Multi Disease Prediction using Chest X-Rays

Submitted by

**Shubhangi Agrawal (20BCE1161)**

**Priyadharshini S (20BCE1016)**

**Shrishti Singh (20BCE1164)**

Submitted To

***Dr Jagadeesh Kannan R***

**SCHOOL OF COMPUTER SCIENCE AND**

**ENGINEERING**

****

April 4, 2023

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| **SNO** | **TOPIC** | **PAGE NO.** |
| **1.** | **Abstract** | **2** |
| **2.** | **Introduction** | **3** |
| **3.** | **Literature Survey** | **6** |
| **4.** | **Proposed Methodology** | **10** |
| **5.** | **Result** | **23** |
| **6.** | **Result Analysis** | **26** |
| **7.** | **Conclusion** | **45** |
| **8.** | **References** | **46** |
| **9.** | **Individual Roles** | **48** |

**Abstract:**

Due to the prevalence and impact of lung diseases globally, multiclass lung disease detection is an essential area of research. According to the World Health Organization (WHO), respiratory diseases account for over 10 percent of all deaths worldwide, with pneumonia, tuberculosis, and lung cancer among the primary causes. For the effective treatment and management of pulmonary diseases, accurate and timely diagnosis is essential. Nonetheless, the procedures such as X-rays and computed tomography (CT) scans, complete blood count (CBC) tests can be time-consuming, expensive, and prone to error. Detecting pulmonary diseases automatically from chest X-ray images using machine learning algorithms has the potential to enhance the diagnosis and treatment of these conditions. The ability to accurately differentiate between distinct types of lung disease is one of the major obstacles in multiclass lung disease detection. This is especially crucial for diseases, such as pneumonia and COVID-19, that may have similar radiographic characteristics. Additionally, the prevalence of multiple lung diseases in the same patient, known as comorbidity, can further complicate diagnosis and treatment. The use of machine learning algorithms, specifically CNNs, in the detection of multiclass lung diseases has a number of potential benefits, as it has the potential to enhance the speed as well as precision of diagnosis, particularly in resource-constrained settings where access to trained medical professionals is limited. CNNs have the ability to recognize patterns and features that are indicative of various lung diseases, such as the presence of opacities or consolidation in infection. The various CNNs that has been used in this research includes, ResNet50, VGG-19 network, and Inception V3, and of all the ResNet50 model performed well with the accuracy of 93.59%, which classifies five different diseases including, COVID, cardiomegaly, pneumonia, pneumothorax, tuberculosis and Normal category Future research in this area should concentrate on the development of more robust and accurate algorithms, particularly for diseases with potentially similar radiographic characteristics, and the validation of these algorithms in diverse patient populations.

**Keywords:** Multiclass lung disease detection, ResNet50, Inception V3, VGG-19, Pooling

**Introduction:**

Medical image analysis is a crucial area in the field of healthcare, with the potential to improve patient outcomes by enabling early detection and diagnosis of diseases. Chest X-rays are among the most commonly performed imaging tests in clinical practice and can provide valuable information about various respiratory and cardiovascular diseases, such as pneumonia, tuberculosis, and lung cancer. However, the interpretation of chest X-rays can be challenging, particularly for complex cases where multiple diseases coexist or where there are subtle abnormalities. Traditional image analysis techniques rely on manually crafted features, which can be time-consuming and subjective. Moreover, these methods may not be able to capture the subtle variations in the images that are indicative of specific diseases.

In contrast, deep learning techniques, particularly convolutional neural networks (CNNs), have shown great potential for automated medical image analysis. These models can automatically learn features from the images, without the need for human intervention, and have achieved remarkable success in various tasks, such as classification, segmentation, and detection.

Transfer learning is another approach that has significantly enhanced the performance of CNNs in medical image analysis. Transfer learning involves leveraging pre-trained models on large datasets to perform new tasks with limited training data. This approach has been shown to improve the performance of CNNs and reduce the need for large amounts of labeled data, which can be challenging to obtain in medical imaging tasks.Transfer learning is a technique used in machine learning where a model is trained on one task and then used for a different but related task. This technique has been successfully applied in image processing to improve the performance of computer vision models.

In image processing, transfer learning involves taking a pre-trained deep neural network that has been trained on a large dataset and using it as a starting point for training a new model on a different dataset. The pre-trained network is typically trained on a large-scale dataset such as ImageNet, which contains millions of labeled images.

The pre-trained network can then be used to extract features from the images in the new dataset. These features can then be fed into a new neural network that is trained to classify the images in the new dataset. By using transfer learning, the new network can leverage the pre-trained network's knowledge of features such as edges, textures, and shapes, which are useful in classifying images.

There are different ways to apply transfer learning in image processing, including:

1. Feature Extraction: This involves using the pre-trained network as a feature extractor. The pre-trained network is used to extract features from the images in the new dataset, which are then used as inputs to a new classifier.
2. Fine-tuning: This involves fine-tuning the pre-trained network on the new dataset. The pre-trained network is modified to adapt to the new dataset, and the weights of the network are updated during training.
3. Domain Adaptation: This involves adapting the pre-trained network to a new domain with different characteristics than the original domain. For example, adapting a network trained on natural images to medical images.

Transfer learning in image processing has many benefits, including reducing the amount of labeled data required to train a new model, improving the generalization ability of the model, and reducing the training time.

Multi-disease prediction using chest X-rays is an application of deep CNNs and transfer learning that has gained considerable attention in recent years. The goal of this approach is to develop models that can accurately and efficiently predict multiple diseases from a single chest X-ray image. These models can be useful for clinicians to quickly identify patients who may have multiple diseases, leading to earlier detection and treatment.

In recent years, the development of deep learning techniques has revolutionized the field of medical image analysis, leading to significant advancements in the diagnosis of various diseases. However, the interpretation of chest X-rays can be challenging and time-consuming, especially for complex cases where multiple diseases coexist.Several studies have demonstrated the potential of deep CNNs and transfer learning for multi-disease prediction using chest X-rays. For example, a recent study by Rajpurkar et al. (2017) developed a model that could accurately identify 14 different pathologies from chest X-rays, including pneumonia, tuberculosis, and lung cancer. The model was trained using a large dataset of over 100,000 images and achieved performance that was comparable to expert radiologists.

Deep CNNs have shown remarkable success in analyzing medical images, including chest X-rays, by automatically extracting and learning meaningful features from the images. Transfer learning, which involves using pre-trained models as a starting point for new tasks, has further enhanced the performance of CNNs in medical image analysis.The combination of deep CNNs and transfer learning has enabled the development of accurate and efficient models for predicting multiple diseases from chest X-rays. These models can aid clinicians in the early detection and diagnosis of diseases, leading to improved patient outcomes and reduced healthcare costs.

In this context, this topic is gaining increasing attention from researchers and healthcare professionals, and its potential applications are vast.

In our project we have used deep CNNs and transfer learning for multi-disease prediction using chest X-rays, including their methodology, advantages, limitations, and potential future directions. The models we have used are VGG19, Resnet50 and Inceptions V3 along with deep CNN . The models were run on batch sizes of 32 and 64 and their accuracy were reported. The models were also run for different epochs to check the deviation in accuracy. Data preprocessing was done to make the dataset fit for modeling. In data preprocessing, scaling , normalization and rgb to grayscale conversion was done to make the images fit for further model implementation.

The dataset has been splitted into test,train and validation to train the model, validate and test the model on test dataset. Literature surveys of many research papers and articles were done to get an idea about the existing work and draw a comparative analysis of the existing work and our proposed work.

**Literature Survey:**

[1] states in his paper a hybrid model for the detection of Covid-19 from chest X-ray..

A combination of the CNN and RNN is used along with Grad-CAMs approach of transfer learning. The dataset consisted of 6000 samples having 2000 for each type. It was observed that the model implemented in the paper has got better accuracy than the other already existing models. HDCNN (Hybrid convolutional neural network) has an accuracy of around 98.2 % while other models range between an accuracy of 87 % to 95 %. CNN has been used to feature extraction and sampling into sequence. The data is then given to the Recurrent Neural Network and then the transfer learning is used as an activation function for image classification. In the future course of the work, the author proposed to analyze the coronavirus through CT scan images and MRI images as well.

[2] in their paper proposed a transfer learning method to classify the cancerous and non-cancerous images. In this work, they tried to find a better combination of parameters for transfer learning from CheXNet . Their study showed that transfer learning from CheXNet for CNN can be effective to learn from the DDSM (Digital Dataset for Screening Mammography) dataset. The authors focused on the fact that since both the datasets consisted of X-ray images but they were from different body parts, thus they wanted to find out if transfer learning could be helpful to learn from DDSM. After trying various configurations , it was found that using 2 dense blocks, the accuracy achieved was the best. It was around 91% on training and 88% on validation dataset. 6 layers gave the best accuracy among 1 to 12 layers tried with a measure of about 98% for training and 90% for validation dataset. A learning rate of e^-2 gave the best result. In the future course of this work, the authors suggested using gridsearch CV or random search CV for parameter hypertuning.

[3] in their proposed a light architecture of transfer learning model from CNN for skin lesion classification. A multinomial classification was done on the taken dataset without applying any preprocessing and image augmentation A comparative analysis was made between the two models i.e. a normal CNN and a transfer learning model using CNN. It was found that the transfer learning model performed better than the conventional CNN with an accuracy of around 86% while the traditional CNN was able to achieve only 68% accuracy. A five-fold cross validation method was used on the dataset. In the future course of this work, the authors aim at obtaining better results by implementing image pre processing and data augmentation.

[4] In recent years, the use of deep learning networks for the multi-disease diagnosis of chest X-rays has received considerable attention. Using real-world data, Chen et al. sought to improve the accuracy of these networks in their study. Studies on the formulation and evaluation of deep learning models for multi-disease diagnosis of chest X-rays were done and ResNet50 with weighted binary cross-entropy loss served as the baseline for the research, which addressed the degradation issue and introduced the Bottleneck structure to reduce computational consumption. Utilizing cutting-edge deep learning networks, such as EfficientNet and CoAtNet, with various loss functions and activation functions may improve the accuracy of multi-disease diagnosis in chest X-rays, according to the study. These networks have demonstrated efficacy and sophisticated representation learning capabilities, and they reduce the need for extensive reimplementation. CoAtNet-0-rw, a CNN+Transformer hybrid network, improved multi-disease diagnosis in a long-tailed dataset compared to EfficientNet.

[5] CXR imaging is a widely used diagnostic instrument for detecting a variety of diseases; however, even for expert radiologists, accurately diagnosing diseases from CXR samples remains a challenge. To resolve this, transfer learning classification approaches were evaluated on a publicly available CXR image dataset. Synthetic minority over-sampling technique (SMOTE) and weighted class balancing are used to mitigate the effects of class imbalance, and a hybrid Inception-ResNet-v2 transfer learning model with data augmentation and image enhancement provides the highest accuracy. The technique proposed by Chandra et al. is deployed in an edge environment utilizing Amazon IoT Core to automate disease detection in CXR images with three categories, pneumonia, COVID-19, and normal. The proposed technique outperforms other TL models, including SqueezeNet, VGG19, ResNet50, and MobileNetV2, according to comparative metrics.

[6] Due to the high mortality rates associated with pneumonia, COVID-19, tuberculosis, and pneumothorax, this article discusses the need for automated classification of lung diseases. The authors propose classifying these diseases in chest X-ray images using convolutional neural networks (CNNs) that have been pre-trained. The dataset included CXR images from four distinct lung diseases, including pneumonia, COVID-19, pneumothorax, and tuberculosis, as well as CXR images from healthy individuals. For the classification of the CXR images, eight CNNs were utilized, including Alexnet, Darknet-19, Darknet-53, Densenet-201, Googlenet, InceptionResnetV2, MobilenetV2, and Resnet-18. During both the training and classification phases, the dataset underwent colour preprocessing, resizing, and data enrichment. The Adam optimizer with a maximal epoch of 30 and a mini-batch size of 32 was used for training. With Densenet-201 and K=5, the proposed method obtained the highest accuracy of 97.2%, indicating that it outperformed the existing state-of-the-art methods. However, additional validation with a larger dataset and application in the actual world is required to confirm the efficacy of the proposed method.

**[7]** Dina M. Ibrahim et al present a deep learning-based model for diagnosing COVID-19, pneumonia, and lung cancer using chest X-ray and CT images. The proposed model comprises three stages, namely data pre-processing, deep learning models for feature extraction, and classification. Four architectures, namely VGG19-CNN, ResNet152V2, ResNet152V2 + Gated Recurrent Unit (GRU), and ResNet152V2 + Bidirectional GRU (Bi-GRU) were considered for evaluation. The results showed that the VGG19 +CNN model achieved the highest accuracy of 98.05%, outperforming the other models. The proposed model can help diagnose chest diseases accurately, reducing decision time and the burden of using several applications for detecting each disease separately. The study provides insight into the use of deep learning-based systems for disease diagnoses and emphasizes the importance of chest images in the early detection of COVID-19.

[8] Nahiduzzaman M et al. propose a lightweight convolutional neural network (CNN) named ChestX-ray6, which can automatically detect multiple diseases from digital chest X-ray images, including pneumonia, COVID19, cardiomegaly, lung opacity, and pleural effusion. The model achieved an accuracy of 80% for the detection of six diseases and 97.94% accuracy for binary classification of normal and pneumonia patients. The study combined several datasets for more variations and created a multiclass environment, used augmentation to balance the datasets, and compared the model's performance with different transfer learning approaches, such as VGG19, ResNet50, DenseNet121, and MobileNetV2. The ChestX-ray6 model outperformed the state-of-the-art models in terms of accuracy, precision, and recall measures. The study used Pycharm Community Edition and Keras with TensorFlow as the backend, and a computer with an Intel(R) Core(TM) i7-6700 CPU @3.40 GHz processor and 32 GB RAM, a NVIDIA GeForce GTX 1650 SUPER 4 GB GPU on a 64-bit Windows 10 Pro operating system for training and testing the model.

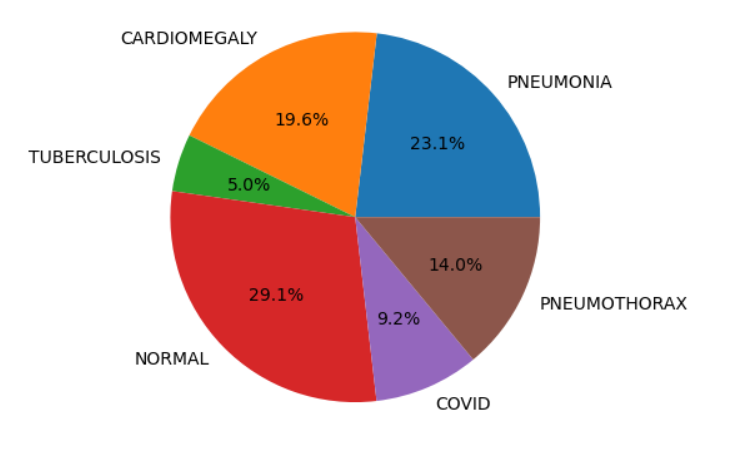
[9] S. Murali et al. compared the performance of pre-trained deep convolutional neural network (CNN) models, VGG-19 and ResNet-50, with a fine-tuned CNN model trained from scratch, called Iyke-Net, for pneumonia detection in chest X-ray images. The study concludes that pre-trained models, with proper fine-tuning, perform comparably to the Iyke-Net model. The authors used data augmentation and dropout regularization to reduce overfitting which improved their generalization performance. The highly imbalanced dataset with more pneumonia cases than normal cases was balanced using data augmentation, which eliminated the possibility of overfitting the model. Specifically, the fine-tuned VGG-19 and ResNet-50 models achieved a recall of 92.03%, which was only slightly lower than the recall of 94.50% achieved by Iyke-Net.

**Proposed Methodology:**

The sources of this research is mainly secondary data which include published research papers, articles and other websites. In depth analysis from these secondary sources has been done for the findings of this research paper.

**1. Dataset description**

The dataset used for the model used in this research is extracted from Kaggle which is into 6 folders for the 5 diseases and normal category scans There are 13,900 X-Ray images (JPEG, PNG & JPG) and 6 categories (Cardiomegaly,Covid,Normal, Pneumonia, Pneumothorax, Tuberculosis ). The distribution of each category is as depicted below:



The link of the dataset used is given below:

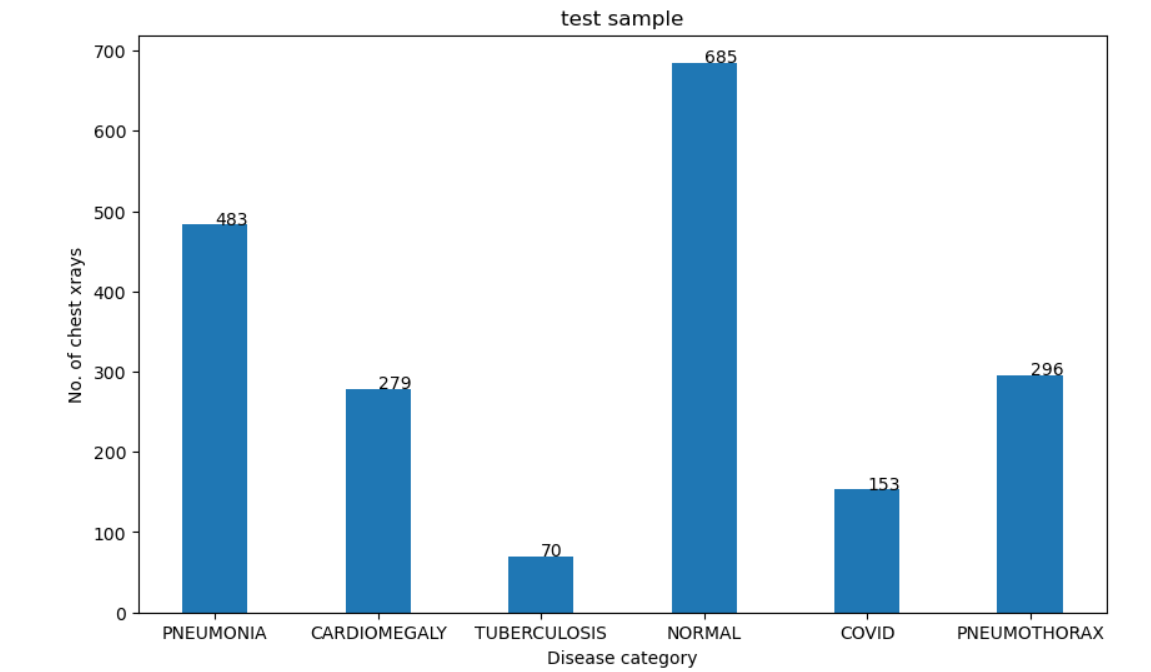
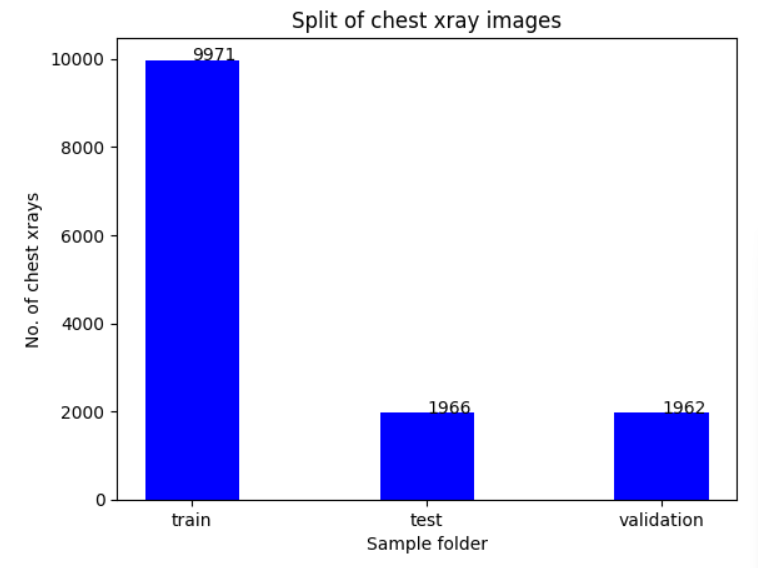
<https://www.kaggle.com/datasets/baro1502/chest-xray>

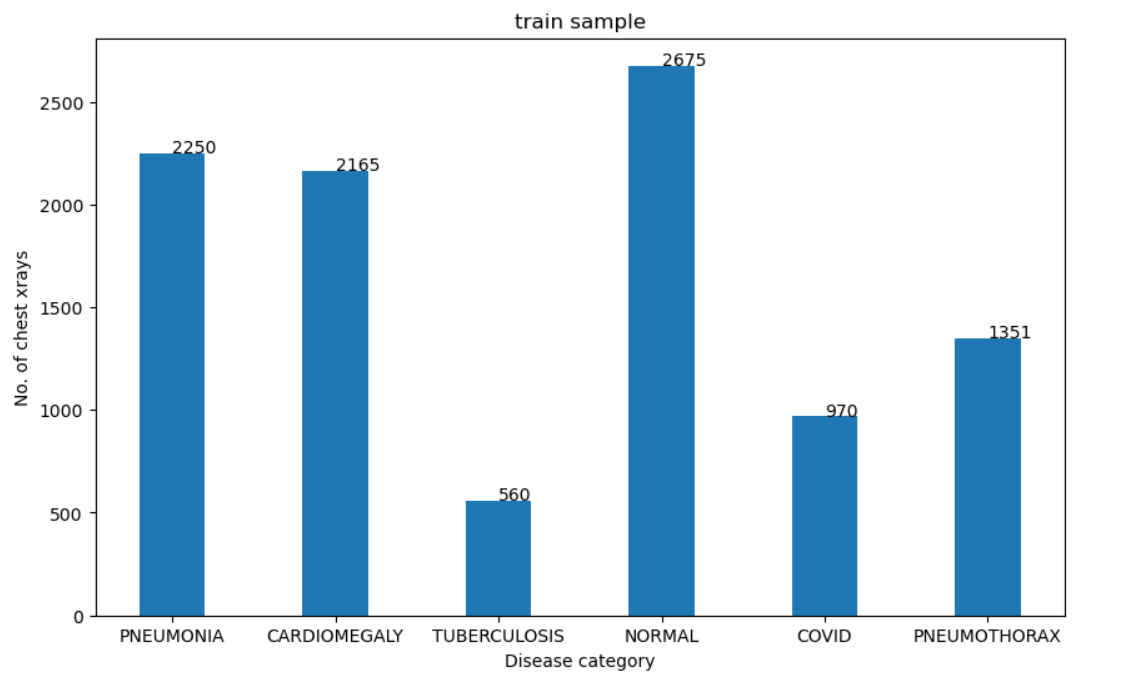
**2. Dataset Splitting**

The chest x-ray image dataset is split into train, test and validation, as represented below.

Training set contains 9971 images, testing set contains 1966 image and the validation set contains 1962 images each belonging to 6 classes after the split.The validation set is typically used to adjust the hyperparameters of the model, such as the learning rate or regularization strength, and to perform early stopping to prevent overfitting.After splitting the data, a new kaggle repository is created, whose link is: <https://www.kaggle.com/datasets/priya1543/image-processing>

The test, train and validation folders contains the following distribution of classes:





**2. Data Preprocessing**

1. Rescaling

We will be training a VGG-19 model on our custom training dataset to classify among the six categories - Cardiomegaly, Covid,Normal, Pneumonia, Pneumothorax, Tuberculosis. The pre-trained CNN model VGG19 inputs a color image of dimensions 224×224, while InceptionV3 requires input dimensions to be 299x299, on the other hand ResNet requires the input image of size 300x300. However, all the images of the dataset are of different dimensions. Hence, they were all resized to the required dimension, for the different models.

1. Normalisation

Normalization has also been performed on the data so that the effect of illumination can be reduced from the X-ray scanned images. The images belonging to both the classes, i.e., “Pneumonia” and “Normal” were visualized, which made it clear that the data is imbalanced, so we used image augmentation to get a better distribution of data and prevent overfitting and thus making our model more robust.Also, to lessen the impact of variations in illumination, a grayscale normalization is carried out. Additionally, [0..1] data allows the CNN to converge more quickly than [0..255].

1. RGB to Grayscale conversion

The images were converted to Grayscale,to reduce the complexity of the image by removing color information and representing the image in a simpler, more compact form. This can make it easier to process and analyze the image, particularly in situations where color information is not necessary or relevant. In the case of chest x-rays, converting them to Grayscale can make it easier for doctors and medical professionals to analyze the images and identify any abnormalities or medical conditions.

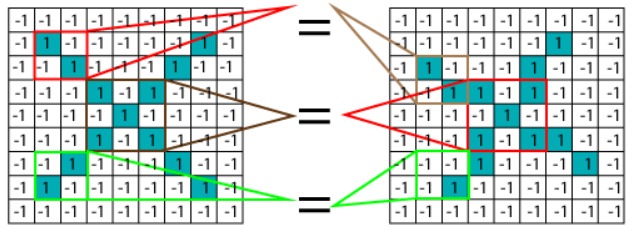
**3. Algorithm**

The algorithm which has been implemented in the prediction model is CNN, i.e. , Convolutional Neural Networks having the layers - Convolutional, ReLU Layer, Pooling and Fully Connected Layer, which has been built from scratch and other DeepCNN pre trained models have also been used which are - VGG19, Resnet50, InceptionV3.

**DeepCNN:**

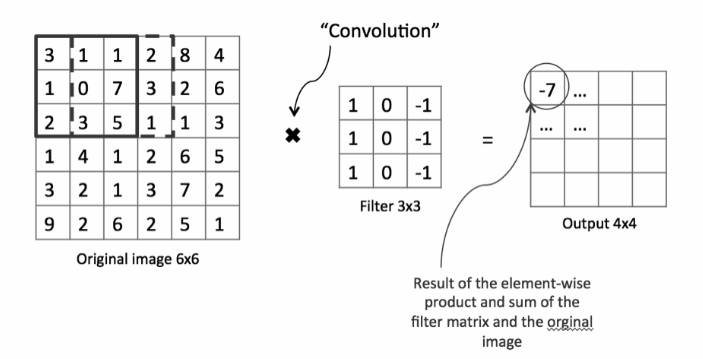
1. Convolutional Layer:

We know that the computer understands images in terms of pixels, but if the model classifies based on each pixel, if there is some distorted image belonging to a particular class, which in this project is obvious to exist, it’ll not be able to recognize the class. For this reason, we use filters which are basically small patches of images. These filters are used in CNN, with the help of which it is able to recognize the pattern by checking similarity based on filters or features or kernel.

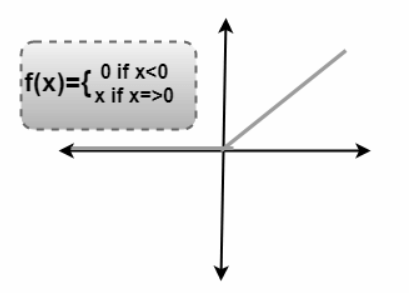


Source : https://www.javatpoint.com/working-of-convolutional-neural-network-tensorflow

So, as we see here the convolutional matches the filters and finds similarity between the images to classify it whether it’s Pneumonia or Normal. It basically first multiplies the corresponding pixel values and adds and divides by the total number of pixels. Now, by keeping track of the feature , we create a map and put an amount of filter at that place.



Further, the filter is slid throughout the image by certain strides, here we have taken strides of 1 as well as 2 and this whole process of filtering is performed again, to check how the features match the area and then the same convolution is applied with every other filter to get the activation maps in the output. Additionally, to solve the border problem, i.e., since the border of the images will be strided less times than the parts of images in the middle, padding is done, i.e., additional columns and rows of zeros are added. Here, padding of “SAME” is used to ensure that the input image is completely covered by the filter and specified stride; it applies padding to the input image. Because the output for stride 1 will be the same as the input, it is named SAME.



1. ReLU Layer

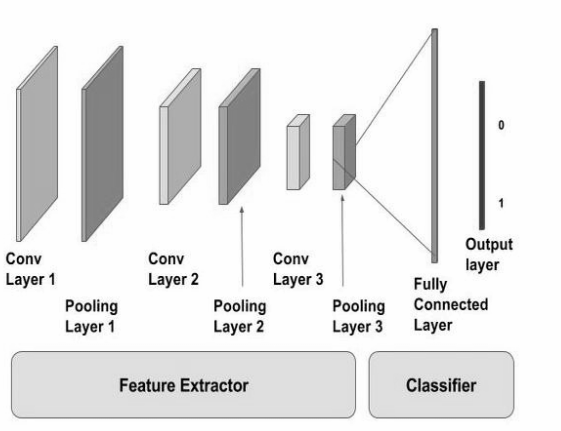
The work of Rectified Linear unit layer is basically to remove all the negative values from the filtered images, i.e. , the output we get in the convolutional layer and replace them with zeroes, so that the values don’t get added up to zero because of those negative values.

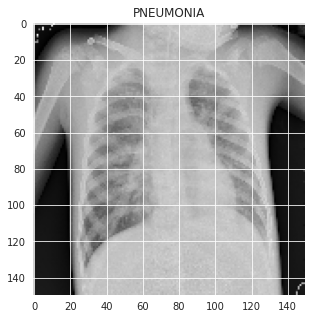
1. Pooling Layer

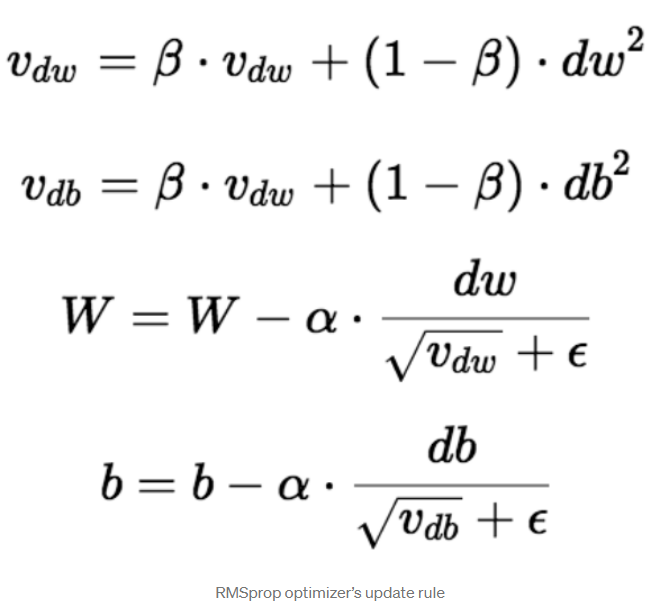
Typically, a pooling layer is added in between two convolutional layers that follow one another. By downsampling the representation, the pooling layer lowers the computation's complexity and number of parameters. The average or maximum pooling function is available. In this model, max pooling is employed because it performs better.

1. Fully Connected Layer

The final layer of the network is fully connected, which means that every neuron in the layers before it is connected to every neuron in the layers after it. This is done by a special reshaping procedure known as flattening which converts multidimensional layers into a one-dimensional fully connected layer.







**4. Optimizer’s used:**

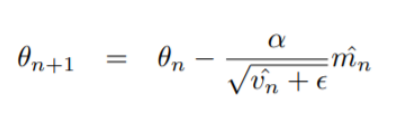
Two optimizers namely RMSprop optimizer and Adam’s Optimizer have been used in the model to check which gives better performance to choose the final model.

1. RMSprop Optimizer:

This implements the RMSprop algorithm.The basic idea behind RMSprop is to keep a moving (discounted) average of the gradient's square and divide it by its root.The vertical oscillations are constrained by it. As a result, we can speed up learning and our algorithm could take more substantial steps in the horizontal direction. The update rule for the RMSprop optimizer is as indicated on the side.

1. Adam’s Optimizer:

Adam is a specially developed adaptive learning rate optimization algorithm for deep neural network training.The Adam optimizer combines the following two methods of gradient descent: By taking into account the "exponentially weighted average" of the gradients, this algorithm is used to speed up the gradient descent algorithm. The algorithm converges faster towards the minima when averages are used.



**5. Architecture of the models**

**Inception V3 Architecture:**

Inception v3 is a popular convolutional neural network (CNN) architecture that is often used in transfer learning for image recognition tasks.

One of the key features of Inception v3 is its use of "inception modules," which are sets of convolutional layers with varying filter sizes that are concatenated together to capture features at different scales. This allows the model to learn more complex and diverse features compared to traditional CNN architectures.

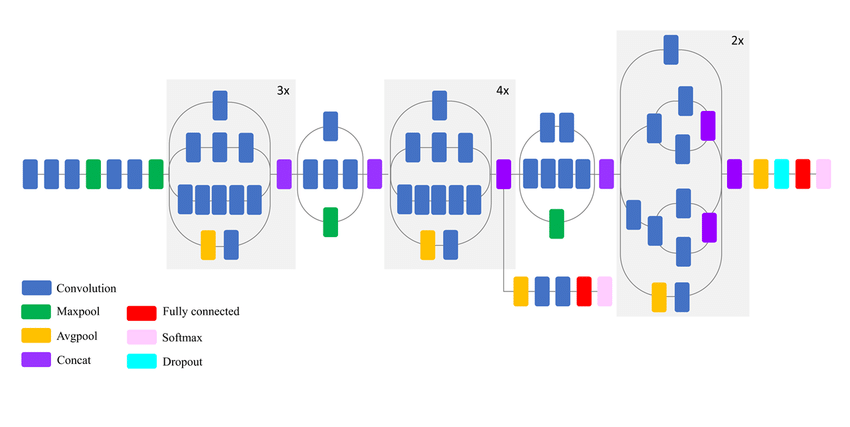
In addition, Inception v3 also includes other techniques such as batch normalization, factorized convolutions, and aggressive data augmentation during training, which help to improve the performance of the model and reduce overfitting.

The Inception v3 architecture is composed of multiple layers of convolutional and pooling operations, followed by fully connected layers. The key innovation of this architecture is the use of multiple "Inception modules" that allow for efficient information processing and dimensionality reduction within the network.

An Inception module is a multi-branch convolutional network that combines feature maps generated by filters of different sizes. This helps to capture features at multiple scales and allows the network to effectively process both local and global features. Each Inception module is composed of a set of parallel convolutions with different filter sizes (1x1, 3x3, and 5x5), as well as a max pooling layer. The outputs of these layers are then concatenated along the depth axis and fed into the next layer.

In addition to the Inception modules, the Inception v3 architecture also includes a number of auxiliary classifiers that are added to intermediate layers of the network. These classifiers help to regularize the network during training and improve its performance.

Architecture Diagram:



Source:https://www.researchgate.net/figure/Schematic-diagram-of-InceptionV3-model-compressed-view\_fig6\_326421398

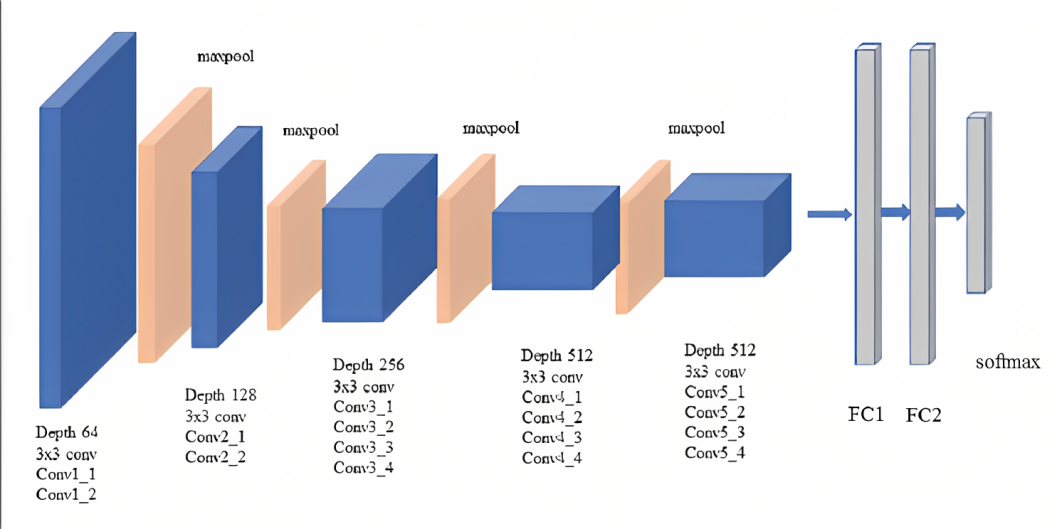
**VGG19 Architecture:**

The VGG19 model is composed of 19 layers, including 16 convolutional layers and 3 fully connected layers. The convolutional layers are grouped into 5 blocks, where each block consists of multiple convolutional layers followed by a max pooling layer. The number of filters in each convolutional layer is fixed to 64, except for the first convolutional layer which has 3 channels to match the RGB input images.

The first two blocks of convolutional layers have 2 convolutional layers each, while the remaining 3 blocks have 4 convolutional layers each. The max pooling layers have a fixed size of 2x2 and a stride of 2, which reduces the spatial dimensions of the feature maps by a factor of 2 after each block.

The three fully connected layers at the end of the network have 4096, 4096, and 1000 neurons, respectively. The output of the final layer represents the probabilities for each of the 1000 classes in the ImageNet dataset. The last layer of the model has been fine tuned to detect the 6 classes of disease.

Overall, the VGG19 model has a total of approximately 138 million parameters, which makes it a relatively large model compared to other CNN architectures.



**ResNet Architecture:**

ResNet's architecture is built on the idea of residual learning, which learns the residual function rather than the actual mapping. By utilizing shortcut connections, which enable the input to skip one or more levels and go straight to the output, ResNet accomplishes this. ResNet's design can be seen as a collection of residual blocks, where each block houses a stack of convolutional layers connected by shortcuts.

Convolutional layer:

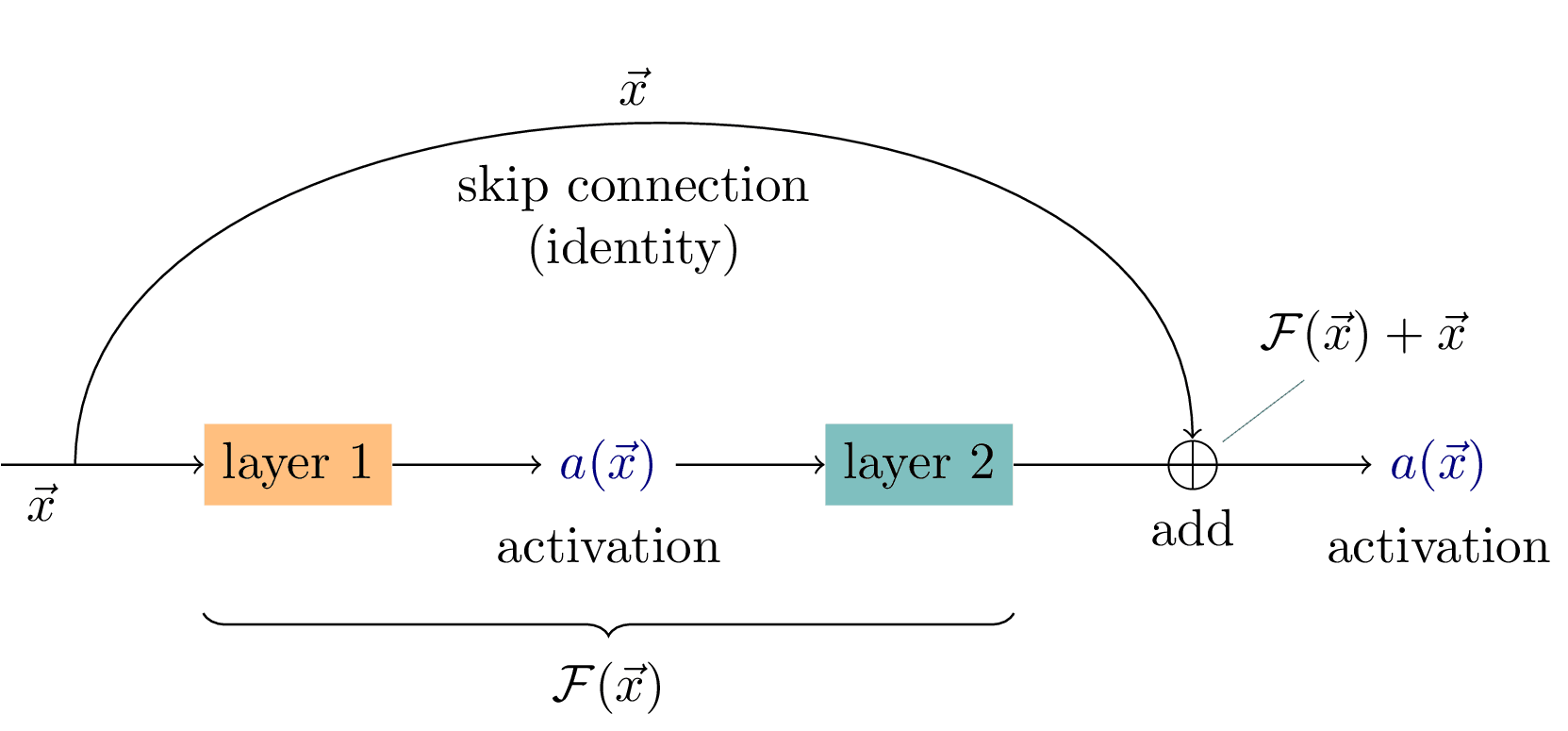
The first layer of the ResNet50 architecture is a 7x7 convolutional layer with stride 2, followed by a max pooling layer. The remaining layers are separated into four phases, each of which consists of multiple residual blocks.

MaxPooling layer:

Following the first convolutional layer, this layer conducts max pooling on the output of the convolutional layer. This layer reduces the spatial dimensions of the features while preserving the essential data.

Residual block:

ResNet50 architecture incorporates residual blocks, which consists of two 3x3 convolutional layers with batch normalization and ReLU activation, followed by an identity mapping, and enable the creation of extremely deep networks. A residual block includes at least two convolutional layers with shortcut connections. The output of the residual block is the sum of the output of the two convolutional layers and the input of the residual block. This skip connection enables gradients to flow directly through the block, that is the connection enables the input of the block to be directly appended to the block's output which helps mitigate the problem of vanishing gradients that can arise in extremely deep networks.



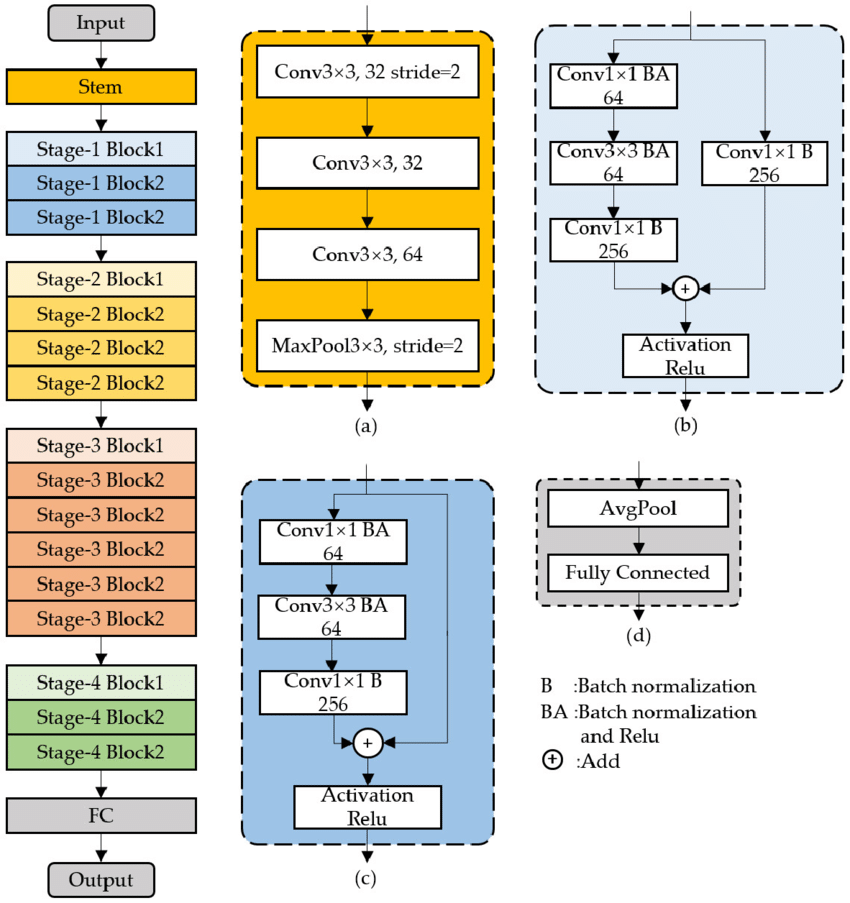
The given diagram represents the skip connection, where the output for the particular block takes the output of the layers, as well as the input given to the block, which is the principle of identity mapping, this in turn reduces the problem of reducing gradients in ResNet.

Fully Connected Layer:

ResNet50's final layer is a fully connected layer that takes the output of the final residual block and classifies it into the desired number of classes.

Softmax Layer:

This layer is applied after the fully connected layer and transforms the fully connected layer's output into a probability distribution over the classes. As the predicted output of the network is the class with the highest probability.



The above diagram represents ResNet50 architecture, where figure a) stem diagram;

b) Stage1-Block1; (c) Stage1-Block2; (d) FC-Block.

**Results:**

The results obtained for various CNN models are mentioned below.

***Table 1.0***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Method** | **Optimizer used** | **Batch size** | **No. of Epochs** | **Training accuracy** | **Test accuracy** |
| ResNet50 RGB | Adam | 32 | 30 | 96.44 | 93.03 |
|  |  |  |  |  |  |
| Inception V3 | Adam | 32 | 20 | 98.32 | 90.69 |
| VGG-19 | Adam | 32 | 10 | 97.12 | 91.65 |
| VGG-19 | Adam | 32 | 15 | 97.98 | 91.14 |
| VGG-19 | Adam | 64 | 20 | 97.79 | 92.17 |
| VGG-19 | RMSprop | 32 | 10 | 95.91 | 91.71 |
| VGG-19 | RMSprop | 64 | 12 | 92.13 | 87.79 |
| CNN | Adam | 32 | 30 | 94.04 | 87.74 |

**Interpretation of the Table 1.0:**

The dataset was separated to training, test and validation in 80:10:10 split, and the batch size for the model used is 32 and 64. The models such as ResNet50, VGG-16, Inception V3, and CNN were implemented, and the images were trained on both RGB and Grayscale for ResNet50 as it gave the highest accuracy of all. For pretrained models such as ResNet50 and VGG-16, the classification accuracy ranges between 92% and 93%, whereas for Inception V3 and the CNN network developed, the accuracy ranges between 89% and 85%. The ResNet50 model achieves the highest accuracy of 93%.

ResNet50 (RGB): When trained with images in RGB scale, ResNet50 achieved a training accuracy of 97.39% and a test accuracy of 93.59%. This model outperformed other CNN models in terms of accuracy and is an effective model for image classification tasks.

InceptionV3: When trained with images in RGB scale, InceptionV3 achieved a high training accuracy of 98.32% but had a lower validation accuracy of 89.97%.The test accuracy was around 90.69%. This suggests that the model may have overfit the training data and may not generalize well to new data.

CNN: When trained with images in RGB scale, the CNN model achieved a training accuracy of 94.04% and a test accuracy of 87.74%. Although this accuracy is lower than the ResNet50 models, it still performed well in image classification tasks. However, further experimentation and analysis may be necessary to improve the model's accuracy.

VGG-19: The testing accuracy of the VGG-19 model has been evaluated using two different optimization algorithms: Adam and RMSprop. Both of these optimization algorithms are commonly used in training deep learning models.

The results of the experiments show that the highest testing accuracy of 92.17% was achieved with the Adam optimizer and a batch size of 64 when trained with 20 epochs, where early stopping at 13 is observed. This means that the model was able to correctly classify 92.17% of the test images that it was presented with. This is a good result and indicates that the model is performing well on the dataset.

When the model was trained with a batch size of 32 and the Adam optimizer, an accuracy of 91.14% was achieved after 15 epochs of training. However, an early stopping was observed at 12 epochs, which means that the model had already converged and there was no significant improvement in accuracy beyond this point. When the model was trained for only 10 epochs, an accuracy of 91.65% was achieved, which is slightly higher than the accuracy obtained with 15 epochs. This suggests that the optimal number of epochs for this dataset with the Adam optimizer and batch size 32 is around 10-12.

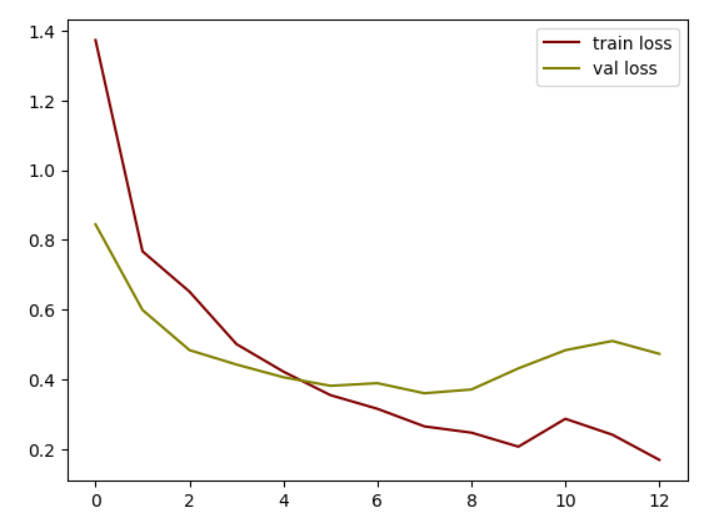
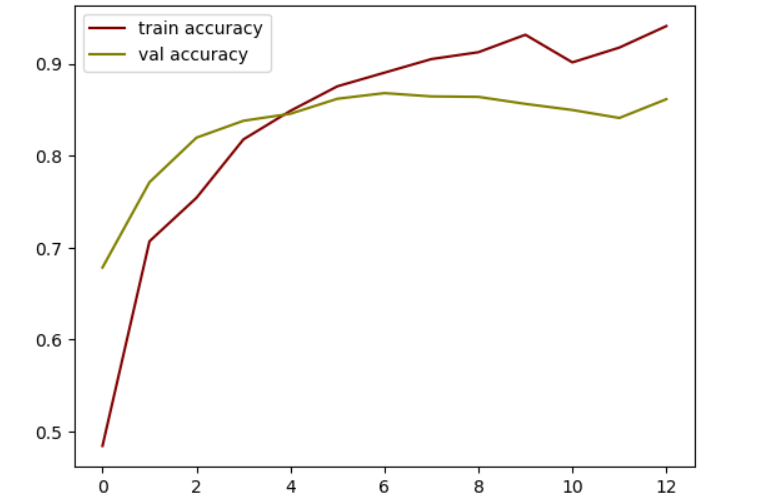
To evaluate the performance of the model with the RMSprop optimizer, the model was trained with both batch sizes of 32 and 64. The results show that a batch size of 32 achieved a higher accuracy of 91.71%, while a batch size of 64 achieved a lower accuracy of 87.79%. This indicates that the RMSprop optimizer is not performing as well as the Adam optimizer on this dataset.

The results of the experiments suggest that the Adam optimizer is more effective than the RMSprop optimizer on this dataset, and that a batch size of 64 is optimal for training the model. The optimal number of epochs for this dataset with the Adam optimizer and batch size 64 is around 13.

**RESULT ANALYSIS:**

**ResNet50:**

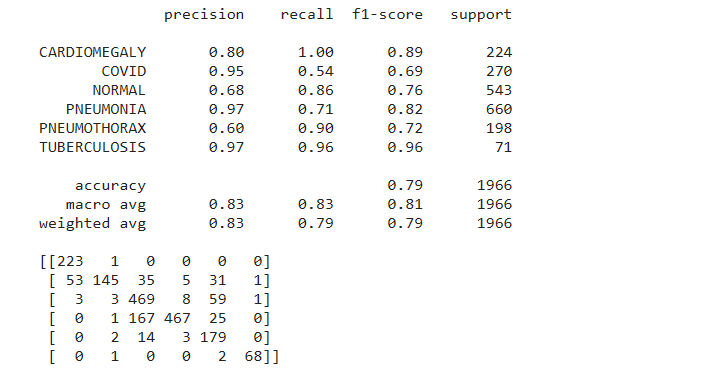
1. **RGB images:**

****

***Figure A: Plot of accuracy of train and validation Figure B: Plot of loss of train and validation***

The figure A represents the plot of training and validation accuracy of ResNet50 when trained with images in RGB scale, and figure B represents the training and validation loss. The training accuracy of the model is 97.39 and the test accuracy is 93.59.

The classification report and the confusion matrix obtained for the test data is mentioned below.



***Figure C: Classification report***

****

***Figure D: Confusion matrix***

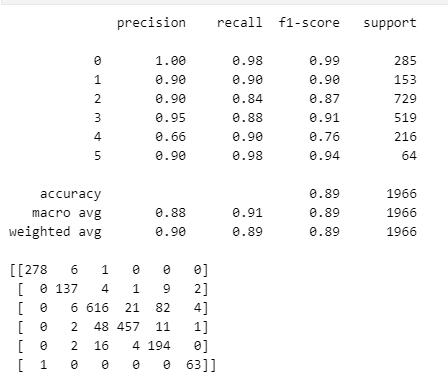
The resnet classification model when implemented for grayscale images also resulted in the accuracy range of 93%.

**Inception V3**

|  |  |
| --- | --- |
| ***Figure A: Plot of train and validation accuracy*** | ***Figure B: Plot of train and validation loss*** |

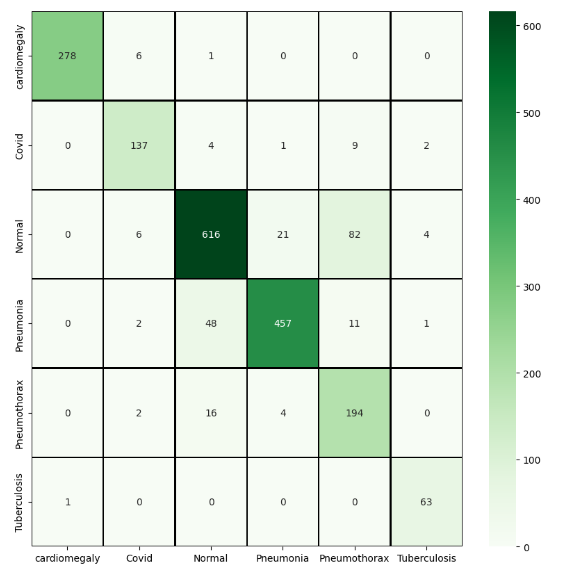
The figure A represents the plot of training and validation accuracy of InceptionV3 when trained with images in RGB scale, and figure B represents the training and validation loss. The training accuracy of the model is 98.32% and the validation accuracy is 89.97%.

The figure B represents the plot of training and validation loss of Inception V3. The train loss decreases and the validation loss also decreases with increasing epoch.

****

***Figure C: Represents the classification report***

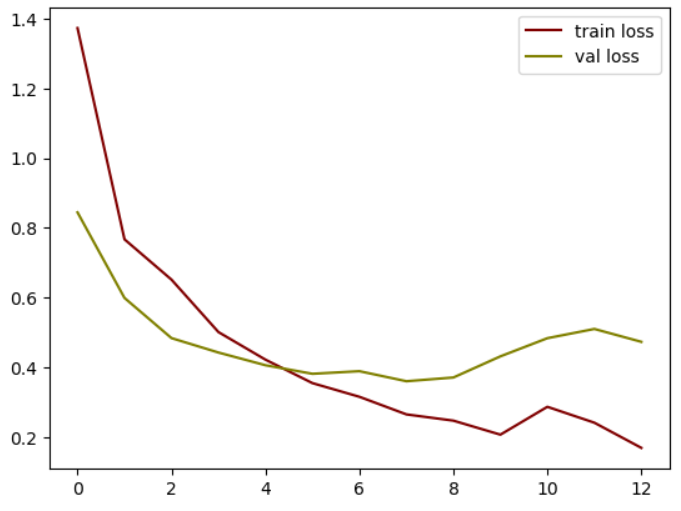
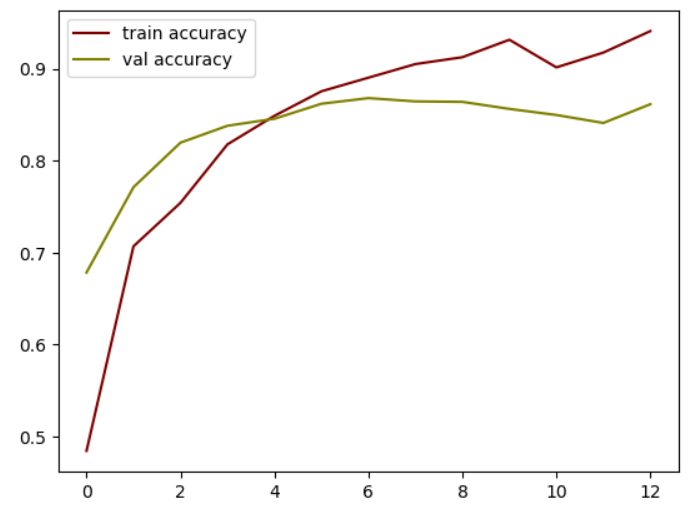
Figure C represents the class wise accuracy and it has been found that the accuracy is highest for class 0 i.e. cardiomegaly. The accuracy recorded is around 99%. The least was for class 4 i.e. Pneumothorax, which was found to be around 76%. The average accuracy came to be 89%.

****

***Figure D: Visual representation of Confusion Matrix***

The above figure represents the visualization of the confusion matrix.

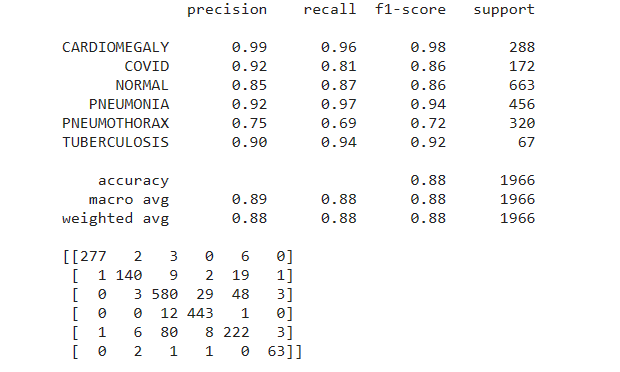
**CNN:**

****

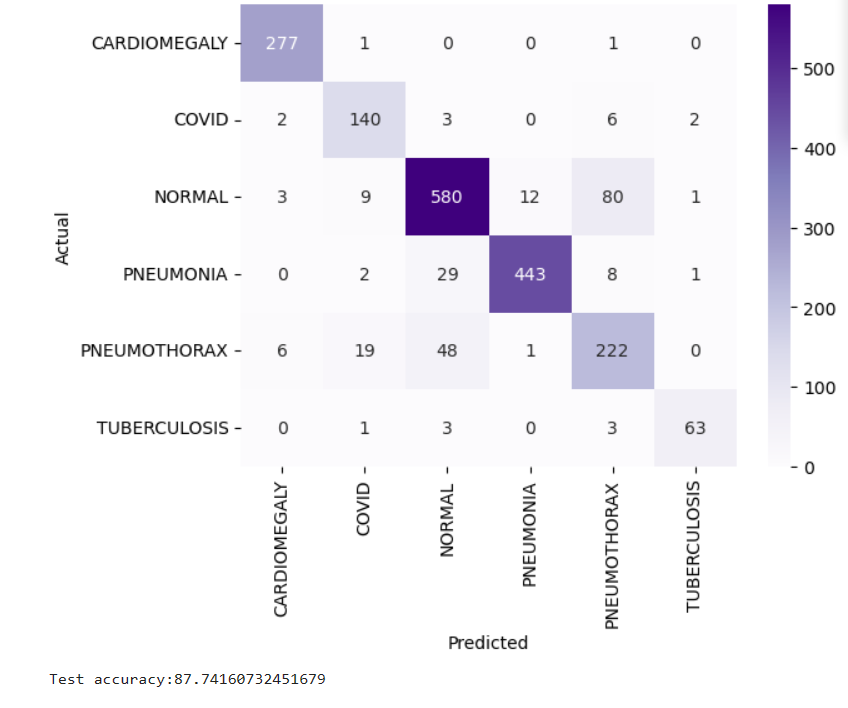
***Figure A: Plot of train and validation accuracy Figure B: Plot of loss of train and validation***

The figure A represents the plot of training and validation accuracy of CNN when trained with images in RGB scale, and figure B represents the training and validation loss. The training accuracy of the model is 94.04 and the test accuracy is 87.74.

The classification report and the confusion matrix obtained for the test data is mentioned below.

****

***Figure C: Classification report***

****

***Figure D: Confusion matrix***

**VGG19:**

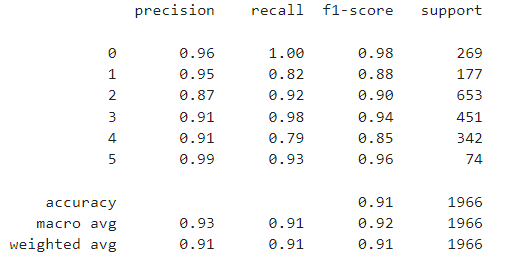
1. **Optimiser : Adam, Batch size: 32**

**15 epochs:**

|  |  |
| --- | --- |
|  |  |

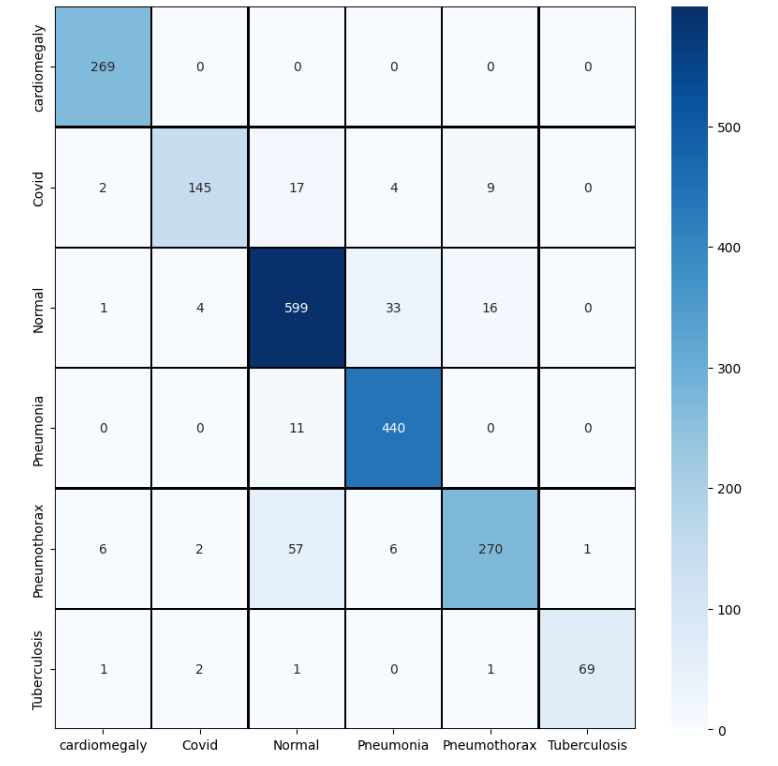
***Figure A: Plot of train and validation accuracy Figure B: Plot of loss of train and validation***

The above graph shows the accuracy obtained against every epoch. The accuracies showed high deviations when the epochs were less in number as a lesser number of epochs give unreliable results and are highly undesirable. As it can be seen, the validation accuracy reaches a maximum when the epoch is around 11, and it again starts showing deviation. The model was initially run for 15 epochs , but it stopped at 13 epochs because of early stopping to avoid overfitting.

****

***Figure C:Classification Report***

We can see the classification report above in which most of the classes has achieved accuracy above 90% while Covid 19 has a little less accuracy of 88% and Pneumothorax having only 86% accuracy.

****

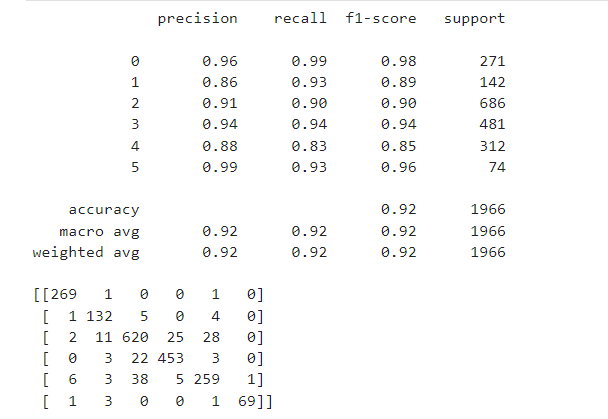
***Figure D: Confusion Matrix***

The results predicted from the above model can be viewed in a confusion matrix which in turn is transformed to a heatmap.

**10 epochs:**

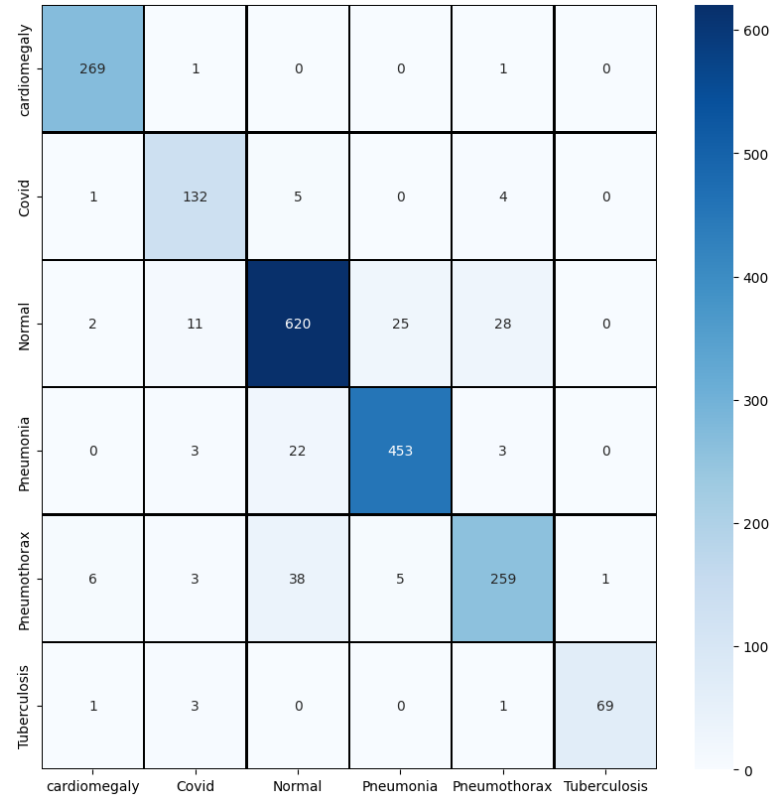
|  |  |
| --- | --- |
| ***Figure A: Train and validation accuracy*** | ***Figure B: Train loss and validation loss*** |

The above graph shows the accuracy when trained for 10 epochs. As it can be seen, the final validation accuracy obtained is around 91% while in the previous experiment, an early stopping at 13 epochs was seen, which shows that between 10-12 epochs, optimal accuracy can be achieved.



***Figure C:Classification Report***

We can see the classification report above in which most of the classes have achieved accuracy above 90% while Covid 19 has a little less accuracy of 89% and Pneumothorax again has only 85% accuracy.

****

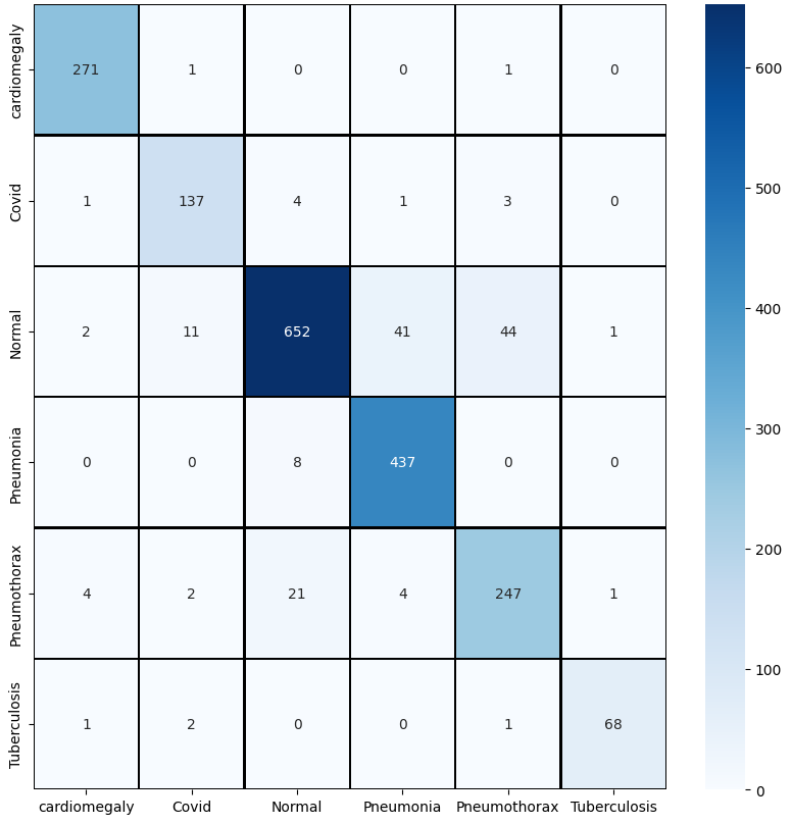
***Figure D: Confusion Matrix***

1. **Optimiser: Adam, Batch size: 64**

**20 epoch - early stop 13**

|  |  |
| --- | --- |
| ***Figure A: Train and validation accuracy*** | ***Figure B: Train loss and validation loss*** |

The above graph shows the accuracy when trained for 20 epochs. As it can be seen, the final validation accuracy obtained is around 92% which is the maximum accuracy obtained for the VGG19 model and an early stopping at 13 epochs was seen, which shows that at 13 epochs, optimal accuracy can be achieved.

****

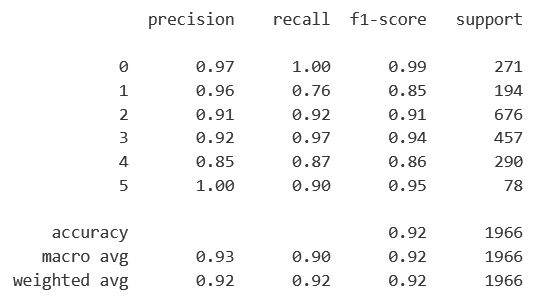
***Figure D: Confusion Matrix***

1. **Optimiser : RMSprop, Batch size : 32**

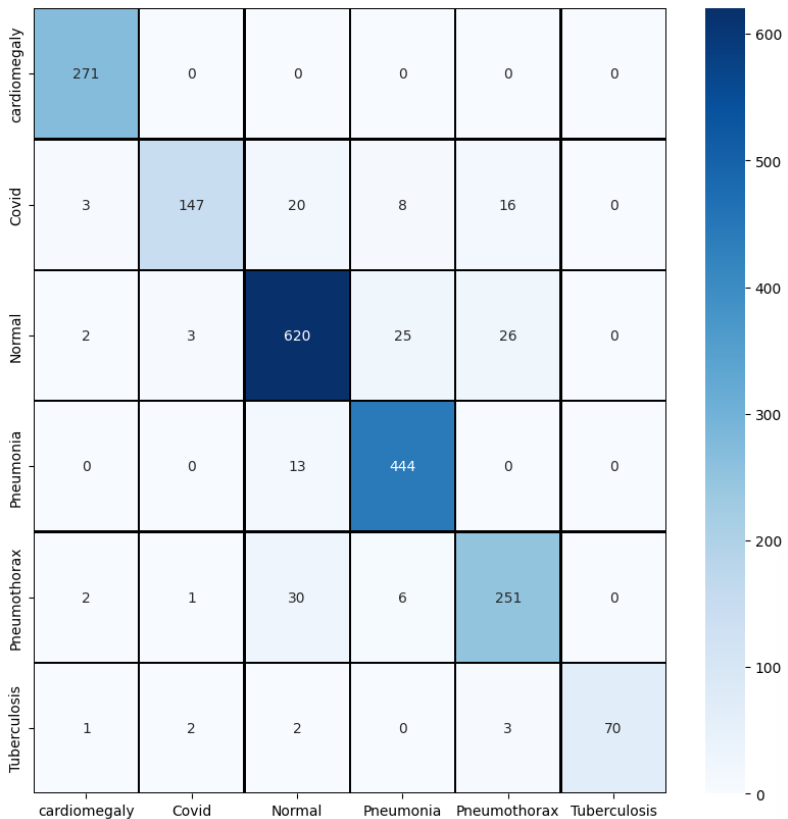
**10 epoch (7 early stop)**

|  |  |
| --- | --- |
| ***Figure A: Accuracy graph*** | ***Figure B: Validation graph*** |

The above graph shows the accuracy when trained for 10 epochs. As it can be seen, the final validation accuracy obtained is around 89% which is lower than Adam’s

****

**Classification report**

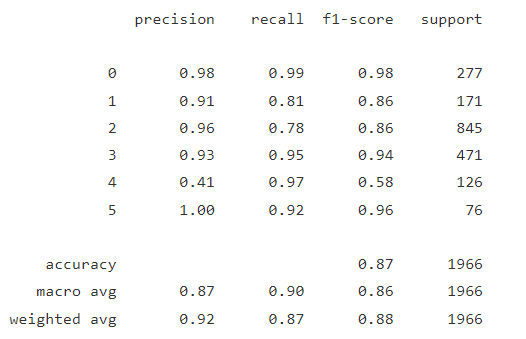
****

***Confusion Matrix***

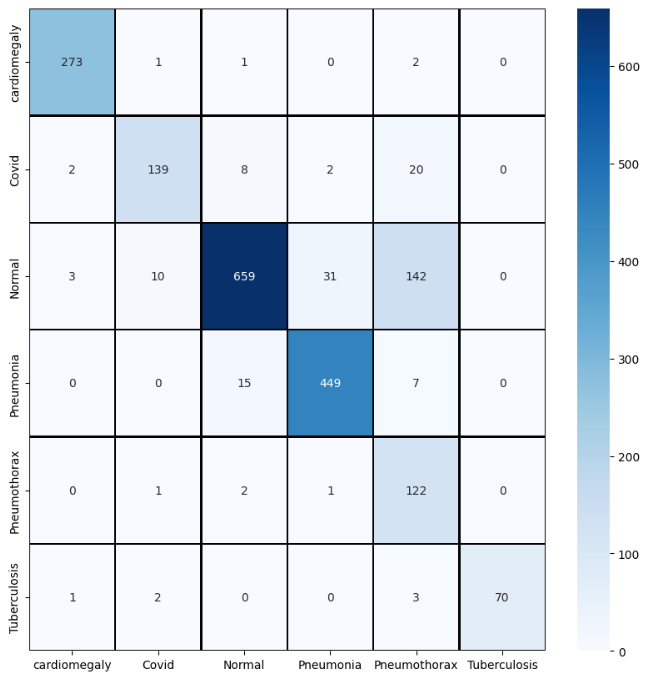
1. **Optimiser : RMSprop, Batch size : 64**

|  |  |
| --- | --- |
| ***Figure A: Train and validation accuracy*** | ***Figure B: Train loss and validation loss*** |

The above graph shows the accuracy when trained for 15 epochs. As it can be seen, the final validation accuracy obtained is around 87% accuracy obtained for the VGG19 model on this dataset and an early stopping at 13 epochs was seen, which shows that at 13 epochs, optimal accuracy can be achieved.

****

**Classification report**

****

***Figure D: Confusion Matrix***

Several models, including VGG19, Resnet50, and InceptionV3, were used for the experiments, and the results were evaluated based on training and test accuracy. In this comparative analysis, we will compare the performance of the different models and their limitations.

The Resnet50 model achieved the highest training accuracy of 97.24% among all the models, but its test accuracy was slightly lower at 93.03%. This model performed well in grayscale, indicating that grayscale normalization helped to reduce the impact of variations in illumination, making the model more robust.

The Inception V3 model achieved the highest test accuracy of 90.69%, but its training accuracy was higher than the test accuracy, indicating that it may have overfitted the data. This model required a larger input image size of 299x299, which may have contributed to the overfitting. The model was also trained with a smaller number of epochs (20) compared to other models.

The VGG19 model achieved moderate accuracy, with its highest accuracy achieved with a batch size of 64 and an epoch of 20, reaching 92.17%. However, the model was not able to achieve a high accuracy compared to other models.

The CNN model, which was built from scratch, achieved the lowest accuracy among all the models, with a test accuracy of 87.74%. This model required 30 epochs to achieve a training accuracy of 94.04%, indicating that it required more training time than the pre-trained models.

In general, the pre-trained models (VGG19, Resnet50, and InceptionV3) achieved better accuracy than the CNN model built from scratch, indicating that transfer learning is an effective approach for medical image classification. However, there were limitations to the models, including overfitting and the requirement for large input image sizes.

Future research could focus on exploring other pre-trained models, combining multiple models for improved accuracy, and using techniques like attention mechanisms to reduce overfitting and improve model performance. Additionally, more research could be done to evaluate the models on larger datasets and with more diverse patient populations to ensure the generalizability of the results.

**Conclusion:**

In conclusion, we have implemented deep CNNs and transfer learning techniques for multi-disease prediction using chest X-rays. We utilized three pre-trained models, VGG19, Resnet50 and InceptionV3, along with a CNN built from scratch to classify six categories - Cardiomegaly, Covid, Normal, Pneumonia, Pneumothorax, and Tuberculosis. The dataset was preprocessed to ensure that the images were scaled, normalized, and converted to grayscale to make them suitable for model implementation. The dataset was also split into training, validation, and testing sets for the purpose of training, hyperparameter tuning, and testing. Our model achieved high accuracy rates ranging from 87.74% to 93.34%, with Resnet50 grayscale achieving the highest accuracy.

While our results are promising, further research can be conducted to improve the accuracy and generalizability of the model. More comprehensive data augmentation techniques could be employed to increase the size and diversity of the dataset, and the model can be fine-tuned to increase its sensitivity and specificity for different disease classes. Additionally, the model's performance can be evaluated on larger datasets, and more advanced deep learning architectures can be explored for even better performance. Overall, our study demonstrates the potential of deep CNNs and transfer learning in accurately classifying chest X-rays and holds promise for clinical applications in the future.

**References:**

[1] Kumar M, Shakya D, Kurup V, Suksatan W. COVID-19 prediction through X-ray images using transfer learning-based hybrid deep learning approach. *Mater Today Proc. 2022;51:2520-2524. doi: 10.1016/j.matpr.2021.12.123. Epub 2021 Dec 13. PMID: 34926174; PMCID: PMC8666290.*

[2] Bens P., Tjeng W. C., Reza R., Arif B., Ettikan K. K.,Transfer Learning from Chest X-Ray Pre-trained Convolutional Neural Network for Learning Mammogram Data*,Procedia Computer Science,Volume 135,2018,Pages 400-407,ISSN 1877-0509.*

[3] Kamil D., Boran S.(2022). Skin Lesion Classification Using CNN-based Transfer Learning Model. *Journal Of Science*

[4] Chen, Y., Wan, Y. & Pan, F. Enhancing Multi-disease Diagnosis of Chest X-rays with Advanced Deep-learning Networks in Real-world Data. *J Digit Imaging* (2023). <https://doi.org/10.1007/s10278-023-00801-4>

[5] Sharma, C. M., Goyal, L., Chariar, V. M., & Sharma, N. (2022). Lung Disease Classification in CXR Images Using Hybrid Inception-ResNet-v2 Model and Edge Computing. *Journal of Healthcare Engineering*, *2022*.

[6] Karaddi, S. H., & Sharma, L. D. (2023). Automated multi-class classification of lung diseases from CXR-images using pre-trained convolutional neural networks. *Expert Systems with Applications*, *211*, 118650.

[7] Dina M. Ibrahim, Nada M. Elshennawy, Amany M. Sarhan, Deep-chest: Multi-classification deep learning model for diagnosing COVID-19, pneumonia, and lung cancer chest diseases, Computers in Biology and Medicine, Volume 132, 2021,104348, ISSN 0010-4825.

[8] Nahiduzzaman M, Islam MR, Hassan R. ChestX-Ray6: Prediction of multiple diseases including COVID-19 from chest X-ray images using convolutional neural network. Expert Syst Appl. 2023 Jan;211:118576. doi: 10.1016/j.eswa.2022.118576. Epub 2022 Aug 27. PMID: 36062267; PMCID: PMC9420006.

[9] A. Victor Ikechukwu, S. Murali, R. Deepu, R.C. Shivamurthy, ResNet-50 vs VGG-19 vs training from scratch: A comparative analysis of the segmentation and classification of Pneumonia from chest X-ray images, Global Transitions Proceedings, Volume 2, Issue 2, 2021, Pages 375-381, ISSN 2666-285X,

**Roles:**

|  |  |  |
| --- | --- | --- |
| **SNO** | **Team member** | **Works** |
| **1.** | **Shubhangi Agrawal (20BCE1161)** | Data Preprocessing,  VGG19 |
| **2.** | **Priyadharshini S.**  **(20BCE1016)** | Resnet50, CNN scratch |
| **3.** | **Shrishti Singh**  **(20BCE1164)** | InceptionV3 |

**Justification:**

Shubhangi Agrawal (20BCE1161) - Data Preprocessing, VGG19:

As the member responsible for data preprocessing, Shubhangi was tasked with preparing the data for training and analysis. This includes tasks such as cleaning the data, splitting the data intro train, test , validation and normalizing the data. Effective data preprocessing is critical to the success of any machine learning project as it can significantly impact the accuracy and efficiency of the models.

In addition to data preprocessing, Shubhangi was also be responsible for implementing the VGG19 model. VGG19 is a convolutional neural network (CNN) architecture that is widely used for image classification tasks. This model has 19 layers and is known for its excellent performance on image recognition tasks.

Priyadharshini S. (20BCE1016) - Resnet50, CNN scratch:

Priyadharshini was responsible for implementing the Resnet50 model, another widely used CNN architecture for image classification. Resnet50 is a deep neural network with 50 layers that is particularly good at detecting features in images.

In addition to Resnet50, Priyadharshini was also be responsible for implementing a CNN from scratch. This involves designing and implementing a CNN architecture from the ground up. By doing so, Priyadharshini could have the opportunity to customize the model to the specific needs of the project, potentially leading to improved accuracy and efficiency.

Shrishti Singh (20BCE1164) - InceptionV3:

As the member responsible for implementing InceptionV3, Shrishti was tasked with implementing another powerful CNN architecture for image classification. InceptionV3 is known for its excellent performance on a variety of image recognition tasks, particularly those involving complex images

Overall, each member of the team has a critical role to play in the project. By dividing responsibilities among the team members, the project can proceed more efficiently and effectively, potentially leading to better results. With Shubhangi handling data preprocessing and implementing VGG19, Priyadharshini implementing Resnet50 and a CNN from scratch, and Shrishti implementing InceptionV3, the team is well-positioned to do a comparative analysis for the multiple disease prediction using chest xrays