG-16 using the Categorical Cross Entropy

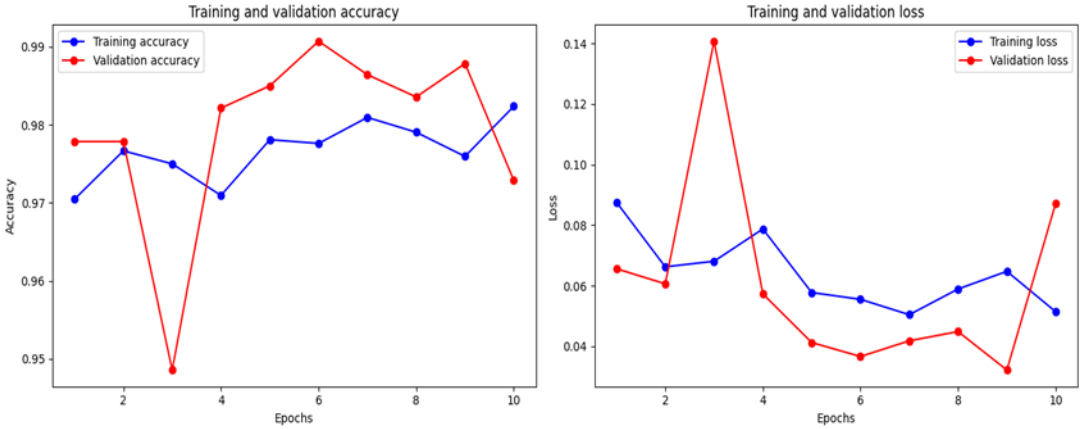
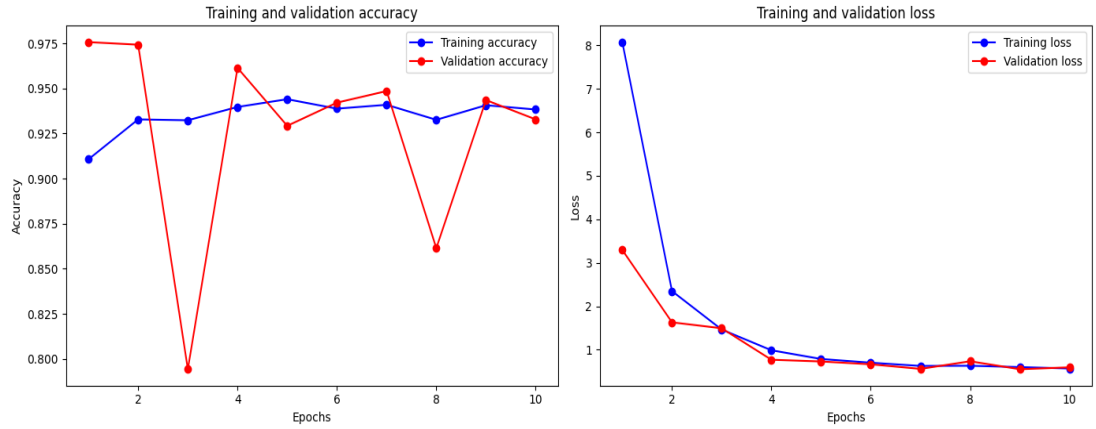


Figure 9

Improved Convergence between training and validation losses with the binary cross entropy



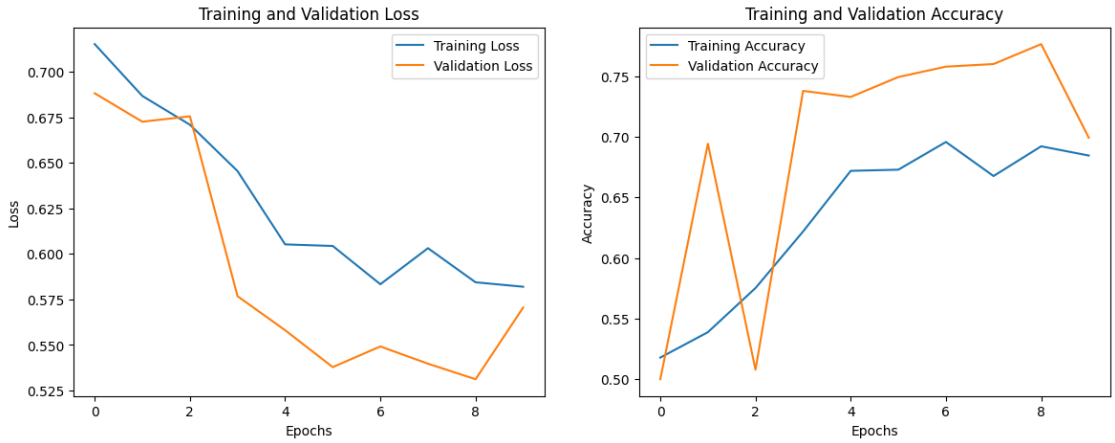
Switching to binary cross entropy resulted in better convergence between training loss and validation loss (Figure 9). The training accuracy started at around 0.91 and reached approximately

0.95 by the last epoch. However, the validation accuracy varied significantly across epochs, indicating less stable generalization ability. Despite this variability, the final validation accuracy (~0.94) was close to the training accuracy, indicating that the model was learning effectively. However, the fluctuations in general movement across validation accuracy graph suggest that the model's performance is sensitive to the learning rate, as noted by (Vidyabharathi et al,2021).

4.2 MobilenetV3Large

Figure 10

Training Performance of MobileNetV3Large



The MobileNetV3Large model also showed divergence when using categorical cross entropy, necessitating a switch to binary cross entropy. This adjustment led to an effective learning and generalization, evidenced by a fairly consistent decrease in both training and validation loss, along with an increase in accuracy. However, initial fluctuations in validation accuracy pointed to potential adjustments needed for the learning rate, as indicated by Nakamura et al. (2021).

4.3 ViT-B16

Figure 11

Training Performance of ViT-B16

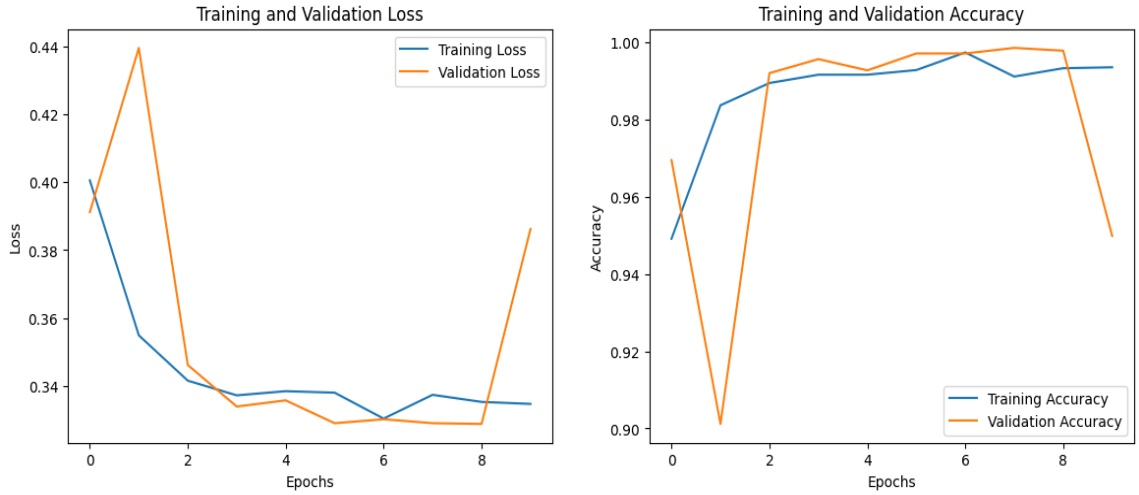
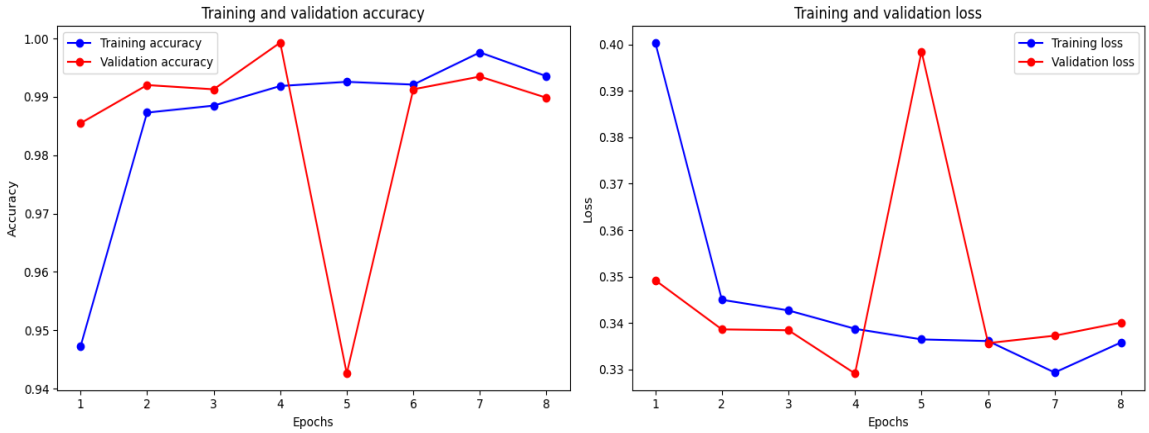


Figure 12

Training Performance of ViT-B16 until epoch 8



Initial training with the ViT-B16 model suggested potential overfitting beyond the 8th epoch. To mitigate this, re-training was limited to 8 epochs, which demonstrated a steady decrease in training loss and more stable validation loss (Figure 12).

4.4 Meta Model- CNN/ Meta Model- CNN+ViT-B16

Table 1

Performance metrics of the baseline models, CNN ensemble, CNN+ViTB-16 ensemble

Metric	VGG16 (Updated)	MobileNetV3	ViT-B16	Ensemble (CNN-Meta)	Ensemble (VGG16 + MobileNetV3 + ViT)
Accuracy	88.00%	68.21%	87.10%	90.00%	**91.93%**
Precision (Normal)	81.00%	93.10%	**99.60%**	87.00%	87.10%
Precision (TB)	**99.00%**	86.30%	79.60%	94.00%	98.20%
Recall (Normal)	**100.00%**	93.10%	74.40%	**95.00%**	98.57%
Recall (TB)	76.00%	43.30%	**99.70%**	86.00%	85.40%
F1-Score (Normal)	89.00%	75.00%	85.10%	91.00%	**92.50%**
F1-Score (TB)	86.00%	58.00%	88.60%	90.00%	**91.40%**

Notable performances in each model are colored in bold. The CNN ensemble model achieved a balanced performance with an overall accuracy of 90%, demonstrating robustness in predictions through a good balance between precision and recall for both classes. The strong F1-scores for both classes suggest that the model handles both TB and normal cases well.

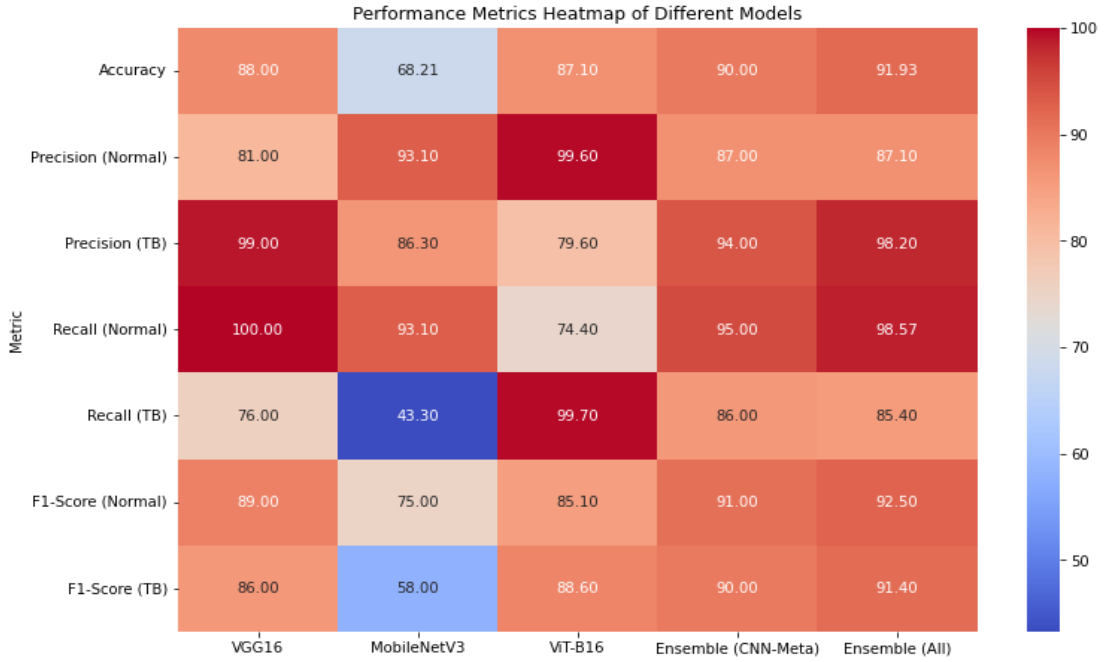
The ensemble model combining VGG16, MobileNetV3, and ViTB-16 achieved the highest accuracy at 91.93%, showcasing the enhanced performance from integrating diverse architectures. This model significantly improved precision and recall for TB, indicating effective sensitivity and minimal false positives, thus highlighting its potential for reliable TB diagnosis.

A detailed discussion will be made in the following section.

5. Discussion & Conclusion

Figure 13

Heatmap Visualization of the results



5.1 Research Goals

This study aimed to determine if an ensemble model combining CNNs and ViT could outperform models based solely on CNNs for the classification of TB images. The primary goal was to assess the effectiveness of stacking techniques that merge these different architectures.

5.2 Findings

Visualization (Figure 13) using Heatmap was used on results to provide a strong visual aid that supports the following analytical narrative (Davila et al., 2023).

As seen from the darker red portion of the heatmap, the ensemble model combining VGG16, MobileNetV3, and ViT achieved superior performance across most metrics compared to both the individual CNN models and the CNN-only ensemble model (CNN-Meta). Specifically, the CNN+ViT ensemble model achieved the highest accuracy at 91.93%, surpassing both the individual VGG16 model (88.00%) and the CNN-Meta ensemble model (90.00%). Additionally, the ensemble model exhibited a significantly high precision for TB (98.20%) and an improved recall for TB (85.40%) compared to the VGG16 model's recall of 76.00%. These improvements demonstrate that integrating ViTs with CNNs enhances the model's ability to correctly classify TB and Normal cases, making it more sensitive in identifying true TB cases and reducing the number of missed diagnoses.

5.3 Comparison to Baseline Models

The VGG16 model demonstrated high precision (99.00%) and recall (76.00%) for TB but had a significant number of missed TB cases (24%). The said model exhibits very high precision of 99.5% - indicating that when it predicts TB, it is almost always correct. Also, the F1 Score for the TB class is approximately 0.86, reflecting a good balance between precision and recall. It means the model performs well in terms of both accurately identifying true positive TB cases and minimizing false positives.

However, the recall for the TB class is 76%, indicating that the model identifies 76% of actual TB cases but misses about 24% of them, resulting in 168 false negatives. This high number of false negatives is concerning, especially in a TB diagnosis context where missing TB cases can have serious implications on the patients. It suggests that the VGG model trained with the proposed dataset may be conservative in predicting TB to avoid false positives, which can lead to missed diagnoses. While the model's high precision is beneficial, the significant number of false negatives indicates a need for improvement to ensure that more TB cases are correctly identified.

The MobileNetV3 model showed the lowest performance among the baseline models, with an accuracy of 68.21% and a low recall for TB (43.30%), indicating a high number of missed TB cases. The training graph in the results section indicated improving performance over epochs for both training and validation accuracy. However, the MobileNet model's F1 Score of approximately 0.577 shows that while it has decent precision, its low recall significantly impacts its overall effectiveness. This suggests that the model is not generalizing as well as desired, particularly in identifying Tuberculosis cases.

The ViT-B16 model achieved strong performance with very high recall for TB (99.70%) but moderate precision (79.60%), reflecting a higher rate of false positives. For Normal cases, the model achieves a high precision of 99.6%, accurately predicting most normal cases. However, the recall for normal is 74.4%, indicating that a significant portion of normal cases is misclassified as TB. This suggests potential overfitting, where the model might be too focused on identifying all positive cases at the expense of precision (Aliferis & Simon, 2024). To enhance reliability, efforts should focus on improving precision for TB and balancing recall for normal cases to reduce false positives.

The results of baseline models can also be found in Appendix B: Confusion matrix – Baseline Models.

The CNN-Meta ensemble model showed balanced performance with an accuracy of 90.00%, precision for TB at 94.00%, and recall for TB at 86.00%, indicating a good balance between precision and recall. Moreover, the advanced CNN+ViT ensemble model outperformed all baseline models and the CNN-Meta ensemble, achieving the highest accuracy and the best overall balance in performance metrics.

Overall, this answers the research question.

> *Does a stacked ensemble model combining CNNs (mobilenetv3, VGG-16) and ViTB-16 outperform an ensemble model comprised solely of CNNs in terms of performance metrics for the classification of TB?*

The accuracy of the 3-type ensemble is higher than the 2-type ensemble with 91.93% compared to two type ensemble's 90%.

5.4 Comparison to Other Works

Compared to similar ensemble approach, Almezhghwi et al. (2021) achieved AUC values of 98% and 97% using a support vector machine on top of AlexNet and VGG16 to classify CXR images, respectively. However, the reliance on handcrafted features limits their approach. In contrast, this study's use of ViTs leverages automatic feature extraction, improving diagnostic accuracy and generalizability to 98.2%.

As explored in the related work section, Nahiduzzaman et al. (2023) developed a framework classifying 17 different lung disorders using an extreme machine learning algorithm (ELM) on top of a CNN, achieving over 90% accuracy. While effective, the computational complexity of ELM could hinder real-time applications. This study's simpler logistic regression meta-learner in the ensemble model, in comparison, balances performance with computational efficiency.

Ali et al. (2023) also employed meta-learning for breast cancer classification, achieving 87.9% accuracy. It is the work done in This study's higher accuracy of 91.93% in TB classification demonstrates the effectiveness of stacking ensemble techniques, particularly when combining CNNs and ViTs.

As explored in the related work section, Ravi et al. (2022) ensembled variants of EfficientNet models (B0, B1, B2) for COVID-19 detection, achieving 98% accuracy. While EfficientNet has proven to be a promising model in the ensemble field, this study's use of MobileNetV3 and ViT-B16 utilizes a vast difference in model architecture, achieving comparable results.

Finally, many works in this field utilize the readily available Shenzen and Montgomery datasets (Jaeger et al., 2014), as also seen in the work by Ammar et al. (2022). However, these datasets consist of only 820 images, with 394 diagnosed as tuberculosis. Although readily accessible, the small dataset size, even with data augmentation, can lead to overfitting and inaccurate results (Han et al., 2024). Particularly with the example by Ammar et al., they have ensembled ViT model with EfficientnetB3 in the output level, achieving the accuracy of 96%. Its validity is however questionable due to the small sample size and the test pool. In contrast, this study uses 7,000 images from TB Portals, with 3,500 TB images, promising higher validity – if not a superior accuracy.

5.5 Limitations and Future Work

The study's generalizability is limited using downsampled data, which may lack diversity in demographics and TB variations. Future studies should use a diverse CXR samples, especially the normal CXR images to match the quality of the dataset from TB portals. Additionally, testing the model on independent datasets, including newly identified TB cases, will further validate its generalizability. Also, future work could explore architectural-level fusion and stacking even more layers to possibly improve the performance.

Also, further evaluations could improve the study's outcomes. For example, using GRAD-CAM to visualize where the CNN focuses can help extract feature importance and evaluate model behavior (Selvaraju et al., 2017). Not to mention the U-Net Segmentation model commonly used in the preprocessing steps (cropping lungs) to feed more information into the model (Rajaraman et al., 2021).

5.6 Conclusion

This research demonstrated the benefits of combining CNNs and ViTs in ensemble models for TB classification, achieving superior performance metrics compared to individual models and CNN-only ensembles. The findings suggest that integrating diverse model architectures in CNNs and ViT-B16 enhances diagnostic accuracy and robustness, making the final meta-model suitable for clinical applications in TB detection.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Retrieved from https://www.tensorflow.org/

Albashish, D. (2022). Ensemble of adapted convolutional neural networks (CNN) methods for classifying colon histopathological images. *PeerJ. Computer Science*, *8*, e1031. https://doi.org/10.7717/peerj-cs.1031

Ali, M. D., Saleem, A., Elahi, H., Khan, M. A., Khan, M. I., Yaqoob, M. M., Khattak, U. F., & Al-Rasheed, A. (2023). Breast Cancer Classification through Meta-Learning Ensemble Technique Using Convolution Neural Networks. *Diagnostics*, *13*(13), 2242. https://doi.org/10.3390/diagnostics13132242

Aliferis, C., & Simon, G. (2024). Overfitting, underfitting and general model Overconfidence and Under-Performance pitfalls and best practices in machine learning and AI. In *Computers in health care* (pp. 477–524). https://doi.org/10.1007/978-3-031-39355-6_10

Almezhghwi, K., Serte, S., & Al-Turjman, F. M. (2021). Convolutional neural networks for the classification of chest X-rays in the IoT era. *Multimedia Tools and Applications*, *80*(19), 29051–29065. https://doi.org/10.1007/s11042-021-10907-y

Ammar, L. B., Gasmi, K., & Ltaifa, I. B. (2022). VIT-TB: Ensemble Learning based VIT model for tuberculosis recognition. *Cybernetics and Systems*, *55*(3), 634–653. https://doi.org/10.1080/01969722.2022.2162736

Arya, M., & Sastry, G. H. (2023). Effective LSTM Neural Network with Adam Optimizer for Improving Frost Prediction in Agriculture Data Stream. In *Communications in computer and information science* (pp. 3–17). https://doi.org/10.1007/978-3-031-27034-5_1

Aurelio, Y. S., De Almeida, G. M., De Castro, C. L., & Braga, A. P. (2019). Learning from Imbalanced Data Sets with Weighted Cross-Entropy Function. *Neural Processing Letters/Neural Processing Letters*, *50*(2), 1937–1949. https://doi.org/10.1007/s11063-018-09977-1

Brad ski, G. (2000). The OpenCV library. *Dr. Dobb's Journal of Software Tools*.

Bjorck, N., Gomes, C. P., Selman, B., & Weinberger, K. Q. (2018). Understanding batch normalization. *Neural Information Processing Systems*, *31*, 7694–7705. https://papers.nips.cc/paper/7996-understanding-batch-normalization.pdf

Capobianco, G., Cerrone, C., Di Placido, A., Durand, D., Pavone, L., Russo, D. D., & Sebastiano, F. (2021). Image convolution: a linear programming approach for filters design. *Soft Computing*, *25*(14), 8941–8956. https://doi.org/10.1007/s00500-021-05783-5

Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., & Liu, Z. (2020). Dynamic ReLU. In *Lecture notes in computer science* (pp. 351–367). https://doi.org/10.1007/978-3-030-58529-7_21

Chollet, F. (2018). *Deep Learning with Python*. Manning Publications.

Chowdhury, M. E. H., Rahman, T., Khandakar, A., Mazhar, R., Kadir, M. A., Mahbub, Z. B., Islam, K. R., Khan, M. S., Iqbal, A., Emadi, N. A., Reaz, M. B. I., & Islam, M. T. (2020). Can AI help in screening viral and COVID-19 pneumonia? *IEEE Access*, *8*, 132665–132676. https://doi.org/10.1109/access.2020.3010287

Cudahy, P., & Shenoi, S. V. (2016). Diagnostics for pulmonary tuberculosis. *Postgraduate Medical Journal*, *92*(1086), 187–193. https://doi.org/10.1136/postgradmedj-2015-133278

Davila, F., Paz, F., & Moquillaza, A. (2023). Usage and Application of Heatmap Visualizations on Usability User Testing: A Systematic Literature review. In *Lecture notes in computer science* (pp. 3–17). https://doi.org/10.1007/978-3-031-35702-2_1

Dey, S., Bhattacharya, R., Malakar, S., Mirjalili, S., & Sarkar, R. (2021). Choquet fuzzy integral-based classifier ensemble technique for COVID-19 detection. *Computers in Biology and Medicine*, *135*, 104585. https://doi.org/10.1016/j.compbiomed.2021.104585

Dey, S., Roychoudhury, R., Malakar, S., & Sarkar, R. (2022). An optimized fuzzy ensemble of convolutional neural networks for detecting tuberculosis from Chest X-ray images. *Applied Soft Computing*, *114*, 108094. https://doi.org/10.1016/j.asoc.2021.108094

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Lecture notes in computer science* (pp. 1–15). https://doi.org/10.1007/3-540-45014-9_1

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021a). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR*. https://openreview.net/pdf?id=YicbFdNTTy

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021b). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ICLR 2021*. https://openreview.net/pdf?id=YicbFdNTTy

Dubey, S. R., Singh, S. K., & Chaudhuri, B. B. (2022). Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, *503*, 92–108. https://doi.org/10.1016/j.neucom.2022.06.111

Dutschmann, T., Kinzel, L., Ter Laak, A., & Baumann, K. (2023). Large-scale evaluation of k-fold cross-validation ensembles for uncertainty estimation. *Journal of Cheminformatics*, *15*(1). https://doi.org/10.1186/s13321-023-00709-9

Elgendi, M., Nasir, M. U., Tang, Q., Smith, D., Grenier, J. P., Batte, C., Spieler, B., Leslie, W. D., Menon, C., Fletcher, R. R., Howard, N., Ward, R., Parker, W., & Nicolaou, S. (2021). The

Effectiveness of image augmentation in deep learning networks for detecting COVID-19: A Geometric Transformation perspective. *Frontiers in Medicine, 8*. https://doi.org/10.3389/fmed.2021.629134

Ennouni, A., Sihamman, N. O., Sabri, M. A., & Aarab, A. (2021). A weighted voting deep learning approach for plant disease classification. *Journal of Computer Sciences/Journal of Computer Science*, *17*(12), 1172–1185. https://doi.org/10.3844/jcssp.2021.1172.1185

Gökalp, O., & Taşçı, E. (2019). Weighted Voting Based Ensemble Classification with Hyper-parameter Optimization. *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*. https://doi.org/10.1109/asyu48272.2019.8946373

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Han, M., Cai, D., Huo, Z., Shen, Z., Tang, L., Yang, S., & Wang, C. (2024). Reducing Overfitting Risk in Small-Sample Learning with ANN: A Case of Predicting Graduate Admission Probability. In *Communications in computer and information science* (pp. 404–419). https://doi.org/10.1007/978-981-97-1277-9_31

Hao, R., Namdar, K., Liu, L., Haider, M. A., & Khalvati, F. (2021). A comprehensive study of data augmentation Strategies for prostate Cancer Detection in Diffusion-Weighted MRI using convolutional Neural Networks. *Journal of Digital Imaging*, *34*(4), 862–876. https://doi.org/10.1007/s10278-021-00478-7

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *DBLP*. https://doi.org/10.1109/cvpr.2016.90

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. *IEEE*. https://doi.org/10.1109/iccv.2019.00140

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3), 90-95.

Jaeger, S., Candemir, S., Antani, S., Wáng, Y. X., Lu, P. X., & Thoma, G. (2014). Two public chest X-ray datasets for computer-aided screening of pulmonary diseases. *Quantitative Imaging in Medicine and Surgery, 4*(6), 475-477. https://doi.org/10.3978/j.issn.2223-4292.2014.11.20

Jeczmionek, E., & Kowalski, P. A. (2021). Flattening layer pruning in convolutional neural networks. *Symmetry*, *13*(7), 1147. https://doi.org/10.3390/sym13071147

Jiang, K., Peng, P., Lian, Y., & Xu, W. (2022). The encoding method of position embeddings in vision transformer. *Journal of Visual Communication and Image Representation*, *89*, 103664. https://doi.org/10.1016/j.jvcir.2022.103664

Kaiser, Ł., Google Brain, Gomez, A. N., University of Toronto, Chollet, F., & Google Brain. (2017). Depthwise separable convolutions for neural machine translation [Journal-article]. *arXiv*. https://arxiv.org/abs/1706.03059v2

Kalaivani, S., & Seetharaman, K. (2022). A three-stage ensemble boosted convolutional neural network for classification and analysis of COVID-19 chest x-ray images. *International Journal of Cognitive Computing in Engineering*, *3*, 35–45. https://doi.org/10.1016/j.ijcce.2022.01.004

Kiflie, M. A., Sharma, D. P., Haile, M. A., & Srinivasagan, R. (2024). EfficientNet Ensemble Learning: Identifying Ethiopian Medicinal Plant Species and Traditional Uses by Integrating Modern Technology with Ethnobotanical Wisdom. *Computers*, *13*(2), 38. https://doi.org/10.3390/computers13020038

Liu, Q., Salanti, G., De Crescenzo, F., Ostinelli, E. G., Li, Z., Tomlinson, A., Cipriani, A., & Efthimiou, O. (2022). Development and validation of a meta-learner for combining statistical and machine learning prediction models in individuals with depression. *BMC Psychiatry*, *22*(1). https://doi.org/10.1186/s12888-022-03986-0

Mabrouk, A., Redondo, R. P. D., Dahou, A., Elaziz, M. A., & Kayed, M. (2022). Pneumonia detection on chest x-ray images using ensemble of deep convolutional neural networks. *Applied Sciences*, *12*(13), 6448. https://doi.org/10.3390/app12136448

Marques, G., Ferreras, A., & De La Torre-Diez, I. (2022). An ensemble-based approach for automated medical diagnosis of malaria using EfficientNet. *Multimedia Tools and Applications*, *81*(19), 28061–28078. https://doi.org/10.1007/s11042-022-12624-6

McKinney, W. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51-56).

Nahiduzzaman, M., Goni, M. O. F., Hassan, R., Islam, M. R., Syfullah, M. K., Shahriar, S. M., Anower, M. S., Ahsan, M., Haider, J., & Kowalski, M. (2023). Parallel CNN-ELM: A multiclass classification of chest X-ray images to identify seventeen lung diseases including COVID-19. *Expert Systems With Applications*, *229*, 120528. https://doi.org/10.1016/j.eswa.2023.120528

Nakamura, K., Derbel, B., Won, K., & Hong, B. (2021). Learning-Rate annealing methods for deep neural networks. *Electronics*, *10*(16), 2029. https://doi.org/10.3390/electronics10162029

Nilsson, N. J. (1965). *Learning machines*.

Oloko-Oba, M., & Viriri, S. (2021). Ensemble of EfficientNets for the diagnosis of Tuberculosis. *Computational Intelligence and Neuroscience*, *2021*, 1–12. https://doi.org/10.1155/2021/9790894

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. *IEEE*. https://doi.org/10.1109/cvpr.2014.222

Patel, K., & Wang, G. (2022). A discriminative channel diversification network for image classification. *Pattern Recognition Letters*, *153*, 176–182. https://doi.org/10.1016/j.patrec.2021.12.004

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Pradhan, A. K., Mishra, D., Das, K., Obaidat, M. S., & Kumar, M. (2022). A COVID-19 X-ray image classification model based on an enhanced convolutional neural network and hill climbing algorithms. *Multimedia Tools and Applications*, *82*(9), 14219–14237. https://doi.org/10.1007/s11042-022-13826-8

Rajaraman, S., Folio, L. R., Dimperio, J., Alderson, P. O., & Antani, S. K. (2021). Improved Semantic Segmentation of Tuberculosis—Consistent Findings in Chest X-rays Using Augmented Training of Modality-Specific U-Net Models with Weak Localizations. *Diagnostics*, *11*(4), 616. https://doi.org/10.3390/diagnostics11040616

Rajaraman, S., Zamzmi, G., Folio, L. R., & Antani, S. (2022). Detecting Tuberculosis-Consistent findings in lateral chest X-Rays using an ensemble of CNNs and vision transformers. *Frontiers in Genetics*, *13*. https://doi.org/10.3389/fgene.2022.864724

Ravi, V., Acharya, V., & Alazab, M. (2022). A multichannel EfficientNet deep learning-based stacking ensemble approach for lung disease detection using chest X-ray images. *Cluster Computing*, *26*(2), 1181–1203. https://doi.org/10.1007/s10586-022-03664-6

Ridnik, T., Ben-Baruch, E., Noy, A., & Zelnik-Manor, L. (2021). ImageNet-21K pretraining for the masses. *arXiv (Cornell University)*. https://arxiv.org/pdf/2104.10972

Roy, M., Baruah, U., & Varma, V. (2023). TransDL: A transfer learning-based concatenated model for Covid-19 identification and analysis of posteroanterior chest X-ray images. *Multimedia Tools and Applications*, *83*(11), 33421–33443. https://doi.org/10.1007/s11042-023-16825-5

Saabni, R., & Schclar, A. (2020). Facial Expression Recognition Using combined Pre-Trained Convnets. *ICAITA 2020*. https://doi.org/10.5121/csit.2020.100908

Sarvamangala, D. R., & Kulkarni, R. V. (2021). Convolutional neural networks in medical image understanding: a survey. *Evolutionary Intelligence*, *15*(1), 1–22. https://doi.org/10.1007/s12065-020-00540-3

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *ICCV'17*. https://doi.org/10.1109/iccv.2017.74

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0197-0

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *INRIA Aerial Image Labeling*. http://export.arxiv.org/pdf/1409.1556

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958. https://jmlr.csail.mit.edu/papers/volume15/srivastava14a/srivastava14a.pdf

Vidyabharathi, D., Mohanraj, V., Kumar, J. S., & Suresh, Y. (2021). Achieving generalization of deep learning models in a quick way by adapting T-HTR learning rate scheduler. *Personal and Ubiquitous Computing*, *27*(3), 1335–1353. https://doi.org/10.1007/s00779-021-01587-4

Waskom, M. L. (2021). Seaborn: Statistical data visualization. *Journal of Open Source Software*, 6(60), 3021.

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, *3*(1). https://doi.org/10.1186/s40537-016-0043-6

Wendler, T., & Gröttrup, S. (2021). Imbalanced data and resampling techniques. In *Springer eBooks* (pp. 1147–1191). https://doi.org/10.1007/978-3-030-54338-9_10

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38-45.

Xu, Y., & Goodacre, R. (2018). On splitting training and validation set: A comparative study of Cross-Validation, Bootstrap and systematic sampling for estimating the generalization performance of supervised learning. *Journal of Analysis and Testing*, *2*(3), 249–262. https://doi.org/10.1007/s41664-018-0068-2

Yamashita, R., Nishio, M., Gian, R. K., DO, & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights Into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

Yang, Y., Lv, H., & Chen, N. (2022). A Survey on ensemble learning under the era of deep learning. *Artificial Intelligence Review*, *56*(6), 5545–5589. https://doi.org/10.1007/s10462-022-10283-5

Yang, Y., & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. *International Conference on Machine Learning*, 412–420. https://ci.nii.ac.jp/naid/10018780153

Yang, Z., Sinnott, R. O., Bailey, J., & Ke, Q. (2023). A survey of automated data augmentation algorithms for deep learning-based image classification tasks. Knowledge and Information Systems, 65(7), 2805–2861. https://doi.org/10.1007/s10115-023-01853-2

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Lecture notes in computer science* (pp. 818–833). https://doi.org/10.1007/978-3-319-10590-1_53

Zhang, Y., Wang, J., Gorriz, J. M., & Wang, S. (2023). Deep learning and vision transformer for medical image analysis. *Journal of Imaging*, *9*(7), 147. https://doi.org/10.3390/jimaging9070147

Appendix A: Hyperparameters

Hyperparameter Tuning

Hyperparameter	VGG-16, MobilenetV3Large	ViT Model
Learning Rate	**0.001**, 0.0001	**0.001** (0.0001 for ViT was too slow so it was not done for this research)
Optimizer	Adam	Adam
Number of Epochs	**8**,10	8,**10**
Dropout Rate	0.4,**0.5**	N/A
Number of Neurons (Dense Layers)	512, 1024, 2048	128, 256, 512
Activation Function	**ReLU**, LeakyReLU	ReLU, **GeLU**
Data Augmentation	shear_range: 0.1, **0.2** zoom range: 0.1, **0.2**	Brightness adjustion with a certain probability, horizontal and vertical flips
Patch Size	N/A	**16**, 32

Appendix B: Confusion Matrix

Baseline Models