Department of Computer Science

Submitted in part fulfilment for the degree of BEng.

Respiratory disease diagnosis from chest X-rays

Daynyus Halinauskas

May 2024

Supervisor: Nick Pears

Contents

List of Figures

List of Tables

Executive Summary

The main objective of this project was to explore the state-of-the-art deep learning techniques for respiratory disease diagnosis using publicly available chest X-ray (CXR) image datasets. I aimed to assess the performance of popularised architectures that are known to perform really well in general image classification tasks, adapting them to smaller datasets that have intricate details and subtle differences among images. These models also utilised feature extraction which allowed the loading of pre-trained weights acquired from ImageNet. While evaluating the chosen models the goal was to develop a hybrid model that uses the strengths of two different architectures, concatenating their dense layers to then make predictions, with the aim of achieving improved performance in disease classification.

Chest X-ray (CXR) imaging is a cost effective modality in medical diagnostics, but it does require very experienced clinicians to be able to interpret them. The shortage of highly qualified radiologists is present all around the world, especially in less developed countries due to the investments required for professional training. Hence the development of many automated analysis and diagnosis systems. Current systems heavily rely on deep learning. This rapid integration of deep learning techniques into medical imaging has experienced issues regarding availability of sufficient data. To address this challenge there is a need to explore and optimise the performance of deep learning approaches that can be used for CXR image analysis with the use of currently available datasets. Through this I aim to contribute to the development of more effective and accessible diagnostic tools that will help advancing healthcare delivery worldwide.

First I focused on researching the two main deep learning architectures that are used for image classification and their application in medical imaging. Most popular Convolutional Neural Network models that were explored are ResNet50, DenseNet121, and EfficientNetB4. The main attention based models that were covered are ViT-B16, DeIT and Swin-Transformer, from which the ViT-B16 was explored further.

Then a standard machine learning pipeline was followed:

1. Defining a problem and researching current solutions around it it
2. Acquiring data

3. Selecting models and frameworks to work with
4. Training the models
5. Evaluating/Testing the models
6. Analysing results

After acquiring the dataset and selecting a suitable framework to work with, all the models would undergo the last 3 stages of the pipeline. In the process loss/accuracy graphs were obtained from the training/validation stage.

The evaluation metrics used in the test stage were accuracy, loss, precision, recall, and f1-score. To further aid the evaluation process, confusion matrices were generated. ResNet50 and ViT-B16 showed the best performance, achieving an overall 86.77% and 86.72% accuracy, respectively, DenseNet121 achieved 83.95%, and EfficientNet 82.52%. After, the best CNN model was combined with the ViT-B16 to form ResViT, and put the test with the goal of outperforming the individual models. Satisfactory results were obtained as ResViT achieved even better performance with an overall 88.20% accuracy. ResViT also had the most stable loss/accuracy graphs, and highest f1-score out of any models, indicating the effectiveness, reliability, and robustness of hybrid architectures.

Based on the acquired results of ResViT and its performance compared to the best individual models, It was evident that utilising both, extraction of local features and acquiring global relationships of image segments holds promise for achieving superior performance.

Creation of reliable and sufficient datasets should also be bought to attention. It should be as important as the exploration and optimisation of different models, since in supervised learning the datasets are considered to be foundation of the task at hand. Further research should focus on acquiring and preparing datasets that were thoroughly checked and approved by professional radiologists to maximise the full potential of deep learning models.

With the outcome of this research future tasks should prioritise the utilisation finer extracted features from models that make up the combination model rather than relying on concatenated dense layers. Fine-tuning should be explored instead of feature extraction, as it most certainly will guarantee better results due to having more trainable parameters, but it is important to take into account the trade-offs, including longer training times and increased computational resource usage.

1 Introduction

Respiratory diseases are one of the most common non-communicable diseases, and are among the primary causes of morbidity and mortality globally[1]. One of the most available diagnostic tools is Chest X-ray Radiography (CXR), known for its affordability, portability, and speed[2]. It can provide detailed images of the chest, focusing primarily on the lungs and surrounding structures. Despite the advantages of CXR, it's interpretation heavily relies on the expertise and experience of clinicians. In regions with limited resources and skilled personnel there is an increased risk of misdiagnosis and delays. Such delays or errors in diagnosis can harm patients or be fatal[3]. This issue is particularly prevalent in less developed countries, where access to specialised training and quality assurance measures may be low due to the time and financial investment required to train qualified radiologists. When faced with these challenges a clear need of improved access to reliable diagnostic services is required in order to mitigate the risk of misdiagnosis and ensure reliable patient care. As a solution to these challenges, automated analysis and diagnosis of medical images became increasingly appealing.

Computer Aided Detection and Diagnosis (CAD) systems emerged in the 1980s-1990s[4] in order to help clinicians read medical images more efficiently and make more accurate diagnostic decisions. These systems initially focused on providing clinicians with tools to more efficiently analyse medical images, aiding in the detection of abnormalities or suspicious areas that may require further investigation.

As technology has advanced, the integration of Artificial Intelligence (AI) and Machine Learning (ML) has transformed medical imaging. Within ML, Deep Learning (DL)[5], a subset that has gained significant attention as it stands out for its important role in Computer Vision. Application of DL techniques have been extensively used in Automated Computer Diagnosis, having the ability to extract valuable insights from data sources like Electronic Healthcare Records (EHR) and by processing medical images through the use of Image Classification[6], thus driving innovation in medical imaging analysis.

Despite the incredible advantages of DL, the rapid adoption of it in medical imaging came with challenges regarding availability of quality data. Unlike

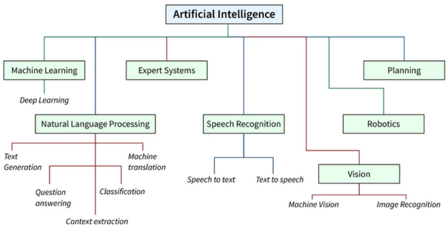

Figure 1.1: Artificial Intelligence Subsets

other fields where large datasets are readily available, medical imaging datasets lack sufficient size. The limited availability of training data can result in overfitting and hinder the generalisation of DL models. Not only there is a requirement for large datasets, but there is also a need for correct labelling. Teams of qualified radiologist manually have to go over thousands of images and label them, this manual work introduces errors, hence why so many medical datasets suffer from misclassified labels. The available datasets also suffer from inter-class similarity and limited diversity, making it difficult to extract smaller features necessary for accurate predictions when faced with multiple different diseases. Imbalanced data further complicates matters, as training DL models with such datasets leads to biased models.

Computational resources required for training DL models from scratch on specific datasets are expensive. Majority of DL architectures demand very high computational power and memory resources beyond of what is typically available to individuals or healthcare institutions. These resources are usually hosted in specialised computing facilities or on the cloud, where the access to such infrastructures can be very costly. So even if a large, high-quality dataset is available, the lack of computational power to process and analyse the data poses a significant limitation.

In this report, I aim to research how the challenges of limited chest X-ray datasets and computational power can be handled with the use of transfer learning on the current state-of-the-art(SOTA) DL architectures available for the task of classification. Convolutional Neural Networks and the newly popularised Vision Transformers will be explored with an additional experimental model which will utilise the features extracted from multiple models.

2 Literature Review

2.1 Machine Learning and Neural Networks

The idea of ML with regard to Neural Networks(NN) can be rooted back to 1950s, where the concept of a perceptron was proposed. A perceptron can be seen as a tiny computing unit shown in 2.1. It takes some inputs, multiplies each input by a weight it holds, sums them up adding a bias, and then runs the total through an activation function. The calculation can represented as $a = f(w^T x + b)$[7]. Despite the introduction of this idea the main limitation that was present in that time was the lack of computational power, therefore the adoption of ML was restricted. Even though this concept was not popularised yet, it will serve as the main foundation to future tasks revolving around images and use of NNs.

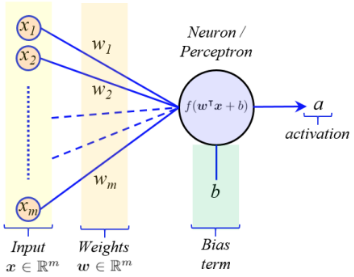

Figure 2.1: Perceptron Illustration

Artificial Neural Networks(ANN), or most commonly referred as NNs, are a collections of perceptrons/nodes connected to each other. This concept took inspiration from the human brain, how the neurons within a brain are capable of producing meaningful information when fired simultaneously, and how a large connection/layer of neurons would lead to complex behaviours[8]. The most common architecture out of the many[9] is known as Feed-forward Neural Network(FFN), which is acyclic and consists of 3 parts which the data flows through: *Input layer* that takes in the initial input, *Hidden layers* which can be multiple collections of layers that perform most of the work on data, and *Output Layer* that produces the final prediction. This architecture has proven to be very powerful within ML with the introduction of backpropagation[10], which allowed the weights that are connected to nodes to be readjusted based on how bad the predictions are.

DL is a sub-field of ML that focuses on networks that have many layers, hence the name "Deep". The depth being how many layers there are and width is the number of nodes present in each layer. These factors effect the amount of information that can be extracted from the input. DL has a wide range of applications in the fields of Computer Vision, Natural Language Processing, Speech Recognition and more. This report will focus on Computer Vision, specifically the task of Image Classification.

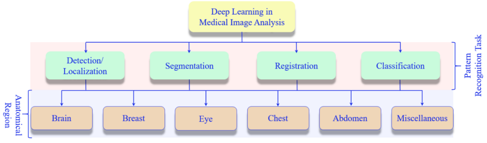

Figure 2.2: Deep Learning in Medical Image Analysis

2.2 Image Classification

The task of image classification is focused on producing predictions based on the input images. It falls in the category of supervised learning where all training data is labelled, meaning each input is associated with a corresponding class label. The model is then optimised using these input-label pairs. The optimised model would then predict the class label for each testing input based on the likelihood scores generated by the model. The major advancement in image classification came from the introduction of Convolutional Neural Networks[11], which were formalised by Yann LeCun as he performed recognition of handwritten digits[12]. Later on popularised by Alex Krizhevsky and AlexNet in 2013[13], by winning the ImageNet competition.

2.2.1 Convolutional Neural Networks

Convolutional Neural Networks(CNN) are FFNs with multiple layers whose structure loosely resembles the hierarchical organisation of neurons in the visual cortex[14]. The networks are made up of 4 main parts: *Convolution Layers* that are responsible for learning features/patterns from input images by applying convolution operations using learnable filters/kernels, *Non-Linearity layers*, which are responsible for transforming the input into output in some way using an activation function, most common being the Rectified Linear Unit (ReLU) that sets negative input values to zero. *Pooling Layers* reduce the size of feature maps produced after a convolution, essentially lowering the number of parameters to work with, Max-Pooling being the most common. *Fully Connected Layers* are used after all the convolutions by flattening the resulting feature maps into a 1d vector, serving as input to the fully connected layer which is usually followed by an activation function like Softmax. These components serve as foundation to all current SOTA CNN models. Popularised models will be explored in this report.

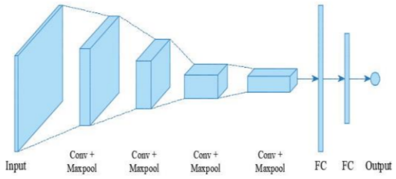

Figure 2.3: Basic CNN structure

The first is **ResNet**[15], that introduced the concept of residual/skip connections to address the degradation problem which deep NNs suffered from. As researchers tried obtaining more information about the input by scaling the depth of NNs they encountered the problem of vanishing gradients[16] during backpropagation the deeper a network got. ResNet's residual blocks which consisted of two or three convolution layers with batch normalisation and ReLU were followed up by skip connections that directly connected the input of the residual block to its output. The skip connection enabled the gradient to flow more easily during training, mitigating the problem of vanishing gradients. These concepts of residual blocks and skip connections enabled ResNet to be deeper while improving performance and being able to learn more complex features.

The second is **DenseNet**[17], that stands out with its densely connected layers that are present in the dense blocks. When compared to standard CNNs where data flows in a sequence, DenseNet's deep connections allow the flow of data between layers at different depths. Having this component allows previously generated feature maps be used in later layers. DenseNet's dense block consists of fully connected convolution layers, coupled with batch normalisation and an activation function. The output of each layer is concatenated with the feature maps from preceding layers within the same dense block. DenseNet also has transition layers between dense blocks to control the number of feature maps passed into the succeeding dense block, consisting of convolution layers followed by pooling layers to reduce the size. These methods encourage feature reuse and also tackle the vanishing gradient problem like ResNet, enabling efficient training of deep NNs.

Lastly **EfficientNet**[18], that was developed by using Neural Architecture Search(NAS)[19], introduced a new approach to balance CNN model efficiency and performance. EfficientNet utilises the compound scaling method that scales the network's *Depth* which is the amount of layers, *Width* that is the amount of nodes in a layer that can produce feature maps, and *Resolution* being the size of the input images. This network achieves great performance while maintaining computational efficiency, making them being applicable to devices with much smaller resources. The main component that is used in this network comes from MobileNetV2[20], they are referred to as MB blocks which consist of depth-wise, point-wise convolution layers and excitation blocks. These layers significantly reduce the computational cost compared to standard CNNs while retaining good performance.

The development of such CNN architectures definitely had an impact on Computer Vision, as they would become the base models to work from in order to introduce novel solutions. While CNNs are dominating the task of image classification an issue that arose was the ability to capture global context effectively. The filters used in CNNs were limited due to their size and could only extract regional features rather than using the whole image for context. In 2017 Researchers from Google were experimenting with attention based mechanism for NLP tasks, eventually coming up with the Transformer[21]. This architecture's highlight was the ability to model dependencies without regard to their distance in the input or output sequence, meaning the context gathered is only limited to computational resources. This was the missing piece that CNNs lacked, hence Google has taken upon themselves the task of exploring the Transformer architecture in image classification tasks, in the process producing the Vision Transformer(ViT) that could compete with top CNN models only using attention based mechanisms[22].

2.2.2 Vision Transformers

ViT is based on the original Transformer, and it utilises the attention mechanism which was inspired by how humans filter out irrelevant information while focusing on the meaningful parts of the data encountered in daily life. ViT's architecture can be observed in Figure 2.4 and it consists of 3 parts: *Image splitting + Positional Encoding*, where the input image is first broken down into patches, they are flattened into 1d, then these patches are assigned positions to preserve the order of an image. A *Classification Head*, which will be used in determining the output label is also added to the list of image patches with a position, all the inputs are referred to as tokens. Second part is the *Encoder Block* that consists of multiple encoder layers. Each encoder layer consists of two-sub layers: *Multi-head Attention(MHA)* and *Multi-layer Perceptron(MLP)* or just dense layers. The Encoder Block also consists of residual connection and normalisation between each sub-layer. The last part being the head of the MLP, which will contain the probabilities of every output label depending on the input image. The main powerhouse of this architecture is the MHA, it essentially traverses through every token and checks the "attention" between a single token and all of the other tokens in the sequence, proving to be very powerful when it comes to capturing global relationships and features.

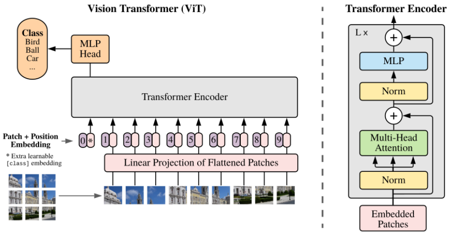

Figure 2.4: ViT Architecture from "An Image is worth 16x16 words"

Since the success of ViTs there has been research about utilising CNNs and their inductive biases with ViT's ability of capturing global relationships, which would compliment them both. The first application of such combination was used in **Detection Tranfromers**(DETR), which was an end-to-end detection model that uses the encoder to capture relation between image features extracted by a CNN model, the decoder of the transformer was used to generate the bounding boxes around objects[23].

Another challenge that was faced in image classification was the use of large-scale data for training and how computationally expensive this can get. **Data-efficient Image Transformer**(DeIT) was developed to solve the data efficiency problem. DeIT utilised self-supervised learning and distillation techniques[24]. Self-supervised learning allowed DeIT to be pre-trained on unlabelled data while distillation enabled to transfer "knowledge" from larger CNN models trained with labelled data to a much smaller model. With combination of these techniques DeIT needed much less data requirements and computational resources while still performing really well compared to other models that were relying heavily on extensive supervised training.

Lastly the issue of scaling images was tackled with the use of **Swin-Transformer** that reduced the cost of calculating the attention of high-resolution images[25]. This model introduced the concept of windowed self-attention to reduce computational complexity, and used shifted window attention to model cross-window relationships. With the introduction of hierarchical architecture of different sized patches this transformer was able to capture local and global dependencies no matter the size of the input. This model architecture proved to be excellent in image classification, still being one of the best when working on ImageNet dataset.

2.3 Machine Learning in Medical Imaging

ML has been used in medical imaging since the 1960s[26]. The first notable contributions related to modern DL techniques appeared in the 1990s[27]. Previous research was constrained by small training datasets and limited computational resources, often resulting in the use of shallow architectures with only 2-3 layers, which are now considered very basic compared to today's standards.

The transition of common diagnostic imaging modalities like X-ray, CT scans, Ultrasound, and MRI from analog to digital formats has indeed opened up numerous opportunities for DL in the medical imaging domain[28], and the use of AI technology to mine clinical data has become a major trend in medical industry. In recent years, DL techniques have been extensively applied to tasks such as Classification, Detection, Segmentation, and Registration of medical images. Many algorithms in the literature for medical image analysis have primarily relied on supervised learning methods.

CNNs are the most researched ML algorithms in medical image analysis due to their ability to preserve spatial relationships, as they are crucial in radiology when it comes to distinguishing normal and abnormal tissue structures[29]. Some of the most used CNN architectures include ResNet, DenseNet, and EfficientNet. Prime example being a CNN model that can compete with medical image experts, CheXNet[30], which uses DenseNet121 architecture. Another reason for such heavy research of CNNs in this field was due COVID-19, since the start of the pandemic the number of research papers that were published, relating to analysis of X-ray images really spiked up. One of the models that was produced received SOTA COVID-19 classification, CovidXRayNet, which used EfficientNetB0 as its backbone[31].

From the introduction of the vanilla ViT in 2021, many papers relating to medical imaging have been published. A ViT model proposed for diagnosis of COVID-19 that was trained on self-collected dataset of X-ray images, getting 92% accuracy score[32]. A paper that introduced global spatial information from ViTs to CNNs for pulmonary disease classification[33]. A combination of the previously mentioned CheXNet and ViT, CheXViT, achieving better accuracy's than CheXNet in multiple categories[34].

2.4 Critical Analysis

Through reviewing DL literature it is evident that the two giants in image classification tasks are CNNs and ViTs, they are currently considered to be SOTA. While these architectures are used for the same task, they differ in the main mechanisms that they use. CNNs use convolutional filters to extract local features of images and combine them together to form high-level representations, while ViTs break down images into patches forming input tokens and use attention in order to capture global relationships between tokens. These architectures can almost be seen as compliments of each other as their limitations are what the opposite does well, but CNNs being a more well-established concept, definitely has more leverage due to extensive research around it. ViTs are a newly established approach that is currently undergoing intense research.

When considering the field of medical imaging and applying these approaches to perform diagnosis there is a requirement for large amounts of good quality data as they are very data hungry, especially the ViTs, due to the lack of inductive biases. The field is known to have a shortage of available X-ray image datasets that can be used for training, and the ones available only contain frontal view, however lateral view images should also be considered as they can also provide valuable information. Therefore further research is required on the availability of such images, and if accuracy will improve with the use of frontal and lateral views combined during training.

In addition to the challenges faced with datasets there are limitations with hardware. SOTA models require very high performance hardware for training due to the complexity. Many papers in literature focus on using very complex models for extremely long times, where they require days if not weeks of training.

I would want to focus on leveraging transfer learning and use of efficient architectures trained under short periods of times as I do not have any access to datasets with lateral views of chest X-ray images. Transfer learning offers a solution to the data limitation problem by providing pre-trained weights obtained form ImageNet. By exploring these methods, I would like to provide insights into what performance can be obtained, and if it can be improved by utilising extracted features from different architectures.

Future research should definitely revolve around the uses of lateral chest X-ray views, potentially using Generative Adversarial Networks(GANs), to generate training data. The exploration of architectures that utilise CNNs and ViTs to capture as much information as possible efficiently should also be brought to attention in upcoming research.

3 Methodology

3.1 Approach

In order to accomplish the set out task, the approach in Figure 3.1 will be followed. First, the problem must be defined by looking at the current work around it and the techniques used in order to solve it. This will be followed by acquiring good quality data, processing and splitting it for the training and testing phases. Once all the data has been pre-processed, the suitable programming language and frameworks will be chosen in order to implement the models at interest. After setting up the environment, a training/validation pipeline must be implemented with set up hyperparameters and pre-trained weights through the use of transfer learning. Acquired models will then be tested and analysed by measuring various metrics that are suitable for multi-class classification.

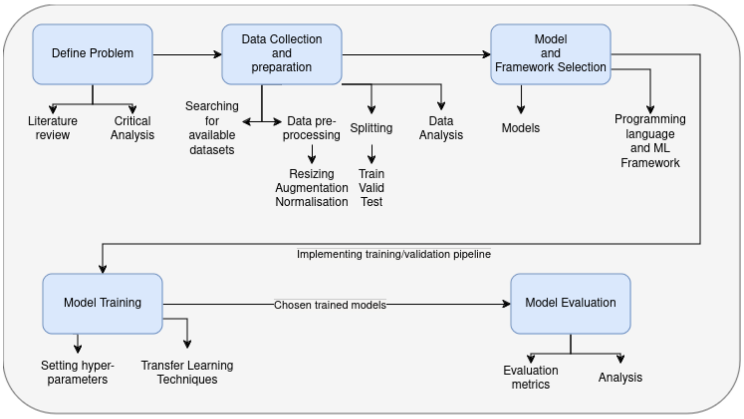

Figure 3.1: Methodology Approach

3.2 Data

The dataset that must be collected should be reasonably large, have various accurate labels and be balanced to prevent biases. The dataset used in this report was obtained from Kaggle[35]. It contains 10,095 frontal view X-ray images, covering 5 different classes: *Bacterial Pneumonia*, *Viral Pneumonia*, *Corona Virus Disease*, *Tuberculosis* and *Normal* shown in Figure3.2.

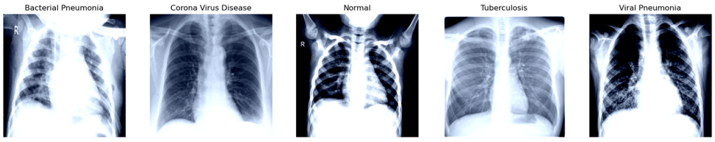

Figure 3.2: Random sample from dataset

The data is very well balanced from analysis observation in Figure3.3, hence the reason for choosing this dataset. The split that will be used is shown in Table 3.1.

Number of classes	Total Number of Images	Training	Validation	Testing
5	10,095	6057 (60%)	2019 (20%)	2019 (20%)

Table 3.1: Dataset split

All images will undergo resizing in order to work with them, and be normalised. Random rotation and random horizontal flipping will be applied to training data in order to introduce more variety, essentially increasing the number of training samples.

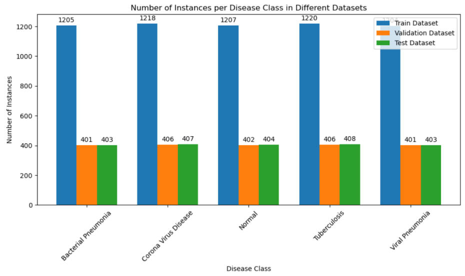

Figure 3.3: Overview of the dataset

3.3 Models

In this report four models will be considered and one created. The created model will be a combination of the best performing CNN and ViT model, using the extracted features of both in order to classify the disease. Each model will have pre-trained weights that were obtained from ImageNet training and have their classifier/dense layer modified to have output of 5 logits instead of probabilities.

ResNet50, a type of ResNet model that contains 50 layers in total. It starts with 64 filters of size 7×7 followed by max-pooling. Then 16 residual blocks, each block contains 3 convolutional layers. After the residual blocks, global average pooling is performed feeding into a fully connected layer that has 2048 inputs and 5 outputs. Modified architecture can be observed in Figure 3.4.

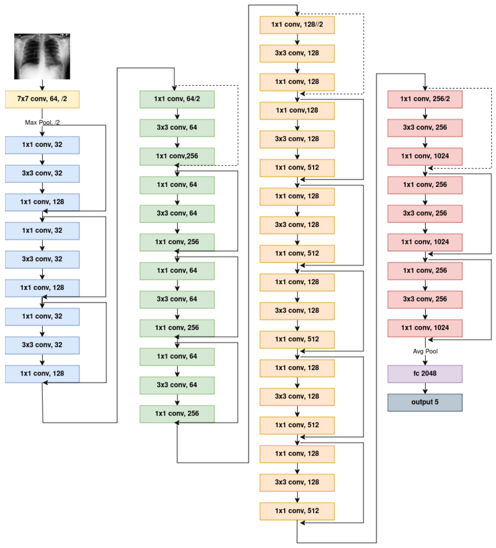

Figure 3.4: Modified ResNet50

DenseNet121, a DenseNet type that has 121 layers in total. Similar to ResNet50 in the beginning it has a convolution layer with 64 filters of size 7×7 that is followed by max-pooling layer. Followed by 4 dense blocks which consist of multiple layers where each layer within a dense block receives input from all preceding layers. Between every dense block there is a transition layer, which just consists of convolutional and pooling operations to reduce the feature maps produced. Global average pooling is performed feeding into a fully connected layer with 1024 inputs and 5 outputs. Modified architecture can be observed in Figure 3.5.

EfficientNetB4, a scaled up base model of EfficientNet that has 390 layers in total. In the beginning the model starts with 64 filters of size 3×3. After it is followed by the MB blocks that have their own configuration of convolution layers, this can be viewed in Figure 3.6. A convolutional layer of filter size 1×1 is used after the MB blocks, followed by global average pooling layer that feeds into a fully connected layer with 1792 inputs and 5 outputs.

ViT-B16 is the same model that was proposed in this paper[22]. Firstly the 224×244 image gets broken down into patches of size 16×16, producing 192 patches in total. The patches are then flattened into 1d and assigned positions to maintain order, also the classification head is added in this phase too. All of the input tokens enter the encoder simultaneously, there

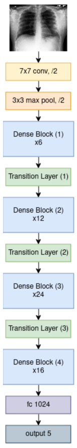

Figure 3.5: Modified DenseNet121

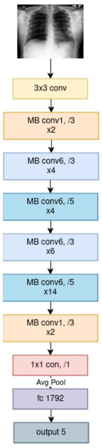

Figure 3.6: Modified EfficientNetB4

are 12 encoders in total, each output is fed into the next encoder until the last one has been reached. Firstly the input tokens are normalised and then split into *Queries*, *Keys* and *Values* to be fed into the MHA, secondly a residual connection is made to be added with the output of the MHA. This output is normalised and passed into the MLP, also the same output gets passed with a residual connection to be added with the output of the MLP layer. Final output will be passed into the next encoder, when all of encoders have been used, the final output of the MLP is used to obtain the MLP head that will contain the 5 output logits. The modified architecture can be viewed in Figure 3.7

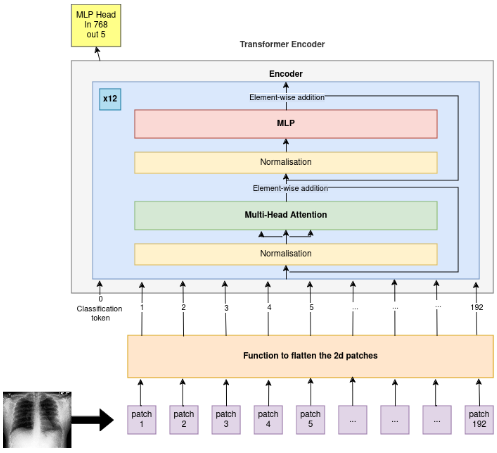

Figure 3.7: Modified ViT-B16

Combination Model is the experimental model that will be explored in this report. It will be a combination of the best performing CNN architecture with the ViT, where the extracted feature maps will be concatenated into a fully connected layer, passed through ReLU then into 5 outputs, the architecture can be viewed in Figure3.8. The inspiration for this experiment came from a paper that used this technique in order to classify leafs which also have small and global details[36].

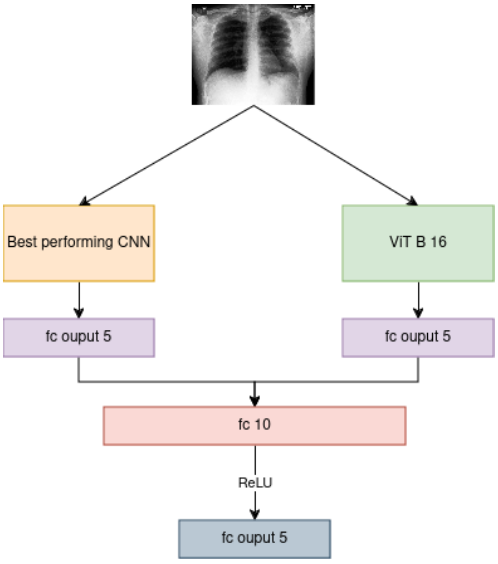

Figure 3.8: Combination Model Architecture

The reason for choosing the mentioned CNNs is that all of them use unique approaches, and try handling a specific challenge that is present in image classification, hence all three models will undergo training and be evaluated. Only one to be used in the combination model.

When it comes the ViT models, the base model that was mentioned in the "An Image is Worth 16x16 words" will be preferable over the others. DeIT will most likely perform slightly better then the ViT-B16 if distillation is applied using one of the CNN models, but this defeats the purpose of my experimental model where I am trying to combine features extracted form both architectures. The Swin-Transformer is an excellent model also, but all the images that will be used for training are scaled to 224×224 and the Swin-Transformer really shines when working with larger scale images, hence the standard ViT-B16 being the choice.

3.3.1 Programming Language and Framework

The language of choice for this task was Python due to it's simplicity and more importantly the availability of libraries tailored for ML and image classification. Libraries such as NumPy, Pandas, Scikit-learn and Matplotlib are just some of the tools that are used in various stages of the pipeline, mainly preparing the image data for training, validation and testing loops to come.

Python being the language used, has many DL frameworks available, but PyTorch was chosen. This framework stood out because of it's clear and well put documentation with a great community available that helps novice users. Additionally PyTorch also has the selected model architectures available. Not only these models are available but pre-trained weights can be utilised too. These pre-trained models were trained on a very large dataset known as ImageNet for extensive period of time, extracting essential features of various images.

The technical specification of the device used is provided in the Table 3.2

Model	CPU	GPU	RAM	Storage	Operating System
Razer Blade 15-B 2019	i7-9750H	GeForce GTX 1660Ti	16 GB	250 GB	Ubuntu 22.04.3 LTS

Table 3.2: Device specification

3.4 Model Training and Evaluation

3.4.1 Hyperparameters

Through careful and thorough observation of literature surrounding the topic, the most suitable hyperparameters were chosen. All images undergo augmentation where the image size is set to 224×224. They will also be normalised with the mean and standard deviation obtained from ImageNet. Only training images undergo further augmentation where a random rotation of up to 10°, and random horizontal flipping with a rate of 0.5 are applied. Most used batch sizes were 16 or 32, with the task of training with smaller computational resources the smaller batch size will be used. For the given task 30 epochs will be used for training/validation loops. The learning rate to be used with the optimiser will be set to 0.001, being the go to in the reviewed literature.

The **loss function** that will be used for multi-class classification is Cross-entropy loss, or sometimes referred to as log loss. This function will measure the difference between two probability distributions. The two probability distributions will be the true labeled data and the predicted data that a model produces. A distribution can be produced as images will be passed in batches of size 16. The function will measure how well the predicted probability distribution fits the true probability distribution by using Softmax algorithm on the logits produced by the models, converting each input into numbers between $0 - 4$, matching it with the true classes. It is calculated in the following way $L = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$, where y_o, c is the true probability, and p_o, c is the predicted. This loss function is available on PyTorch in the functional library.

The **optimiser** that will be used is Adaptive Moments Estimation with Weight Decay(AdamW). This algorithm combines advantages of two other methods known as AdaGrad and RMSProp. Main strength of it lies in maintaining the learning rate for every parameter. Weight decay is a known regularisation technique that is used to prevent overfitting, when a model fits training data too much and loses the ability to recognise unseen images. This optimisation algorithm is available on PyTorch in the optimisation library and the implementation can be viewed here[37]. It will use the learning rate that was mentioned, have *beta1* and *beta2* set to be 0.9 and 0.999, respectively. Weight decay is set to 0.01.

A **scheduler** is a technique that is used to dynamically adjust the learning rate during training. The learning rate changes on predefined conditions set, this helps to optimise the training process and improve the convergence of the optimisation algorithm. The scheduler that will be used is known as "ReduceLROnPlateau", it can be obtained from PyTorch. The learning rate is reduced when it has stopped improving after a set number of epochs known as *patience*. The scheduler options are set to the following: mode is set to *min*, as we want to minimise the loss, *patience* will be set to 5, and the learning rate will be reduced by a factor of 0.1.

The used hyperparameters and their settings can be viewed from Table3.3. All models that undergo training will use the same train, validation and test datasets. The hyperparameters shown in the table will be used for all models in order to ensure fairness when evaluated and compared. The training loop will utilise both training and validation data sets. Validation is performed immediately after one training epoch. There are 30 epochs in total but the model with lowest loss and best accuracy for validation set in an epoch will be saved, and then used in the "Model Evaluation" stage of the pipeline.

All components selected for this project do not have legal or professional constraints. PyTorch, is an open source framework licensed under Berkeley Software Distribution which provides unrestricted access to the entire codebase, including the PyTorch framework and other libraries used in the development. The Chest X-ray dataset comes with no restrictions, allowing its utilisation without any issues.

3.4.2 Transfer Learning

A method in Transfer learning called feature extraction will be used in every model. This method "freezes" all the layers in a model except the newly added dense layer that has 5 outputs. This reduces the computation costs significantly due to the reduction of learnable parameters. The "frozen"

Hyperparameter	Setting
Data Augmentation	Train/Val/Test: Resize images to 224×224 Train: Random rotation of 10° Train:Random horizontal flipping with probability of 0.5 Normalisation: *Mean* $(0.485, 0.456, 0.406)$, *Std* $(0.229, 0.224, 0.225)$
Batch Size	Train: 16 Val: 16 Test: 16
Number of epochs	30
Learning Rate	0.001
Loss function	Cross-entropy Loss
Optimiser	AdamW beta1: 0.9 beta2: 0.999 weight decay: 0.01
Scheduler	ReduceLROnPlateau mode: *min* patience: 5 factor: 0.1

Table 3.3: Hyperparameter Setting Summary

layers will utilise the learned features from ImageNet.

3.4.3 Evaluation Metrics

Training and Validation Stage

During the training and validation stages, *Loss* and *Accuracy* will be tracked with every epoch. Having a well-balanced dataset will definitely be beneficial in this context, as loss calculations will be less biased toward any specific class. These metrics will be graphically represented for each model, helping in hyperparameter tuning if necessary.

Testing Stage

Once all the models have been trained, during the testing stage, evaluation will rely on *Accuracy*, *Precision*, *Recall*, and *F1-score* metrics. Additionally, predictions made by the models will be presented in the form of *Confusion Matrices*.

4 Results & Evaluation

4.1 Breakdown

In the initial phase, all four models: ResNet50, DenseNet121, EfficientNetB4, and ViT-B16, will undergo training using both training and validation datasets. Evaluation metrics will be obtained for each model, plotted on graphs, and compared. Then these trained models will be tested using the testing dataset containing unseen data. Testing stage metrics will be gathered, followed by a comprehensive evaluation of the CNNs. Best performing model will be selected for integration into the experimental Combination Model with ViT-B16.

Once the top performing model is selected, it will be merged with ViT-B16 , and subjected to the same training/validation/testing stages. The resulting combination model will then be compared to the individual models, and an overall analysis will be conducted to provide insights into the proposed architecture. This analysis will dive into reasons why the combination model may have performed better or worse than the four standalone models.

Lastly, elements of confusion matrix are going to be used. They consist of **True Positives(TP)**, which occur when the model correctly predicts a sample that belongs to a specific class, **True Negatives(TN)**, which are not usually used in multi-class classification but in our case can be considered as predicting other classes correctly, **False Positives(FP)** occur when the model incorrectly predicts a sample to belong to a specific class, **False Negatives(FN)** work the same way as TNs, but here it can be considered as predicting other classes incorrectly.

Reason for TN and FN not being used is that because this concept is usually adapted to binary classification tasks while we have a multi-class classification task hence they are neglected or can be considered in the way described in the previous paragraph.

The explanation and definition of used metrics will be provided concretely in the upcoming subsections.

4.2 Training/Validation of 4 Models

In this stage the metrics used are Loss and Accuracy

Loss function that is used in this stage is discussed in the following section3.4.1. In summary, the objective is to minimise the loss produced by the loss function, as a smaller value signifies more accurate predictions made by the model.

Accuracy is a metric that measures the proportion of correctly classified images out of the total number of images across all classes. It is calculated in the following way $Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100$

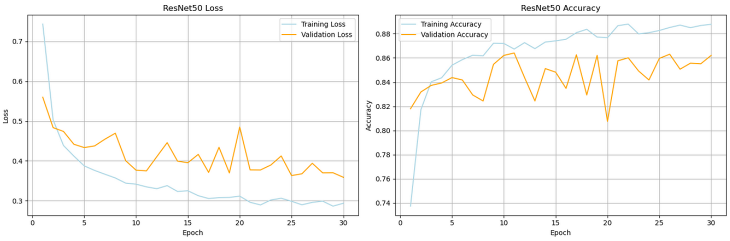

Figure 4.1: ResNet50 Loss/Accuracy

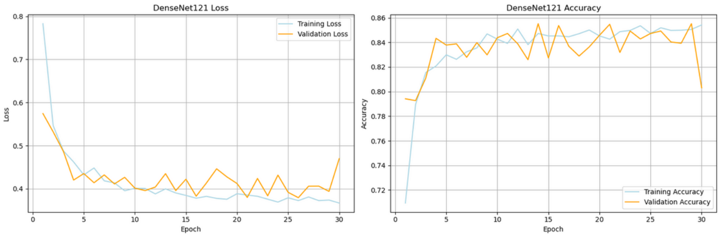

Figure 4.2: DenseNet121 Loss/Accuracy

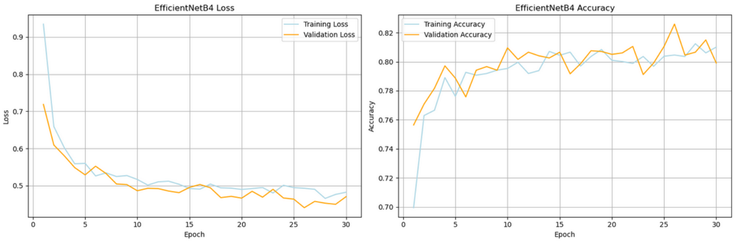

Figure 4.3: EfficientNetB4 Loss/Accuracy

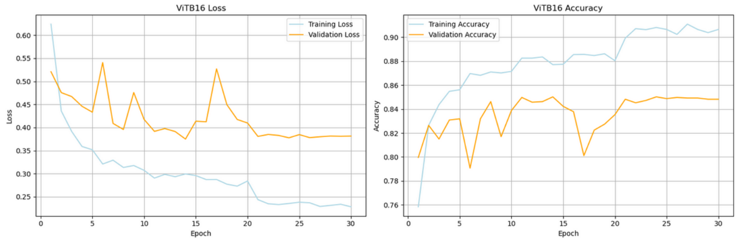

Figure 4.4: ViT-B16 Loss/Accuracy

The loss/accuracy over epochs for each model can be viewed above, where ResNet50 is 4.1, DenseNet121 is4.2, EfficientNetB4 is4.3, and ViT-B16 is4.4. It can be observed that every model's validation loss is gradually decreasing staying closely with the training loss. Same can be said about the validation accuracy of every model, it is gradually increasing with the training accuracy. Particularly DenseNet121 and EfficientNetB4 exhibit very efficient fits as the training and validation are very close and not as noisy. ResNet50 and ViT-B16 potentially would require slight architecture or hyperparameter tuning as their validation lines are quite unstable with significant fluctuations, but nonetheless the respective metrics are improving over epoch. Results can be viewed in the Table4.1.

Model	Train Acc	Train Loss	Val Acc	Val Loss	Time
ResNet50	88.8	0.2936	86.2	0.3587	77 min
DenseNet121	86.8	0.3438	85.8	0.3770	75 min
EfficientNetB4	81.5	0.4498	85.0	0.3776	81 min
ViT-B16	92.1	0.1990	85.1	0.3557	81 min

Table 4.1: Train Loss/Accuracy Results

4.3 Testing 4 models

In this stage the metrics used Accuracy, Precision, Recall and F1-score. Confusion matrices will be used to represent the TP, TN, FP, FN values.

Precision is used to measure the accuracy of only TP predictions made by the model for each class. It provides insights on how the model is able to correctly identify inputs belonging to a specific class while minimising FP. It can be calculated in the following way $Precision_C = \frac{TP_C}{TP_C+FP_C}$, where C represents a specific class. Generally high precision is desired as it would mean there is a low rate of FP.

Recall is a metric that measures the ability of the model to correctly predict all TP instances of a particular class among all instances that truly belong to that class. It can be calculated in the following way $Recall_C = \frac{TP_C}{TP_C+FN_C}$, where C represents a specific class. Generally high recall is also desired as it would mean there is low rate of FN.

Precision and Recall can be considered as compliments of each other, and the combination of both can provide a balanced assessment of model's performance. Precision focuses on minimising the FP, Recall focuses on minimising the FN.

F1-score is the combination of precision and recall represented in a single value, which provides an averaged and balanced assessment of model's performance in multi-class classification. It can be calculated in the following way $F1\text{-}score_C = 2 \times \frac{P_C \times R_C}{P_C+R_C}$, where C represents a specific class, P_C is the precision , and $R_C$is the recall for the class C.

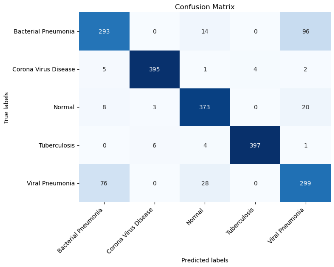

Figure 4.5: ResNet50 Confusion Matrix

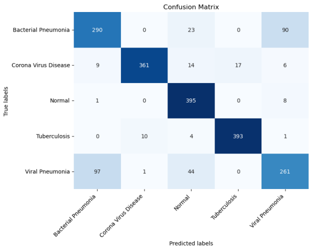

Figure 4.6: DenseNet121 Confusion Matrix

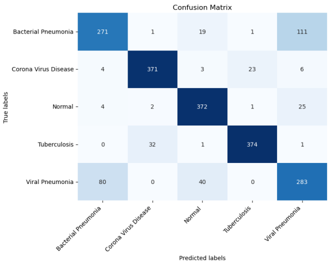

Figure 4.7: EfficientNetB4 Confusion Matrix

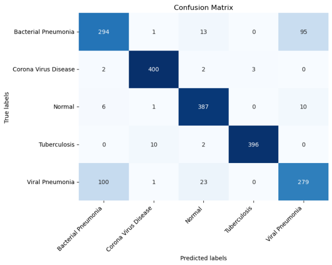

Figure 4.8: ViT-B16 Confusion Matrix

Model	**Test Acc**(%)	**Test Loss**
ResNet50	86.77	0.3522
DenseNet121	83.95	0.414
EfficientNetB4	82.52	0.4248
ViT-B16	86.72	0.3367

Table 4.2: Test Loss/Accuracy Results

Model	**Class**	**Accuracy**(%)	**Precision**(%)	**Recall**(%)	**F1-score**(%)
ResNet50	Bacterial Pneumonia	72.7	77	73	75
	Corona Virus Disease	97.05	98	97	97
	Normal	92.33	89	92	91
	Tuberculosis	97.30	99	97	98
	Viral Pneumonia	74.19	72	74	73
DenseNet121	Bacterial Pneumonia	71.96	73	72	72
	Corona Virus Disease	88.70	97	89	93
	Normal	97.77	82	98	89
	Tuberculosis	96.32	96	96	96
	Viral Pneumonia	64.76	71	65	68
EfficientNetB4	Bacterial Pneumonia	67.25	75	67	71
	Corona Virus Disease	91.15	91	91	91
	Normal	92.08	86	92	89
	Tuberculosis	91.67	94	92	93
	Viral Pneumonia	70.22	66	70	68
ViT-B16	Bacterial Pneumonia	72.95	73	73	73
	Corona Virus Disease	98.28	97	98	98
	Normal	95.79	91	96	93
	Tuberculosis	97.06	99	97	98
	Viral Pneumonia	69.23	73	69	71

Table 4.3: Results for 4 models

After undergoing the testing stage, the confusion matrices for ResNet50 4.5, DenseNet121 4.6, EfficientNetB4 4.7, and ViT-B16 4.8 were obtained, after which total Accuracy and Loss of each model were calculated and shown in the Table 4.5. All models have a good performance above 80% accuracy, but the best performing model was ResNet50 with a total accuracy of 86.77% and loss of 0.3522. The performance of ViT-B16 should also be bought to attention as it performed just almost as good as the ResNet50 and outperformed other CNN models.

Upon further investigation of how the models perform within each class, the results are displayed in the Table 4.3. It can be observed that Corona Virus, Tuberculosis and Normal classes have a high and balanced precision and recall, meaning all models are able to identify and distinguish these classes really well. For Bacterial Pneumonia and Viral Pneumonia classes all models struggle having lower precision and recall compared to other classes. This will be explored further in the Analysis section4.5.

4.4 Combination Model

4.4.1 CNN Selection

ResNet50 has highest accuracy across all classes and demonstrates the most consistent precision and recall rates, therefore it will be integrated with ViT-B16 to form the Combined Model.

The combined model will be referred to as **ResViT Model**.

4.4.2 ResViT Training/Validation

The same methodology and metrics are used as in the this subsection 4.2

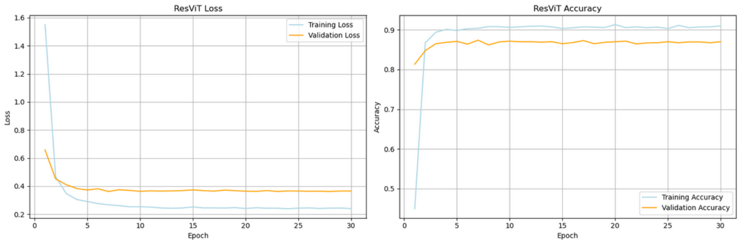

Figure 4.9: ResViT Loss/Accuracy

The loss/accuracy over epochs for ResViT can be viewed above in the Figure 4.9. It can be observed that this model's accuracy and loss start in a good position, converging just around the fifth epoch. The curves are smooth and stable, signifying that the model is efficient. The obtained results can be viewed in the Table 4.4.

Model	Train Acc	Train Loss	Val Acc	Val Loss	Time
ResViT	91.6	0.2405	87.4	0.3658	79 min

Table 4.4: ResViT Train Loss/Accuracy Results

4.4.3 ResViT Testing

The same methodology and metrics will be used as in the this subsection 4.3.

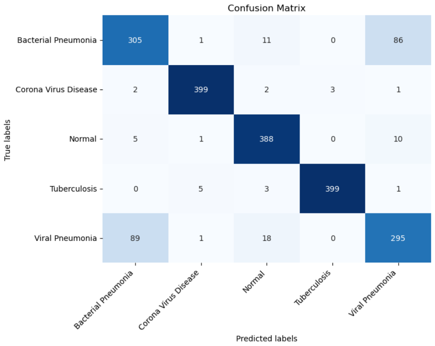

Figure 4.10: ResViT Confusion Matrix

After undergoing the testing stage, ResViT's confusion matrix can be viewed in the Figure4.10. Achieving an accuracy of 88.20% and loss of 0.3362 that are shown in the Table 4.5.

When investigating how the model performed for every class, all precision and recall values were fairly balanced, but this model also experienced lower performance when encountering Bacterial and Viral Pneumonia classes. The obtained results can be viewed in the Table 4.6

Model	**Test Acc**(%)	**Test Loss**
ResViT	88.20	0.3362

Table 4.5: ResViT Test Loss/Accuracy Results

Model	**Class**	**Accuracy**(%)	**Precision**(%)	**Recall**(%)	**F1-score**(%)
ResViT	Bacterial Pneumonia	75.68	76	76	76
	Corona Virus Disease	98.03	98	98	98
	Normal	96.04	92	96	94
	Tuberculosis	97.79	99	98	99
	Viral Pneumonia	73.20	75	73	74

Table 4.6: Results for ResViT

4.5 Analysis

EfficientNetB4 and DenseNet121 have performed the worst out of all the models, achieving overall accuracies of 82.52% and 83.95%, respectively. This outcome was unexpected, especially considering the fact that EfficientNet architectures typically outperform DenseNet in image classification tasks. These results highlight the importance of hyperparameter tuning tailored to each specific model. Fine-tuning hyperparameters for individual architectures could potentially unlock their full potential and improve overall performance.

ResNet50 and ViT-B16 have performed the best out of the 4 standalone models, having close overall accuracies of 86.77% and 86.72%, respectively, with ResNet50 being up by 0.05%. ResNet50 exhibited superior performance in identifying cases of Viral Pneumonia and Tuberculosis, whereas ViT-B16 excelled in detecting instances of Bacterial Pneumonia, Corona Virus Disease and Normal classes.

ResViT proved to be successful, surpassing the overall accuracy of both ViT-B16 and ResNet50 by 2%. Upon closer examination of specific classes, ResViT outperformed ViT-B16 by 3% in identifying cases of Bacterial Pneumonia, while lagging behind ResNet50 by only 1% in detecting Viral Pneumonia. ResViT exhibited more stable precision and recall values, with minimal differences between them, resulting in an overall superior F1-score compared to both ResNet50 and ViT-B16. These findings suggest that this type of architecture holds promise for enhanced robustness and reliability, particularly in scenarios involving multiple diseases with subtle distinctions.

When observing the Loss/Accuracy graph of ResViT4.9 it's evident that convergence occurs rather fast, typically around the fifth epoch, after which both loss and accuracy stabilise. This early convergence prompts further investigation into the potential benefits of utilising earlier extracted features of each model within the combination rather than solely relying on the output class logits. Such an approach could provide finer details about each class and potentially enhance performance. Additionally, there's scope for exploring the development of more complex and deeper combination models consisting of multiple pre-trained architectures like ResViT, but on a larger scale where the concatenated outputs undergo further processing rather than just having a dense layer. These models could serve as robust backbones, facilitating the discovery of optimal architectures that require fewer resources for training and deployment while having great accuracy.

There is a trend of significantly lower performance in all models, including ResViT, for classifying Bacterial and Viral Pneumonia, compared to other classes. After reviewing the confusion matrices of each model, it becomes evident that a big portion of misclassifications occurs between these two classes, with the model often predicting the opposite label.

Several factors may contribute to these misclassifications. The dataset itself may be a contributing factor. There is possibility that mislabelled data is present as this is a fairly small dataset that was not created under a supervision of professional radiologists, this could hinder the model's ability to detect the subtle differences between these conditions.

Another critical aspect to consider is the feature extraction process. The models examined leverage features learned from ImageNet. These features might not sufficiently encapsulate the characteristics of pneumonia. Fine-tuning the model could offer a more promising approach, allowing for the adaptation of higher level features to better distinguish between viral and bacterial pneumonia. This adjustment could be particularly beneficial given the fact that theses two conditions may appear very similar on X-ray images.

4.5.1 Limitations

One of the primary limitations of this research lies in the dataset used. While the dataset is balanced and has variety of diseases, it is comparatively small in scale and doesn't have many classes when compared with other available datasets, such as the NIH Chest X-ray dataset [38]. Upon closer examination of the dataset, significant data quality issues were uncovered, including instances where images appeared altered, smudged, or had undergone prior data augmentation. This poses challenges to the reliability of the dataset. Potential mislabeling was observed within the classes of Bacterial and Viral Pneumonia, leading to misclassification by all the models evaluated but this has to be further investigated.

While this research aimed to assess the performance of various architectures within the confines of computational limitations, it's essential to acknowledge additional constraints related to model selection and time management. Opting for more complex models could have potentially shown superior performance but the trade-off would involve longer training times and increased GPU resource requirements. Given these circumstances, I also managed other projects on my course that utilised the GPU resources of my primary device. In order to adapt to these limitations, I prioritised smaller models and imposed constraints on training epochs, potentially impacting the depth of exploration and model optimisation.

Despite these limitations, the outcomes of this research remain valuable. The evaluation and comparison of various models shed light on their performance under constrained conditions, providing insights into their strengths and weaknesses. Then the use of the proposed combination model has also shown improved performance compared to individual models, highlighting the potential of leveraging pre-trained models to a specific dataset. This success opens doors for further research opportunities, particularly in the realm of employing combined pre-trained models that employ different techniques for extracting information from images. Such investigations could lead to advancements in the field, enhancing the robustness and reliability of image classification systems for medical applications.

5 Conclusion

In this research three CNN models were tested, among which ResNet50 demonstrated the highest performance with an accuracy of 86.77%. Base Vision Transformer, ViT-B16, also performed exceptionally well, closely matching ResNet50 with an accuracy of 86.72%. The primary objective of the study was to investigate whether combining the strengths of the best CNN architecture with a vision transformer would improve performance. This objective was successfully achieved, as the combination of ResNet50 and ViT-B16 yielded an accuracy of 88.20%, representing a 2% improvement over the individual models.

The significance of this study lies in its confirmation that further research should explore the combination of multiple architectures or the development of new architectures capable of capturing features across different scales. The results demonstrate the potential of combining ResNet50 and ViT-B16. The study suggests that feature extraction techniques, while useful, may not be optimal for medical images, which demand finer feature extraction than those used in ImageNet. Fine-tuning may offer improved performance but it comes with the trade-off of increased training time and computational resources.

The main limitation hindering the further advancement of image classification in chest X-ray imaging is the availability of reliable, high-quality, and adequately sized datasets. Future research efforts should prioritise the creation of large datasets that have been thoroughly evaluated by professional radiologists multiple times before their utilisation in ML tasks. With access to very good datasets, even less complex models could demonstrate great performance in chest X-ray image classification tasks. Future research should not only concentrate on refining model architectures but also prioritise efforts to enhance dataset quality.

It is important to recognise the role of novel conventions that utilise features extracted from current SOTA architectures to witness further advancements in chest X-ray image classification. Model complexity and computational resources remain an extremely important consideration but the availability of reliable, high quality datasets are a foundation of any successful ML task.

In conclusion, I hope that these challenges will be tackled in future medical imaging research with collaborative efforts between various ML disciplines and the domain knowledge of professional radiologists.

Bibliography

[1] W. W. Labaki and M. K. Han, ‘Chronic respiratory diseases: A global view,’ *The Lancet Respiratory Medicine*, vol. 8, no. 6, pp. 531–533, 2020.

[2] S. Huang, Y.-C. Wang and S. Ju, ‘Advances in medical imaging to evaluate acute respiratory distress syndrome,’ *Chinese Journal of Academic Radiology*, vol. 5, no. 1, pp. 1–9, 2022.

[3] G. D. Schiff, O. Hasan, S. Kim *et al.*, ‘Diagnostic Error in Medicine: Analysis of 583 Physician-Reported Errors,’ *Archives of Internal Medicine*, vol. 169, no. 20, pp. 1881–1887, Nov. 2009.

[4] K. Doi, ‘Computer-aided diagnosis in medical imaging: Historical review, current status and future potential,’ *Computerized medical imaging and graphics*, vol. 31, no. 4-5, pp. 198–211, 2007.

[5] Y. Lecun, Y. Bengio and G. Hinton, ‘Deep learning,’ English (US), *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, Publisher Copyright: © 2015 Macmillan Publishers Limited. All rights reserved., ISSN: 0028-0836. DOI: 10.1038/nature14539.

[6] A. Maier, C. Syben, T. Lasser and C. Riess, ‘A gentle introduction to deep learning in medical image processing,’ *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 86–101, 2019, Special Issue: Deep Learning in Medical Physics, ISSN: 0939-3889. DOI: https://doi.org/10.1016/j.zemedi.2018.12.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S093938891830120X.

[7] F. Rosenblatt, ‘The perceptron: A probabilistic model for information storage and organization in the brain.,’ *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

[9] A.-N. Sharkawy, ‘Principle of neural network and its main types,’ *Journal of Advances in Applied & Computational Mathematics*, vol. 7, pp. 8–19, 2020.

[10] D. E. Rumelhart, G. E. Hinton and R. J. Williams, ‘Learning representations by back-propagating errors,’ *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[11] K. Fukushima, ‘Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,’ *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, ISSN: 1432-0770. DOI: 10.1007/BF00344251. [Online]. Available: https://doi.org/10.1007/BF00344251.

[12] Y. Bengio and Y. Lecun, ‘Convolutional networks for images, speech, and time-series,’ Nov. 1997.

[13] A. Krizhevsky, I. Sutskever and G. E. Hinton, ‘Imagenet classification with deep convolutional neural networks,’ *Communications of the ACM*, vol. 60, pp. 84–90, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:195908774.

[14] D. H. Hubel and T. N. Wiesel, ‘Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex,’ *The Journal of Physiology*, vol. 160, 1962. [Online]. Available: https://api.semanticscholar.org/CorpusID:17055992.

[15] 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.

[16] S. Hochreiter, ‘The vanishing gradient problem during learning recurrent neural nets and problem solutions,’ *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[17] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, ‘Densely connected convolutional networks,’ in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

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

[19] Y. Liu, Y. Sun, B. Xue, M. Zhang, G. G. Yen and K. C. Tan, ‘A survey on evolutionary neural architecture search,’ *IEEE transactions on neural networks and learning systems*, vol. 34, no. 2, pp. 550–570, 2021.

[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, ‘Mobilenetv2: Inverted residuals and linear bottlenecks,’ in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[21] A. Vaswani, N. Shazeer, N. Parmar *et al.*, ‘Attention is all you need,’ *Advances in neural information processing systems*, vol. 30, 2017.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, ‘An image is worth 16x16 words: Transformers for image recognition at scale,’ *arXiv preprint arXiv:2010.11929*, 2020.

[23] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, ‘End-to-end object detection with transformers,’ in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles and H. Jégou, ‘Training data-efficient image transformers & distillation through attention,’ in *International conference on machine learning*, PMLR, 2021, pp. 10 347–10 357.

[25] Z. Liu, Y. Lin, Y. Cao *et al.*, ‘Swin transformer: Hierarchical vision transformer using shifted windows,’ in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 10 012–10 022.

[26] G. S. Lodwick, T. E. Keats and J. P. Dorst, ‘The coding of roentgen images for computer analysis as applied to lung cancer.,’ *Radiology*, vol. 81, pp. 185–200, 1963. [Online]. Available: https://api.semanticscholar.org/CorpusID:41924.

[27] Y. Wu, M. L. Giger, K. Doi, C. J. Vyborny, R. A. Schmidt and C. E. Metz, ‘Artificial neural networks in mammography: Application to decision making in the diagnosis of breast cancer.,’ *Radiology*, vol. 187, no. 1, pp. 81–87, 1993.

[28] A. Maier, C. Syben, T. Lasser and C. Riess, ‘A gentle introduction to deep learning in medical image processing,’ *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 86–101, 2019.

[29] J. Ker, L. Wang, J. Rao and T. Lim, ‘Deep learning applications in medical image analysis,’ *Ieee Access*, vol. 6, pp. 9375–9389, 2017.

[30] P. Rajpurkar, J. Irvin, K. Zhu *et al.*, ‘Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning,’ *arXiv preprint arXiv:1711.05225*, 2017.

[31] M. M. A. Monshi, J. Poon, V. Chung and F. M. Monshi, ‘Covidxraynet: Optimizing data augmentation and cnn hyperparameters for improved covid-19 detection from cxr,’ *Computers in biology and medicine*, vol. 133, p. 104 375, 2021.

[32] D. Shome, T. Kar, S. N. Mohanty *et al.*, ‘Covid-transformer: Interpretable covid-19 detection using vision transformer for healthcare,’ *International Journal of Environmental Research and Public Health*, vol. 18, no. 21, p. 11 086, 2021.

[33] E. Verenich, T. Martin, A. Velasquez, N. Khan and F. Hussain, ‘Pulmonary disease classification using globally correlated maximum likelihood: An auxiliary attention mechanism for convolutional neural networks,’ *arXiv preprint arXiv:2109.00573*, 2021.

[34] M. Faisal, J. T. Darmawan, N. Bachroin, C. Avian, J. S. Leu and C.-T. Tsai, ‘Chexvit: Chexnet and vision transformer to multi-label chest x-ray image classification,’ in *2023 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, IEEE, 2023, pp. 1–6.

[35] Tolga, ‘Chest x-ray dataset,’ [Online]. Available: https://www.kaggle.com/datasets/tolgadincer/labeled-chest-xray-images.

[36] C. P. Lee, K. M. Lim, Y. X. Song and A. Alqahtani, ‘Plant-cnn-vit: Plant classification with ensemble of convolutional neural networks and vision transformer,’ *Plants*, vol. 12, no. 14, p. 2642, 2023.

[37] PyTorch, ‘Official adamw implementation from pytorch,’ [Online]. Available: https://pytorch.org/docs/stable/generated/torch.optim.AdamW.html.

[38] N. I. of Health, ‘Nih chest 14 x-rays,’ [Online]. Available: https://www.kaggle.com/datasets/nih-chest-xrays/data.