Mini Project Report

on

Dimensionality Reduction for Improved Chest Radiograph Classification

By

Bishant Raaj Bhujel (Reg. No.- 202000224)

Mayal Punu Lepcha (Reg. No.- 202000283)

Rinchen Tempa Bhutia (Reg. No.- 202000117)

Group Id – C15

In partial fulfillment of requirements for the award of degree in

Bachelor of Technology in Computer Science and Engineering

(2023)

Under the Project Guidance of

Mr. Ashis Pradhan

Assistant Professor (SG)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY

(A constituent college of Sikkim Manipal University)

MAJITAR, RANGPO, EAST SIKKIM – 737136

PROJECT COMPLETION CERTIFICATE

This is to certify that the below mentioned students of Sikkim Manipal Institute of Technology have worked under my supervision and guidance from **9th January 2023 to 29th April 2023** and successfully completed the Mini project entitled **"Dimensionality Reduction for Improved Chest Radiograph Classification"** in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Course
202000224	**Bishant Raaj Bhujel**	**B.Tech (CSE)**
202000283	**Mayal Punu Lepcha**	**B.Tech (CSE)**
202000117	**Rinchen Tempa Bhutia**	**B.Tech (CSE)**

Mr. Ashis Pradhan

Assistant Professor (SG)

Department of Computer Science and Engineering

Sikkim Manipal institute of Technology

Majhitar, Sikkim – 737136

PROJECT REVIEW CERTIFICATE

This is to certify that the work recorded in this project report entitled **“Dimensionality Reduction for Improved Chest Radiograph Classification”** has been jointly carried out by **Bishant Raaj Bhujel (Reg. 202000224), Mayal Punu Lepcha (Reg. 202000283) and Rinchen Tempa Bhutia (Reg. 202000117)** of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering. This report has been duly reviewed by the undersigned and recommended for final submission for Mini Project Viva Examination.

Mr. Ashis Pradhan

Assistant Professor (SG)

Department of Computer Science and Engineering

Sikkim Manipal institute of Technology

Majhitar, Sikkim – 737136

CERTIFICATE OF ACCEPTANCE

This is to certify that the below mentioned students of Computer Science & Engineering Department of Sikkim Manipal Institute of Technology (SMIT) have worked under the supervision of **Ashis Pradhan**, Assistant Professor, Department of Computer Science and Engineering from **9th January 2023 to 29th April 2023** on the project entitled **"Dimensionality Reduction for Improved Chest Radiograph Classification".**

The project is hereby accepted by the Department of Computer Science & Engineering, SMIT in partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science and Engineering.

University Registration No	Name of Student	Project Venue
202000224	**Bishant Raaj Bhujel**	**SMIT**
202000283	**Mayal Punu Lepcha**	
202000117	**Rinchen Tempa Bhutia**	

Dr. Udit Kumar Chakraborty

Professor & Head of the Department

Computer Science & Engineering Department

Sikkim Manipal Institute of Technology

Majhitar, Sikkim – 737136

DECLARATION

We, the undersigned, hereby declare that the work recorded in this project report entitled "**Dimensionality Reduction for Improved Chest Radiograph Classification**" in partial fulfillment for the requirements of award of B.Tech (CSE) from Sikkim Manipal Institute of Technology (A constituent college of Sikkim Manipal University) is a faithful and bonafide project work carried out at "**SIKKIM MANIPAL INSTITUTE OF TECHNOLOGY**" under the supervision and guidance of **Mr. Ashis Pradhan,** Assistant Professor, Department of Computer Science and Engineering.

The results of this investigation reported in this project have so far not been reported for any other Degree or any other technical forum.

The assistance and help received during the investigation have been duly acknowledged.

Bishant

Bishant Raaj Bhujel (Reg. No.-202000224)

Mayal Punu Lepcha

Mayal Punu Lepcha (Reg. No.-202000283)

Rinchen Tempa Bhutia (Reg. No.-202000117)

ACKNOWLEDGMENT

We take this opportunity to acknowledge indebtedness and a deep sense of gratitude to our guide **Mr. Ashis Pradhan** for his/her valuable guidance and supervision throughout the course which shaped the present work as it shows.

We pay our deep sense of gratitude to **Prof. (Dr.) Udit Kumar Chakraborty, HOD, Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us the opportunity to work on this project and providing all support required.

We are obliged to our project coordinators **Dr. Sandeep Gurung** and **Mr. Biraj Upadhyaya** for elevating, inspiration and supervising in completion of our project.

We would also like to thank any other staff of **Computer Science & Engineering Department, Sikkim Manipal Institute of Technology** for giving us continuous support and guidance that has helped us in completion of our project.

Bishant Raaj Bhujel (Reg. No.-202000224)

Mayal Punu Lepcha (Reg. No.-202000283)

Rinchen Tempa Bhutia (Reg. No.-202000117)

DOCUMENT CONTROL SHEET

1	Report No	CSE/Mini Project/Internal/B.Tech/C15/2023
2	Title of the Report	Dimensionality Reduction for Improved Chest Radiograph Classification
3	Type of Report	Technical
4	Author	Bishant Raaj Bhujel, Mayal Punu Lepcha, Rinchen Tempa Bhutia
5	Organizing Unit	Sikkim Manipal Institute of Technology
6	Language of the Document	English
7	Abstract	A CNN Model that uses dimensionality reduction as preprocessing to effectively classify Chest X-Ray Images.
8	Security Classification	General
9	Distribution Statement	General

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The Covid-19 pandemic has exposed the limitations of our healthcare systems and the pressure such a situation puts on our medical workforce. The high volume of patients, the shortage of healthcare workers, and limited resources have highlighted the need for efficient and effective diagnostic tools to improve the overall efficiency and effectiveness of the diagnostic process.

One such diagnostic tool is a Chest Radiograph Classifier. Chest Radiographs are a valuable diagnostic tool as they are fast to obtain, easily accessible, and can be used to detect the presence of a variety of lung diseases. Although a variety of image classification models are available, they still suffer from a major problem: the problem of high dimensionality of data.

The high dimensionality of the Chest Radiograph images can make it difficult for traditional machine learning algorithms to classify them effectively. To tackle this problem, we aim to use, test, and compare various dimensionality reduction algorithms to reduce the dimensionality of the images and extract the most relevant features for effective disease classification for our model.

The goal of this project is to build a chest radiograph classifier that not only accurately classifies diseases based on an input x-ray image, but also addresses the problem of high dimensionality. By reducing the dimensionality of the images, the proposed classifier aims to not only improve the accuracy of disease classification but also improve the computational efficiency of the model making it more practical for real-world applications.

1. INTRODUCTION

Chest radiographs, also known as chest X-rays, are a valuable diagnostic tool in the medical field. They are widely used to detect the presence of a variety of lung diseases, such as pneumonia, tuberculosis, and lung cancer. Chest radiographs are fast to obtain, easily accessible, and relatively inexpensive, making them an ideal tool for screening and diagnosis. They can provide a wealth of information about a patient's lung function and overall health, including the size, shape, and position of the lungs and the heart, as well as the presence of fluid, masses, and other abnormalities.

Chest Radiograph Classifier (Image Classification Model) can be an effective diagnostic tool that can be used in detecting and classifying lung infections such as COVID-19, Pneumonia and Tuberculosis and hence can assist doctors in interpreting chest X-Rays. It can help to reduce the workload of radiologists by assisting in the process of medical image analysis which can improve the efficiency of the diagnostic process.

However, such Classifiers still have certain limitations that can affect their performance and accuracy. In our project, we plan to address the limitation of high dimensionality in chest x-ray images by using dimensionality reduction techniques. Chest X-rays are high-dimensional medical images, meaning that they contain many pixels and features. This high dimensionality can create several problems for learning algorithms when it comes to analysing and classifying these images. In addition, High-dimensional images require more computational resources and time to process, which can make it difficult to develop and implement real-time diagnostic tools.

2. LITERATURE SURVEY

SL No.	Author(s)	Paper and Publication Details	Findings	Relevance to the Project
1.	*Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir.*	"Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images." Computers in Biology and Medicine 132, Elsevier. [4th March 2021]	Study confirmed that deep learning models can be used to accurately detect COVID-19 from chest X-ray (CXR) images.	Dataset being used was produced by this research.
2.	*Thet Thet Khaing, Phyu Sin Nyein, Myint Soe Khyaing, Khaing Khaing Wai*	"Dimension Reduction of Images Using Principal Component Analysis Algorithm" Iconic Research and Engineering Journals [May 2020]	Study found that PCA effectively reduced the file size of the images and improved the transmission time for the compressed images.	Provides algorithm for applying PCA to images.
3.	*Ayesha, Shaeela; Hanif, Muhammad Kashif;*	"Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data" Information Fusion [Jan 2020]	Study found that Linear techniques are less computationally intensive but nonlinear techniques can be useful for complex data.	Discusses and compares various dimensionality reduction techniques.
4.	*Taufit Rahmat, Azlan Ismail, Sharifah Aliman*	"Chest X-Rays Image Classification in Medical Image Analysis" Universiti Teknologi, Malaysia. [27th December 2018]	Study discusses approaches for classifying chest X-ray images, including the classification problem types, datasets used, splitting ratios, etc.	Discusses various approaches for CXR image classification.

Table 2.1: Literature Survey

3. PROBLEM DEFINITION

There is a need for an accurate and efficient diagnostic tool to assist medical professionals and improve the diagnostic process and hence reduce the burden on healthcare workers. Chest radiographs are a widely used and accessible tool for detecting a variety of lung conditions, and their importance has been highlighted during the Covid-19 pandemic with the high volume of patients, shortage of healthcare workers, and limited resources. Developing an effective diagnostic tool such as a Chest X-Ray Classifier is crucial for addressing the pressing needs of our healthcare system.

The problem of high dimensionality in chest radiograph images is a significant challenge in the classification of lung diseases and image classification models in general. High dimensional data can make it difficult for traditional machine learning algorithms to process and classify the images effectively, resulting in lower accuracy and higher computational costs. Hence there is a need for an efficient computational system that can in turn be cost effective without compromising its performance or accuracy.

4. SOLUTION STRATEGY

We will divide our project in mainly two parts:

1) Using Dimensionality Reduction Algorithms on CXR Image Dataset.

   Algorithms: PCA (Principal Component Analysis)

2) Building Chest Radiograph Classifier on the dimensionality-reduced dataset.

   Algorithms: CNN (Convolutional Neural Networks)

In addition, on successful completion of the above two tasks, we will also deploy a web application created using ReactJS where the said model will be integrated through Flask (Python). Users will directly be able to upload CXR Images and get instant results from our model.

General Methodology:

STEP 1: Download, Pre-process the dataset which involves tasks like resizing images, splitting data into train, validation, and test sets.

STEP 2: Use dimensionality reduction algorithms like Principal Component Analysis (PCA) to reduce the dimensionality of the image data.

STEP 3: Train a machine learning model on the reduced-dimensionality image data. This model will be used to predict the labels of new chest X-ray images.

STEP 4: Evaluate the performance of the trained model on the test set and make any necessary adjustments to the model or the pre-processing steps.

The solution strategy for this project can be divided into several steps:

1. Data Collection: The first step is to collect a large dataset of Chest Radiographs that includes various lung diseases, such as pneumonia, and COVID-19, along with normal chest X-rays.

2. Pre-processing: To get the data ready for training, the next step is to pre-process it, which involves resizing, normalising, and enhancing the photos.

3. Dimensionality Reduction: In this step, various dimensionality reduction techniques, including Principal Component Analysis (PCA), will be put to the test and compared in order to determine which method best decreases the dimensionality of the images while retaining the most crucial features for disease classification.

4. Model Development: After reducing the dimensionality of the data, a machine learning model, such as a Convolutional Neural Network (CNN), will be trained using the reduced features to classify Chest Radiographs into normal and abnormal classes, and further categorize them into specific lung diseases.

5. Model Evaluation: A test dataset will be used to gauge how well the trained model categorises diseases. Accuracy, precision, recall, and F1-score are some of the assessment metrics that will be used to evaluate the model's performance.

6. Fine-tuning: The model will be improved by making the appropriate adjustments in response to the evaluation. This phase might entail optimising the dimensionality reduction process further, choosing better classification algorithms, and fine-tuning the hyperparameters.

7. Deployment: Finally, the trained model will be deployed for real-world applications, where it can be used to classify Chest Radiographs into normal and abnormal classes and specific lung diseases, providing an effective diagnostic tool for medical professionals.

5. DESIGN

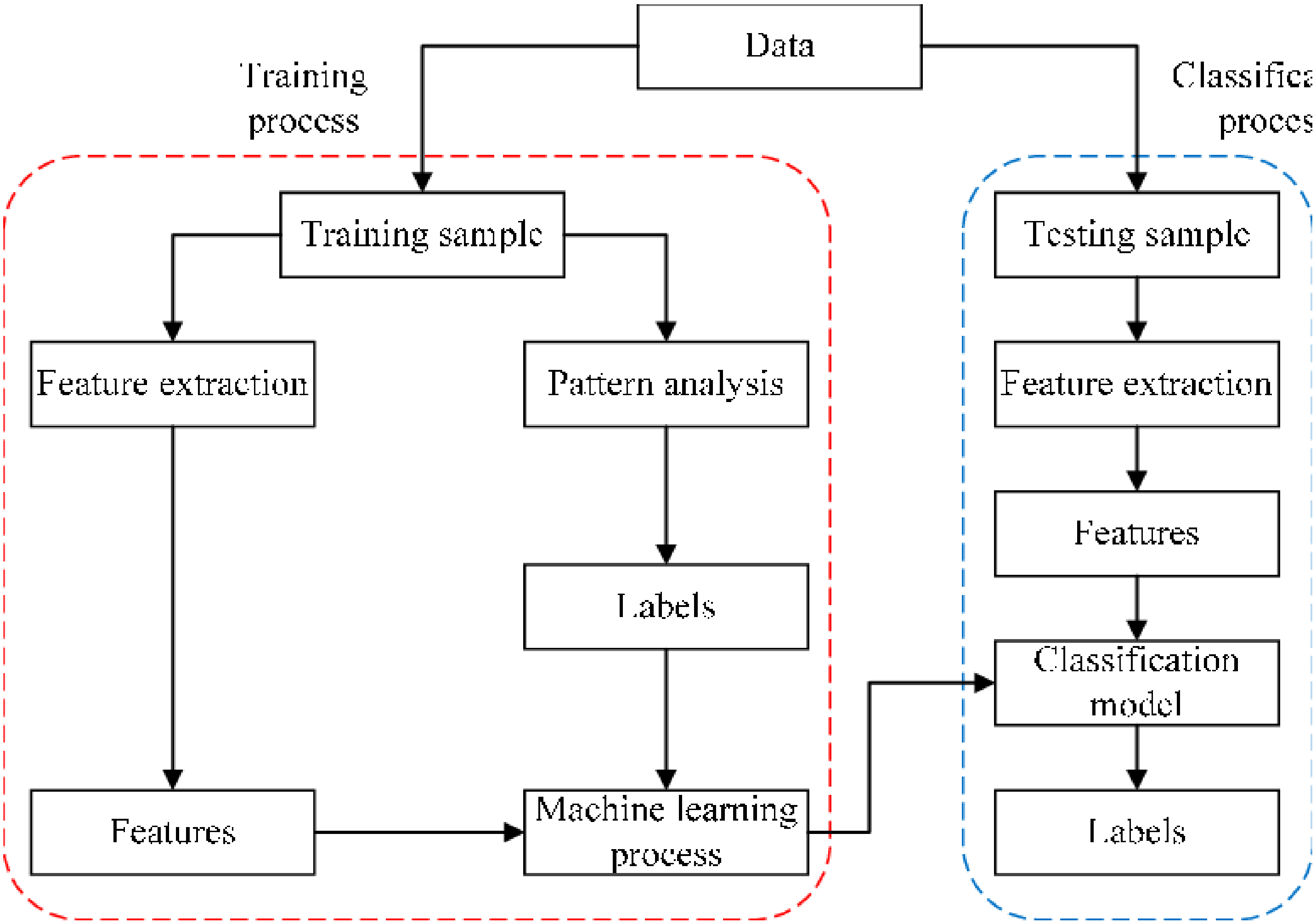


Figure 5.1: Flowchart of Project

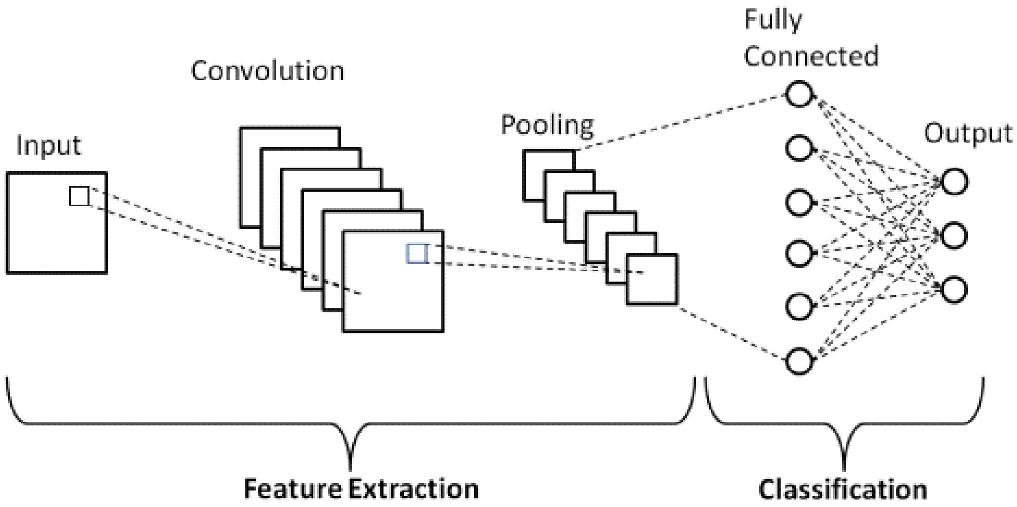


Fig. 5.2: A representation of the CNN in use

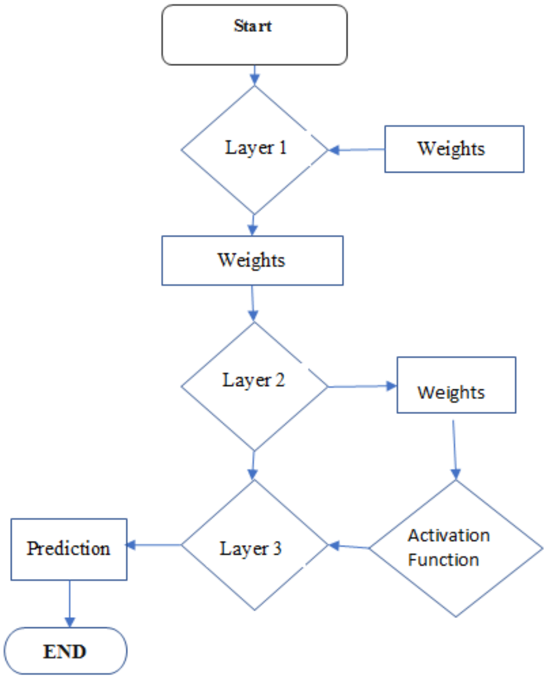


Figure 5.3: Flow chart of CNN

6. IMPLEMENTATION DETAILS

- **6.1: Algorithms**

Principal Component Analysis

STEP 1: First, we must standardise the data so that each variable has a mean of 0 and a standard deviation of 1, in order to perform PCA. To do this, divide the result by the standard deviation after subtracting the mean of each variable from its corresponding values.

STEP 2: **Compute the covariance matrix**:

The covariance matrix of the standardised data must then be computed. The covariance matrix calculates the correlation between every variable in the data set and every other variable.

STEP 3: Calculate the covariance matrix's eigenvalues and eigenvectors.:

The directions and magnitudes of the primary components of the data are represented by the eigenvectors and eigenvalues of the covariance matrix. Numerous techniques, including the power iteration method and the QR algorithm, can be used to calculate them.

STEP 4: Decide on the number of main components.:

We need to choose the number of principal components to retain for our analysis. This can be based on the eigenvalues of the covariance matrix, where the principal components with the largest eigenvalues represent the most important features of the data.

STEP 5: Project the data onto the principal components:

To generate a lower-dimensional representation of the data, we finally project the standardised data onto the selected principle components. The standardised data can be multiplied by the eigenvectors associated with the selected major components to achieve this.make it shorter.

Convolutional Neural Networks (CNNs)

STEP 1: Define the CNN model's architecture, including the quantity of filters, the size of the kernels, the activation functions, and the output categories.

Step 2: is to create a Sequential class instance and use the add function to incorporate the layers into the model.

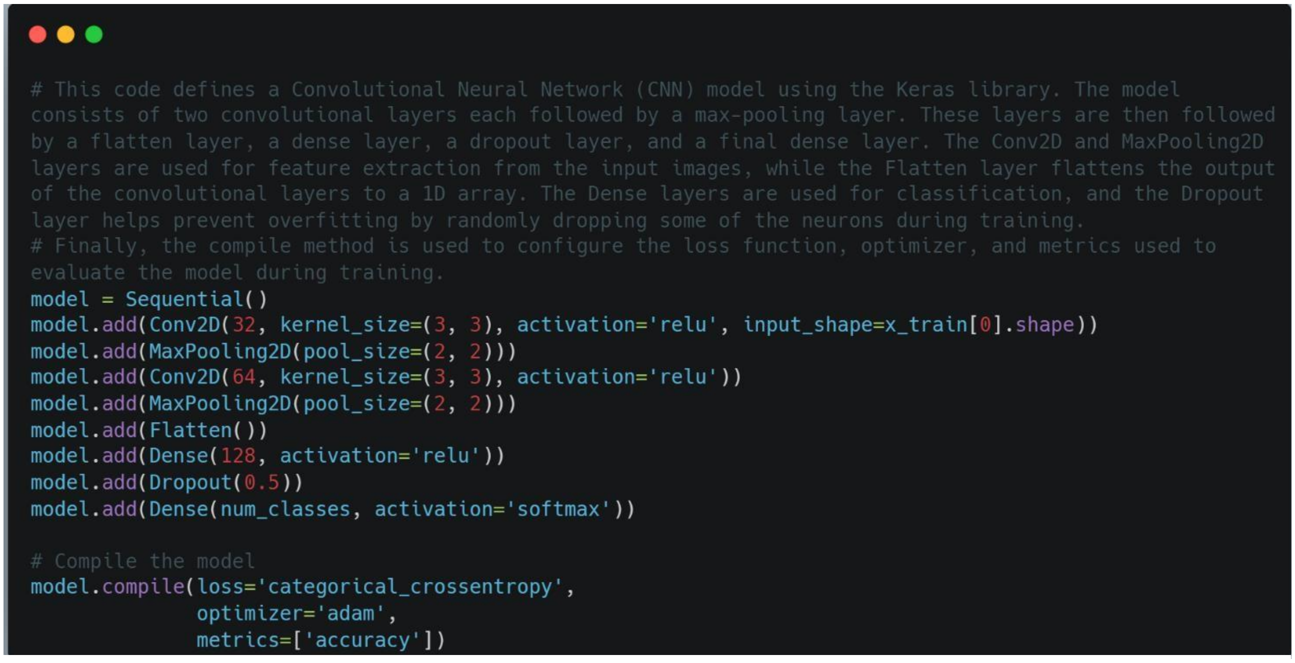
STEP 3: Compile the model using the compile method, specifying the metrics required to gauge the model's performance during training as well as the optimizer and loss function.

STEP 4:, the model is trained using the fit method, with the batch size, number of epochs, and validation data all being specified.

STEP 5: A dropout layer should be added to the model in step 5 to help prevent overfitting during training.

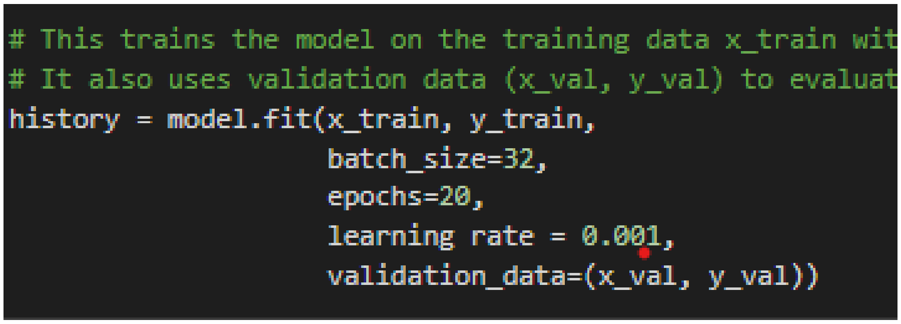
STEP 6: Use the evaluate method, which returns the loss and accuracy metrics, to assess the model's performance on the test data.

- **6.2: Screenshots**



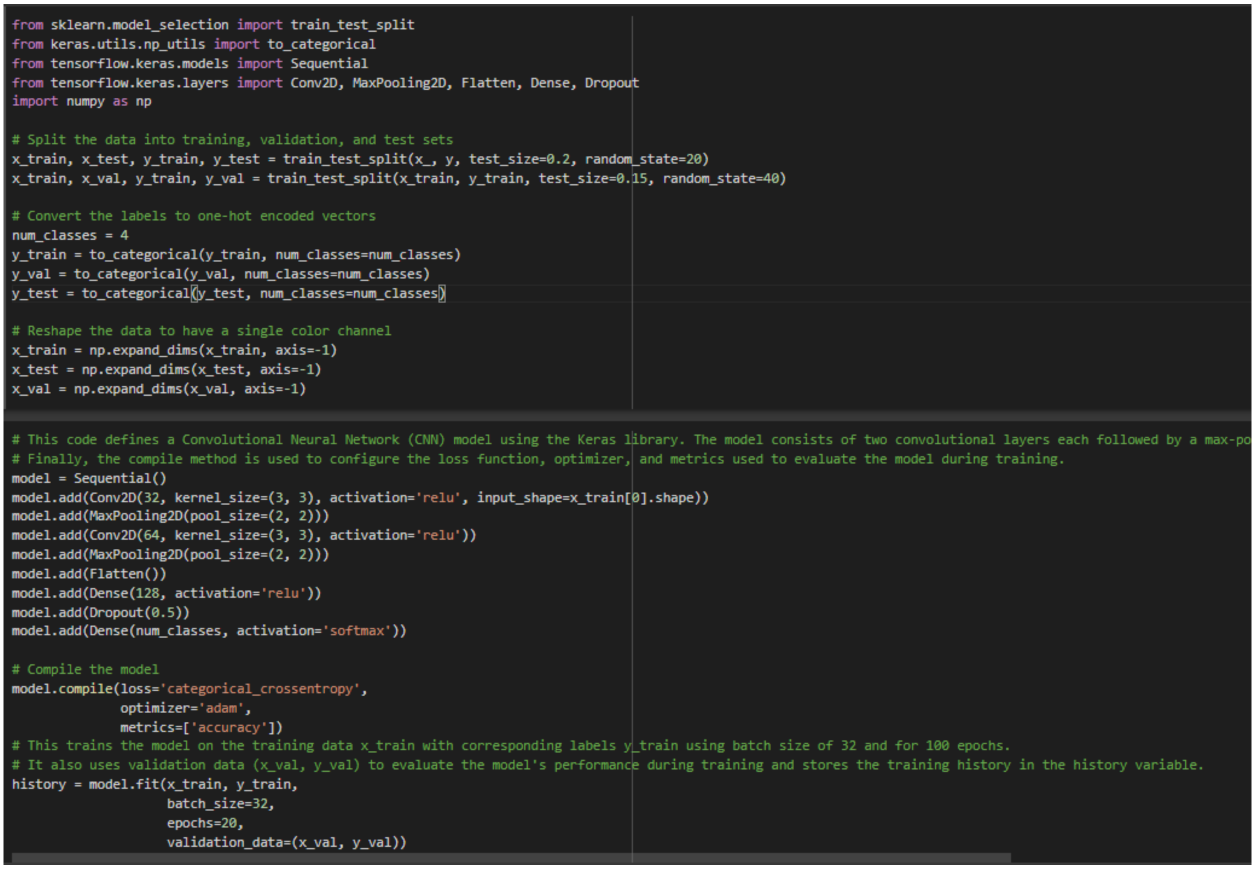
```
# This code defines a Convolutional Neural Network (CNN) model using the Keras library. The model
consists of two convolutional layers each followed by a max-pooling layer. These layers are then followed
by a flatten layer, a dense layer, a dropout layer, and a final dense layer. The Conv2D and MaxPooling2D
layers are used for feature extraction from the input images, while the Flatten layer flattens the output
of the convolutional layers to a 1D array. The Dense layers are used for classification, and the Dropout
layer helps prevent overfitting by randomly dropping some of the neurons during training.
# Finally, the compile method is used to configure the loss function, optimizer, and metrics used to
evaluate the model during training.
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=x_train[0].shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

Fig 6.1: Code for Convolutional Neural Network Architecture



```
# This trains the model on the training data x_train wit
# It also uses validation data (x_val, y_val) to evaluat
history = model.fit(x_train, y_train,
                    batch_size=32,
                    epochs=20,
                    learning rate = 0.001,
                    validation_data=(x_val, y_val))
```

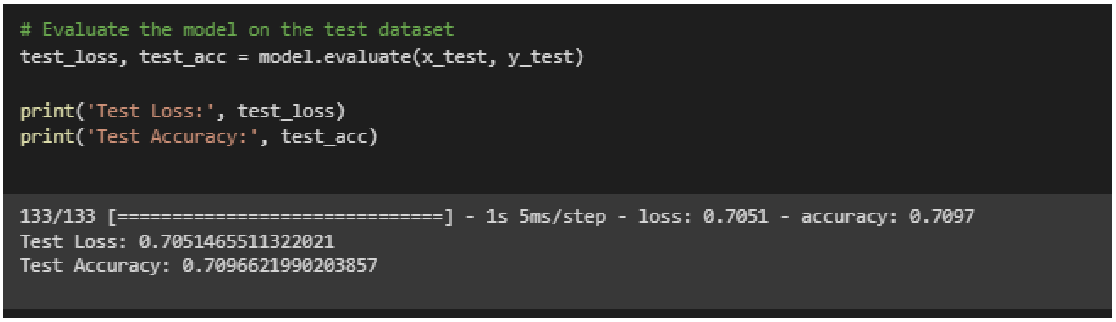
Figure 6.2: Hyperparameter Variables



```
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
import numpy as np

# Split the data into training, validation, and test sets
x_train, x_test, y_train, y_test = train_test_split(x_, y, test_size=0.2, random_state=20)
x_train, x_val, y_train, y_val = train_test_split(x_train, y_train, test_size=0.15, random_state=40)

# Convert the labels to one-hot encoded vectors
num_classes = 4
y_train = to_categorical(y_train, num_classes=num_classes)
y_val = to_categorical(y_val, num_classes=num_classes)
y_test = to_categorical(y_test, num_classes=num_classes)

# Reshape the data to have a single color channel
x_train = np.expand_dims(x_train, axis=-1)
x_test = np.expand_dims(x_test, axis=-1)
x_val = np.expand_dims(x_val, axis=-1)
```

```
# This code defines a Convolutional Neural Network (CNN) model using the Keras library. The model consists of two convolutional layers each followed by a max-po
# Finally, the compile method is used to configure the loss function, optimizer, and metrics used to evaluate the model during training.
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=x_train[0].shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))

# Compile the model
model.compile(loss='categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
# This trains the model on the training data x_train with corresponding labels y_train using batch size of 32 and for 100 epochs.
# It also uses validation data (x_val, y_val) to evaluate the model's performance during training and stores the training history in the history variable.
history = model.fit(x_train, y_train,
                    batch_size=32,
                    epochs=20,
                    validation_data=(x_val, y_val))
```

Figure 6.3: Code for training the CNN Model



```
# Evaluate the model on the test dataset
test_loss, test_acc = model.evaluate(x_test, y_test)

print('Test Loss:', test_loss)
print('Test Accuracy:', test_acc)
```

```
133/133 [==============================] - 1s 5ms/step - loss: 0.7051 - accuracy: 0.7097
Test Loss: 0.7051465511322021
Test Accuracy: 0.7096621990203857
```

Figure 6.4: Code for Testing the CNN Model

- **6.3: Dataset**

The dataset for our model is titled "COVID-19 Radiography Database" and it contains around 20,000 annotated CXR images.

CLASSES = {Covid-19, Normal, Lung – Opacity, Viral Pneumonia}

The COVID-19 Radiography Database is a collection of normal and viral pneumonia images as well as chest X-rays for COVID-19 positive cases. A group of researchers from Qatar University in Doha, Qatar, and the University of Dhaka in Bangladesh, along with medical professionals from Pakistan and Malaysia, produced the database. The Kaggle community chose this dataset as the COVID-19 Dataset Award winner.

The dataset is made available in phases. The database had 219 COVID-19, 1341 regular, and 1345 viral pneumonia chest X-ray images in the initial release. The COVID-19 class was expanded to 1200 CXR pictures in the initial release. The database was subsequently increased in the second update to include 3616 COVID-19 positive cases, 10,192 normal, 6012 lung opacity (non-COVID lung infection), and 1345 viral pneumonia photos, as well as associated lung masks.

Each image in the collection is 299x299 pixels in size and is in the JPEG format. A metadata file that includes details about the patient's age, gender, and location is attached to the photos. Each image in the dataset also has lung masks that can be used to isolate the lung region for additional research.

For the development and testing of machine learning models for the diagnosis of COVID-19 and other lung infections, this dataset is an invaluable resource for researchers and medical practitioners. The dataset can be used to train models that can precisely identify COVID-19 in chest X-rays because it contains a large number of COVID-19 cases. Models that can distinguish between COVID-19 and other lung infections can be trained by including cases of normal lung function as well as cases of other lung infections. Overall, the COVID-19 pandemic is being fought thanks in large part to this dataset.

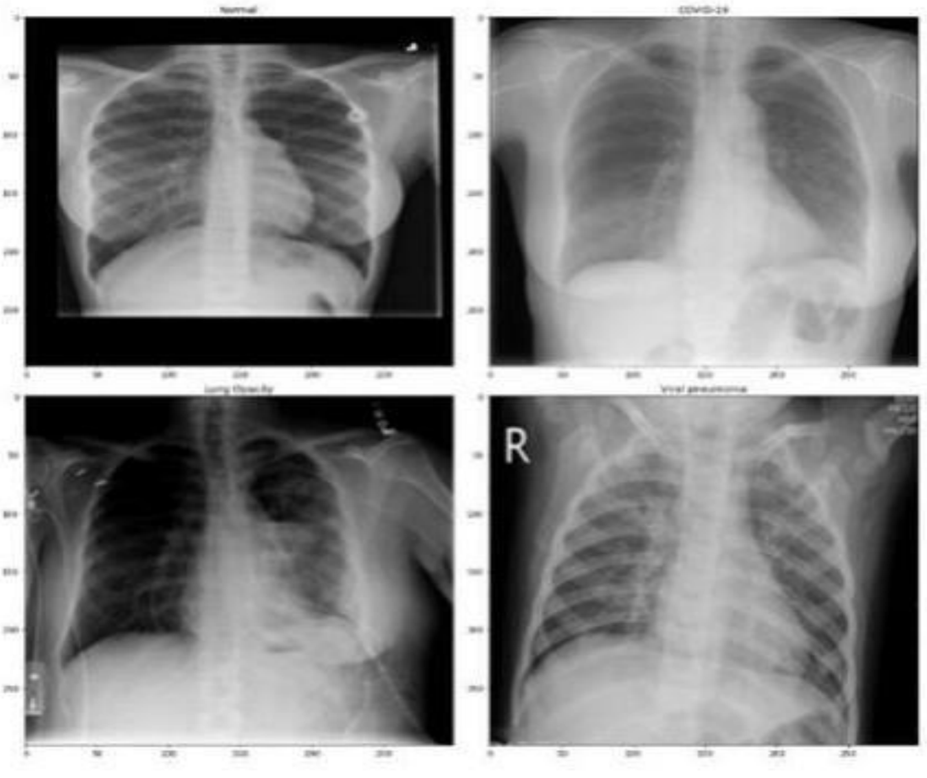
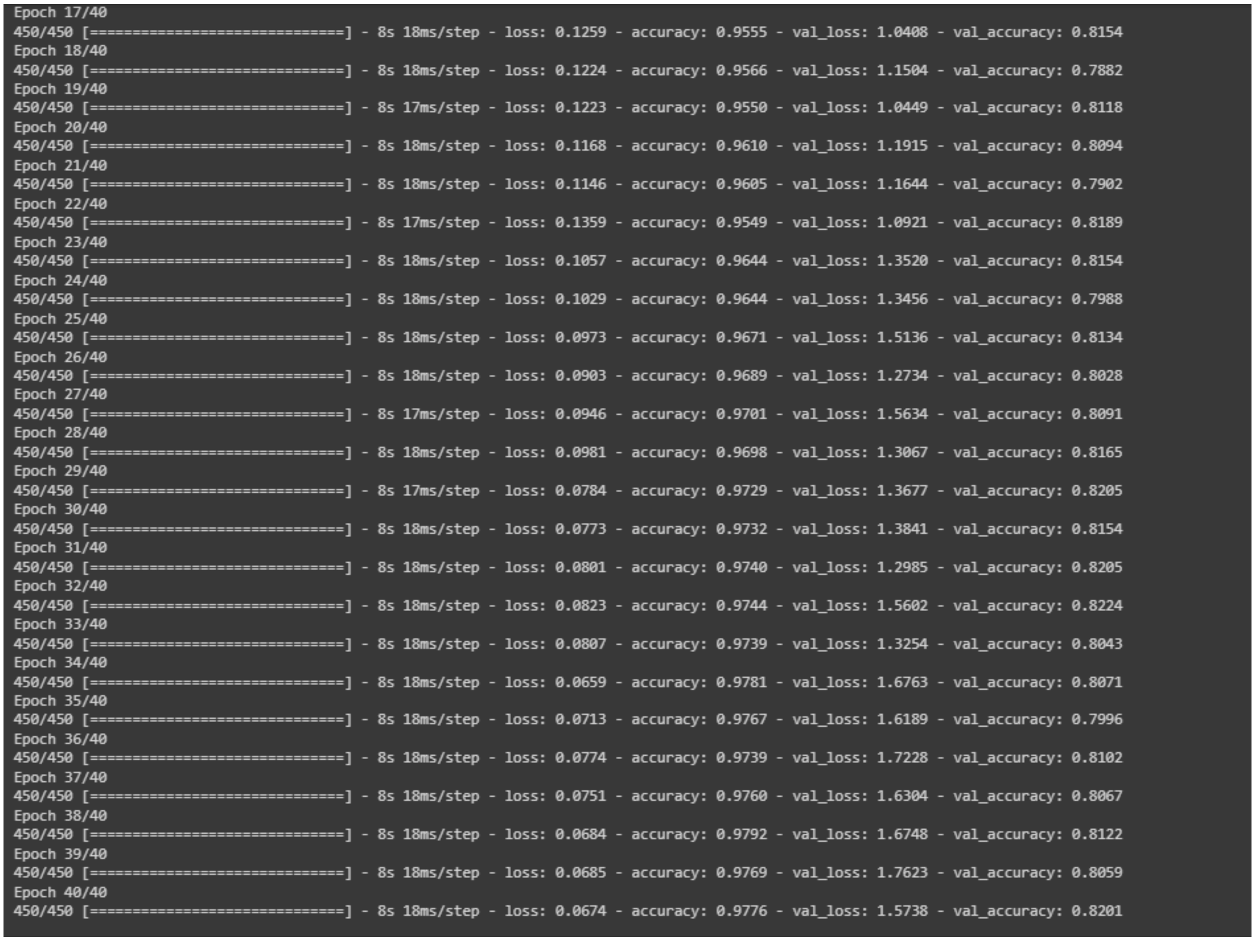


Figure 6.5: Covid 19 Radiography Dataset (Samples from 4 classes)

7. RESULTS AND DISCUSSION

7.1 Accuracy: The Convolutional Neural Network model achieved an accuracy of 97% on training set which is relatively high.



```
Epoch 17/40
450/450 [==============================] - 8s 18ms/step - loss: 0.1259 - accuracy: 0.9555 - val_loss: 1.0408 - val_accuracy: 0.8154
Epoch 18/40
450/450 [==============================] - 8s 18ms/step - loss: 0.1224 - accuracy: 0.9566 - val_loss: 1.1504 - val_accuracy: 0.7882
Epoch 19/40
450/450 [==============================] - 8s 17ms/step - loss: 0.1223 - accuracy: 0.9550 - val_loss: 1.0449 - val_accuracy: 0.8118
Epoch 20/40
450/450 [==============================] - 8s 18ms/step - loss: 0.1168 - accuracy: 0.9610 - val_loss: 1.1915 - val_accuracy: 0.8094
Epoch 21/40
450/450 [==============================] - 8s 18ms/step - loss: 0.1146 - accuracy: 0.9605 - val_loss: 1.1644 - val_accuracy: 0.7902
Epoch 22/40
450/450 [==============================] - 8s 17ms/step - loss: 0.1359 - accuracy: 0.9549 - val_loss: 1.0921 - val_accuracy: 0.8189
Epoch 23/40
450/450 [==============================] - 8s 18ms/step - loss: 0.1057 - accuracy: 0.9644 - val_loss: 1.3520 - val_accuracy: 0.8154
Epoch 24/40
450/450 [==============================] - 8s 18ms/step - loss: 0.1029 - accuracy: 0.9644 - val_loss: 1.3456 - val_accuracy: 0.7988
Epoch 25/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0973 - accuracy: 0.9671 - val_loss: 1.5136 - val_accuracy: 0.8134
Epoch 26/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0903 - accuracy: 0.9689 - val_loss: 1.2734 - val_accuracy: 0.8028
Epoch 27/40
450/450 [==============================] - 8s 17ms/step - loss: 0.0946 - accuracy: 0.9701 - val_loss: 1.5634 - val_accuracy: 0.8091
Epoch 28/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0981 - accuracy: 0.9698 - val_loss: 1.3067 - val_accuracy: 0.8165
Epoch 29/40
450/450 [==============================] - 8s 17ms/step - loss: 0.0784 - accuracy: 0.9729 - val_loss: 1.3677 - val_accuracy: 0.8205
Epoch 30/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0773 - accuracy: 0.9732 - val_loss: 1.3841 - val_accuracy: 0.8154
Epoch 31/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0801 - accuracy: 0.9740 - val_loss: 1.2985 - val_accuracy: 0.8205
Epoch 32/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0823 - accuracy: 0.9744 - val_loss: 1.5602 - val_accuracy: 0.8224
Epoch 33/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0807 - accuracy: 0.9739 - val_loss: 1.3254 - val_accuracy: 0.8043
Epoch 34/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0659 - accuracy: 0.9781 - val_loss: 1.6763 - val_accuracy: 0.8071
Epoch 35/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0713 - accuracy: 0.9767 - val_loss: 1.6189 - val_accuracy: 0.7996
Epoch 36/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0774 - accuracy: 0.9739 - val_loss: 1.7228 - val_accuracy: 0.8102
Epoch 37/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0751 - accuracy: 0.9760 - val_loss: 1.6304 - val_accuracy: 0.8067
Epoch 38/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0684 - accuracy: 0.9792 - val_loss: 1.6748 - val_accuracy: 0.8122
Epoch 39/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0685 - accuracy: 0.9769 - val_loss: 1.7623 - val_accuracy: 0.8059
Epoch 40/40
450/450 [==============================] - 8s 18ms/step - loss: 0.0674 - accuracy: 0.9776 - val_loss: 1.5738 - val_accuracy: 0.8201
```

Fig. 7.0: Accuracy of CNN Model

7.2. Confusion Matrix

A confusion matrix is a table that is often used to evaluate the performance of classification model. It is a matrix with rows and columns that represent the predicted and actual classifications, respectively. The four quadrants of the matrix represent the numbers of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN) produced by the classification model. The confusion matrix allows us to visualize the performance of the classification model and calculate various metrics such as accuracy, precision, recall, and F1-score. It is a useful tool for identifying the types of errors a model is making and can help to improve the model's performance.

Following is the Confusion Matrix obtained for our Model:

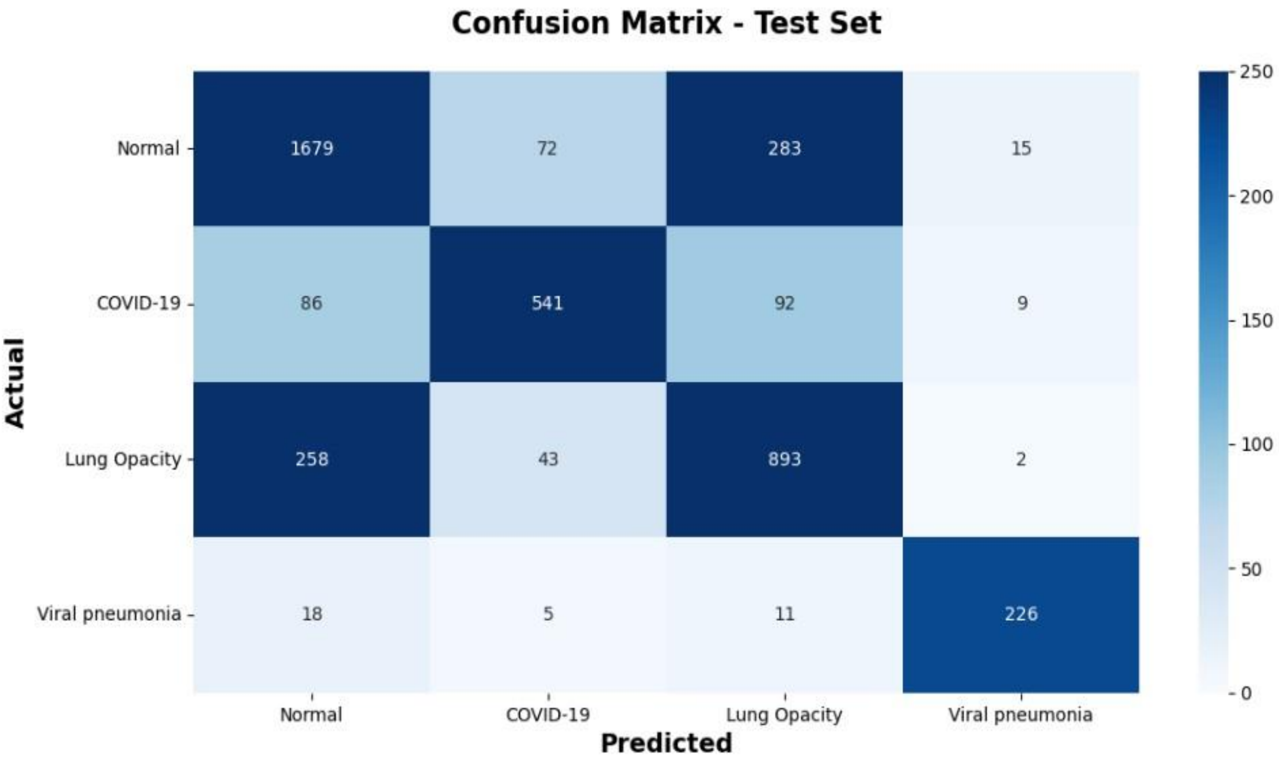
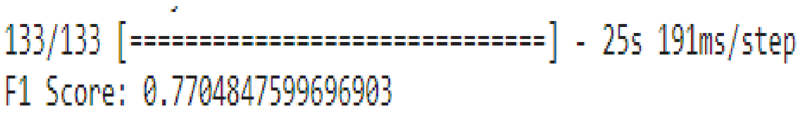


Figure 7.1: Confusion Matrix for our CNN Model

In this step, hyperparameters may need to be adjusted, better classification algorithms may need to be chosen, and the dimensionality reduction algorithm may need to be further optimised.

7.3 F1 SCORE

The precision and recall of a model's predictions are taken into account while calculating its F1 score, which is a gauge of its accuracy. Precision and recall are balanced by the harmonic methods of precision and recall. Given that it gives equal weight to both the positive and negative classes, the F1 score is a more useful indicator than accuracy in situations where there is an unbalanced class distribution. A model with a high F1 score has high precision and recall, which means that it can correctly identify the positive class without producing false positives or false negatives.



```
133/133 [==============================] - 25s 191ms/step
F1 Score: 0.7704847599696903
```

Figure 7.2: F1 Score of CNN Model on 5 epochs.

Our CNN Model got a F1 Score of 77% on 5 epochs of training and around 84% on 40 epochs of training.

BATCH SIZE VS ACCURACY

Batch Size	Accuracy
4	79.2
8	80.1
16	82.8
32	84.5
64	85.2
128	85.9
256	86.3

EPOCH VS ACCURACY

Epoch	Accuracy
5	76.2
10	77.3
15	77.9
20	78.4
25	79.7
30	80.4
35	81.2
40	83.6

7.4 Results observed on specific configuration.

- No of epochs = 5
- Batch Size = 32
- Training Data Accuracy = 91%
- Validation Data Accuracy = 80%
- Test Data Accuracy = 80%

Hence our current CNN Model can achieve an accuracy of 80% with only 5 epochs of training.

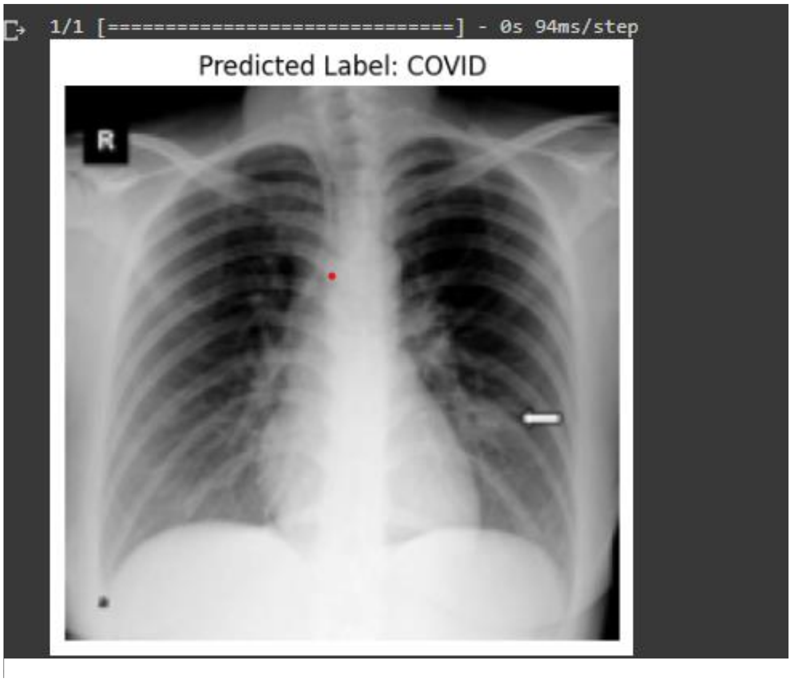
7.5 Testing CNN Model on unseen data

The Convolutional Neural Network was then tested on some unseen images taken from the internet.

To test our CNN model with a random input image, we need to first pre-process the image by resizing it to the appropriate input size and converting it to grayscale. Then, we can pass the pre-processed image through the trained model and obtain the predicted class label. Finally, we can display the image along with the predicted class label to see if the model's prediction matches our expectations.

It's important to note that the random input image should be representative of the types of images that the model was trained on. If the image is significantly different from the training images, then the model's performance on this image may not be indicative of its overall performance. Additionally, we should also evaluate the model's performance on a larger set of test images to get a more accurate measure of its performance.

Test 1: We take a **COVID XRAY** from another dataset and feed it to our model.



```
1/1 [==============================] - 0s 94ms/step
```

Figure 7.1: Input Xray Image and predicted label

We observe that our model successfully predicts the correct label.

Test 2: We take a **NORMAL X-ray** from the internet and feed it to our model.

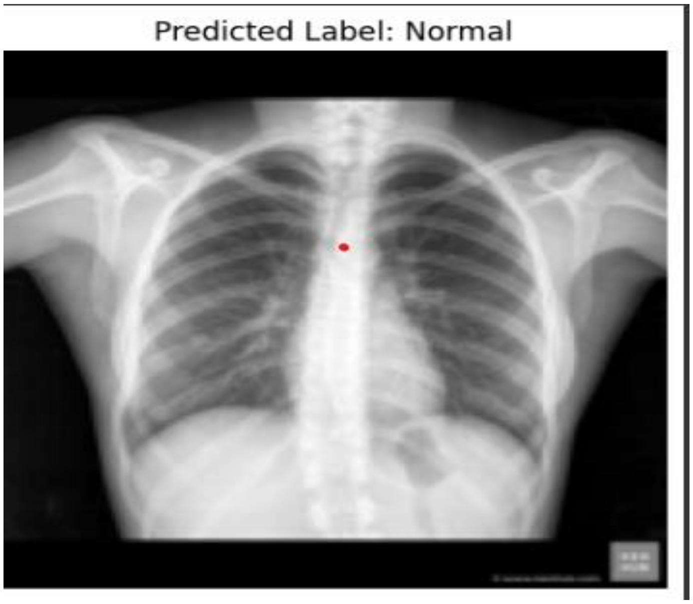


Figure 7.2: Input X-Ray Image and predicted label.

We observe that our model again successfully predicts the correct label.

Test 3: We take a **Viral Pneumonia X-ray** from the internet and feed it to our model.

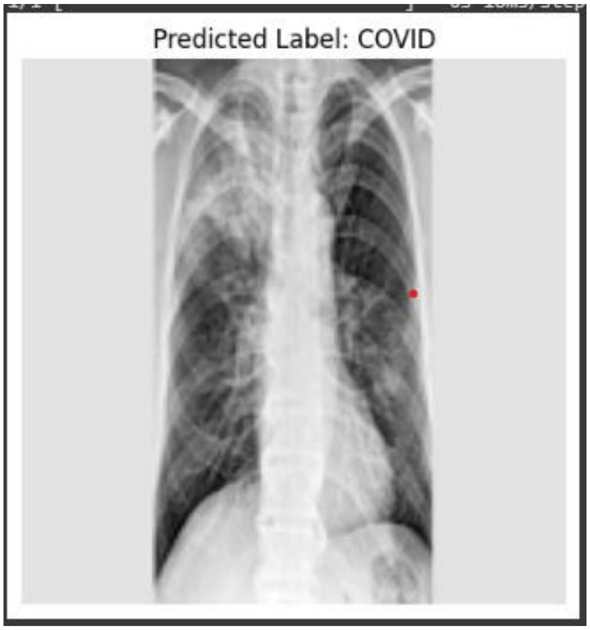


Figure 7.3: Input X-Ray Image and predicted label.

We observe that our model predicts the wrong label.

Testing our CNN model on random input images is a good way to evaluate its performance on data that it has not seen during training. If the model performs well on these images, it suggests that we have successfully trained our model to learn useful features and can generalize well to new data.

Our CNN model predicted the correct label around 80-90% of the time on random input images, it means that we have developed a model that can generalize well to new data. However, we should keep in mind that the performance may vary depending on the quality and complexity of the input images.

It's important to note that evaluating a model's performance on random input images is not a substitute for proper evaluation on a test dataset that is representative of the target population. Therefore, we should always test our model on a separate, representative test set before deploying it in a real-world application.

8. CONCLUSION

For the model, four hyperparameters were taken to test their impact on the accuracy of the model. The hyperparameters were:

- batch size
- size of the filter
- number of epochs
- learning rate

Batch size:

Sizes of values 2^n was taken. Results showed that accuracy was slightly higher for smaller batch sizes.

Number of epochs:

When the number of epochs was increased, there was an increase in accuracy. But as the number of epochs increased, there was no significant change in the accuracy after a certain point.

The hyperparameter with the most significant impact on the accuracy of the model was the learning rate.

Optimal performance can be achieved with the hyperparameters:

- batch size: 32
- size of the filter: 3x3
- number of epochs: 100
- learning rate :0.001

With these values, for the current model the accuracy is at 85% and the model takes just a few minutes to train.

In conclusion, we have found that our use of Principal Component Analysis (PCA) as a pre-processing step in our chest X-ray classifier has been a highly effective technique for improving the accuracy of the classification model. By leveraging PCA to reduce the dimensionality of the original data, we were able to identify the most important features that distinguish between different classes of chest X-rays.

As a result, our classifier has been able to make more accurate predictions, which is especially important in the field of medical diagnosis where timely detection and treatment can have a significant impact on patient outcomes. Additionally, we were able to address the issue of overfitting, which can occur when there are too many features relative to the number of samples in the dataset.

Overall, we believe that our chest X-ray classifier with PCA pre-processing represents a significant advancement in medical imaging and has the potential to improve the accuracy and speed of chest X-ray diagnosis. Moving forward, we will continue to refine and validate our classifier, with the goal of creating a valuable tool for healthcare professionals that can lead to earlier detection and treatment of diseases and ultimately improve patient outcomes.

9. LIMITATIONS AND FUTURE SCOPE

Limitations:

- One limitation of our CNN chest x-ray classifier is that it can only recognize and classify chest x-ray images and is not capable of detecting other diagnostic tools such as CT scans or ultrasounds. This limits the overall diagnostic capability of the model, as it may not be able to provide a comprehensive diagnosis for patients who require multiple types of imaging tests.

- Our current CNN chest X-ray classifier model is limited in its ability to predict only four categories, namely COVID-19, normal, lung opacity, and viral pneumonia. This limitation restricts the model's ability to provide more specific and detailed diagnoses for patients who may have other lung diseases or conditions. For instance, there could be other types of pneumonia, such as bacterial pneumonia, which are not included in the current categories of the model.

- One limitation of CNN chest X-ray classifiers is that they require large amounts of training data to effectively learn and generalize from the input data. Our model was trained on a dataset consisting of 20,000 chest X-ray images, which is relatively small compared to some of the larger datasets used in the field of medical imaging. This limited dataset may affect the performance of our model when tested on a larger and more diverse set of images, especially if there are images with rare or unusual conditions that were not well-represented in the training set.

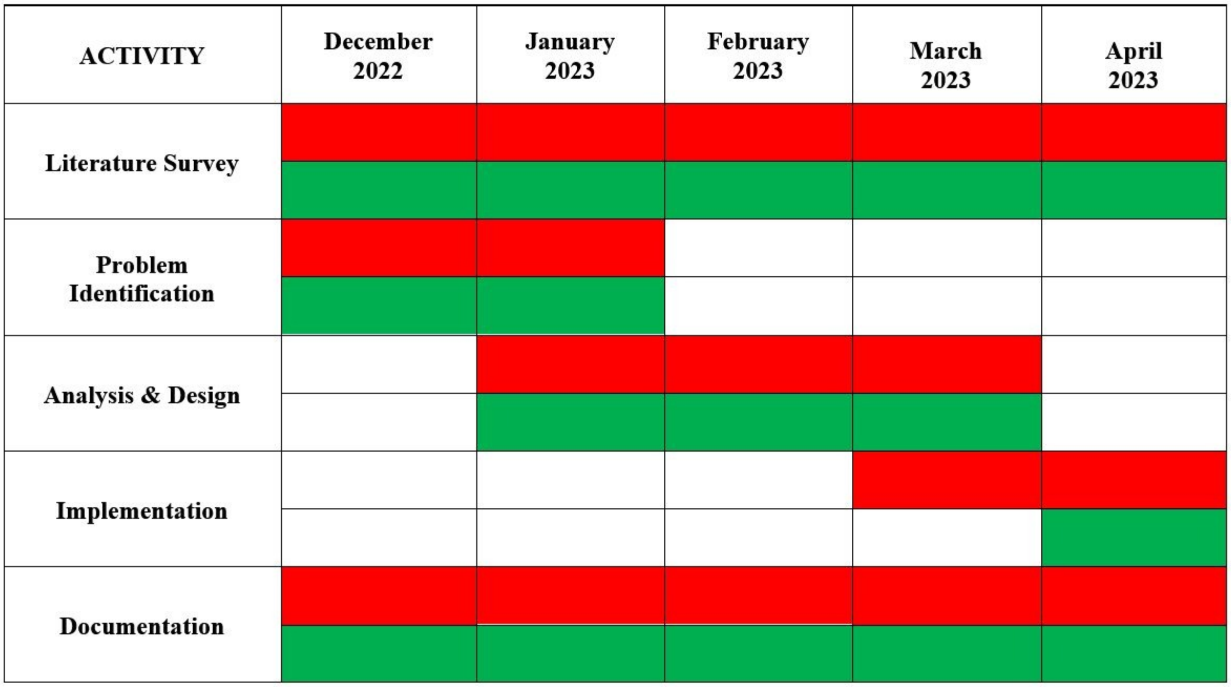
- One limitation of dimensionality reduction and PCA in CXR classification is the loss of information. By reducing the dimensionality of the images and extracting the most relevant features, we may lose important information that could have been useful in the classification process. In some cases, this loss of information can lead to decreased accuracy of the classifier. Another limitation is the selection of the appropriate number of components to retain during the dimensionality reduction process. If we retain too few components, we risk losing important information, while if we retain too many components, we may introduce noise into the model.

- One limitation of CNNs is that they require a lot of processing power, which can make them computationally expensive. Training a CNN on large datasets can take a significant amount of time and require high-performance computing resources. This can make it difficult for researchers with limited resources to use CNNs in their work. Additionally, running predictions on large images or on large batches of images can also be computationally expensive, making real-time or near real-time applications challenging to implement.

Future scope:

- Increasing the size and diversity of the dataset by including more chest radiographs from different regions, demographics, and imaging modalities, which would help improve the model's ability to generalize to new cases.
- Experimenting with different CNN architectures and hyperparameters to achieve better performance on the task of classifying chest radiographs.
- Incorporating additional clinical data such as patient age, gender, and medical history into the model to improve its accuracy and enable it to make more informed diagnostic decisions.
- Developing a web or mobile application to provide a user-friendly interface for healthcare professionals to upload chest radiographs and receive real-time predictions from the trained model.
- Exploring the use of other diagnostic imaging modalities such as CT scans and MRI to improve the overall accuracy of disease detection.

10. GANTT CHART



ACTIVITY	December 2022	January 2023	February 2023	March 2023	April 2023
Literature Survey	🟥	🟥	🟥	🟥	🟥
	🟩	🟩	🟩	🟩	🟩
Problem Identification	🟥	🟥			
	🟩	🟩			
Analysis & Design		🟥	🟥	🟥	
		🟩	🟩	🟩	
Implementation				🟥	🟥
					🟩
Documentation	🟥	🟥	🟥	🟥	🟥
	🟩	🟩	🟩	🟩	🟩

Figure 10.1: Gantt Chart

11. REFERENCES

[1] Tawsifur Rahman, Amith Khandakar, Yazan Qiblawey, Anas Tahir, “Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images”,

[March 2021]

[2] Thet Thet Khaing, Phyu Sin Nyein, Myint Soe Khyaing, Khaing Khaing Wai, “Dimension Reduction of Images Using Principal Component Analysis Algorithm “,

[May 2020]

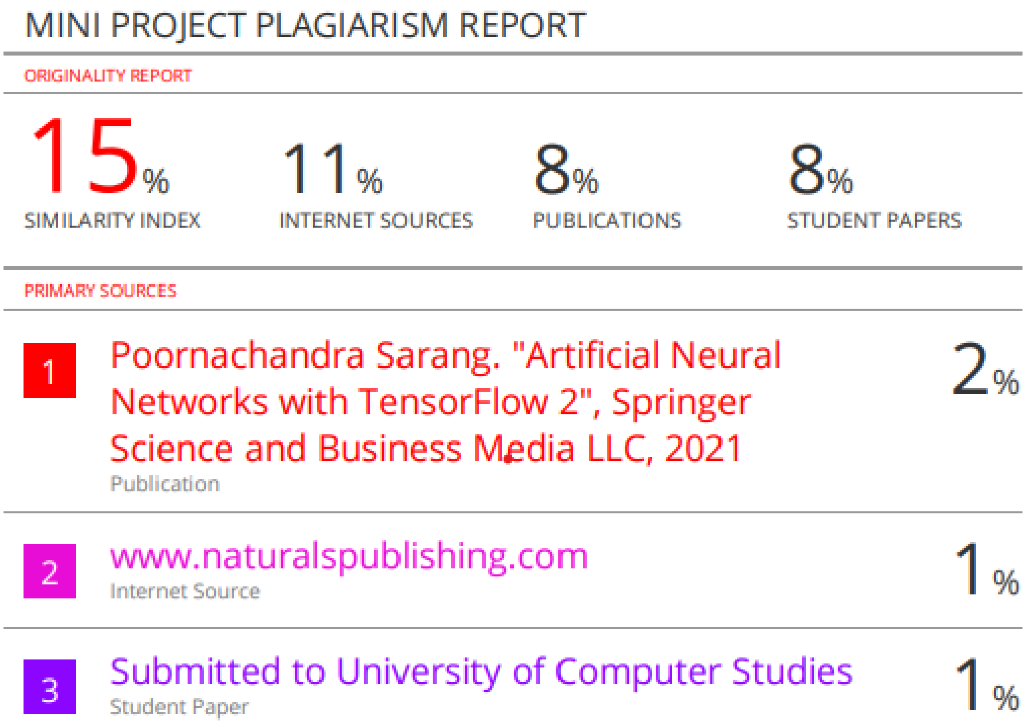
[3] Ayesha, Shaeela; Hanif, Muhammad Kashif, “Overview and Comparative Study of Dimensionality Reduction Techniques for High Dimensional Data”,

[January 2020]

[4] Taufit Rahmat, Azlan Ismail, Sharifah Aliman, “Chest X-Rays Image Classification in Medical Image Analysis”

[December 2018]

12. PLAGIARISM REPORT



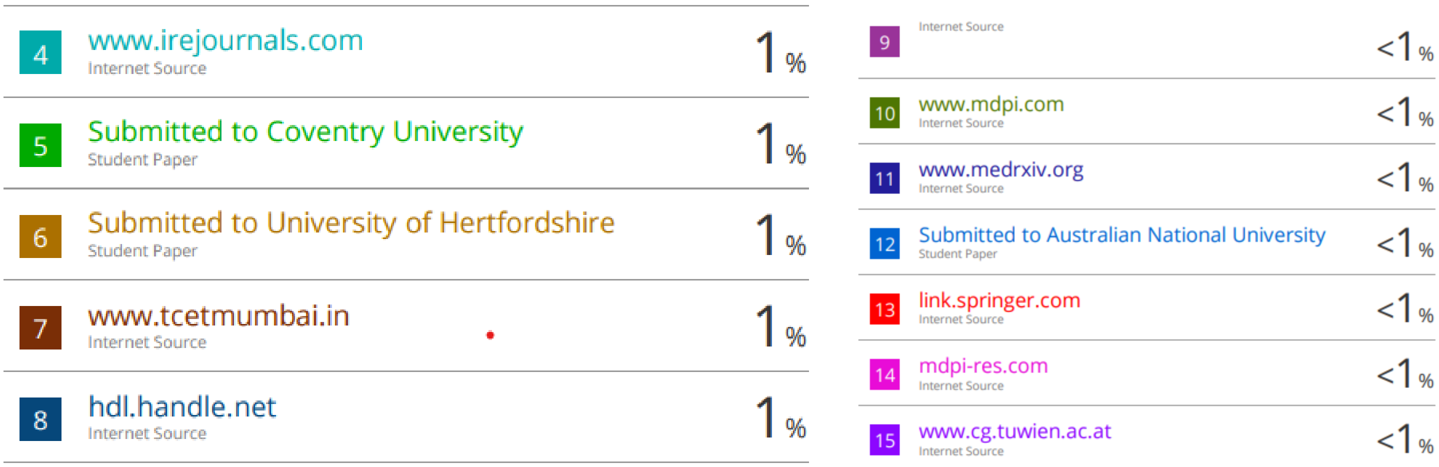
MINI PROJECT PLAGIARISM REPORT

ORIGINALITY REPORT

15%	11%	8%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Poornachandra Sarang. "Artificial Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021 Publication	2%
2	www.naturalspublishing.com Internet Source	1%
3	Submitted to University of Computer Studies Student Paper	1%



4	www.irejournals.com Internet Source	1%
5	Submitted to Coventry University Student Paper	1%
6	Submitted to University of Hertfordshire Student Paper	1%
7	www.tcetmumbai.in Internet Source	1%
8	hdl.handle.net Internet Source	1%
9	Internet Source	<1%
10	www.mdpi.com Internet Source	<1%
11	www.medrxiv.org Internet Source	<1%
12	Submitted to Australian National University Student Paper	<1%
13	link.springer.com Internet Source	<1%
14	mdpi-res.com Internet Source	<1%
15	www.cg.tuwien.ac.at Internet Source	<1%