INTELLIGENT COVID-19 DETECTION FROM CHEST X-RAYS

A PROJECT REPORT

Submitted by,

NAVEETH Z A - (17BIT031)

SHINTA C - (17BIT033)

JANANI SRI - (17BIT056)

In partial fulfilment for the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE 641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **"INTELLIGENT COVID-19 DETECTION FROM CHEST X-RAYS"** is the bonafide work of "**NAVEETH Z A-(17BIT031), SHINTA C-(17BIT033) and JANANI SRI-(17BIT056)"** carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. ALAMELU M, Associate Professor,	**Dr. THIRUMAL P C, Associate Professor,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of Information Technology,	Department of Information Technology,
Kumaraguru College of Technology	Kumaraguru College of Technology
Coimbatore – 641 049	Coimbatore – 641 049

The candidates with University register number **17BIT031, 17BIT033, 17BIT056** were examined in the Project Viva-Voce examination held on …………………

Internal Examiner External Examiner

DECLARATION

We affirm that the project work titled **"INTELLIGENT COVID-19 DETECTION FROM CHEST X-RAYS"** being submitted in partial fulfilment for the award of B. Tech Information Technology is the original work carried out by us. It has not formed the part of any other project work submitted for the award of any degree or diploma, either in this or any other University.

NAVEETH Z A (17BIT031) **SHINTA C (17BIT033)**

JANANI SRI (17BIT056)

I certify that the declaration made above by the candidates is true.

Dr Thirumal P C,

Associate Professor,

Department of Information Technology,

Kumaraguru College of Technology,

Coimbatore – 641 049.

ACKNOWLEDGEMENT

We express our profound gratitude to the management of Kumaraguru College of Technology for providing as with the required infrastructure that enabled us to successfully complete the project.

We extend our gratefulness to our Principal, **Dr. Saravanan D,** for providing us the necessary facilities to pursue the project.

We would like to acknowledge **Dr. Alamelu M,** Associate Professor and Head, Department of Information Technology, for her support and encouragement throughout this project.

We thank our project coordinators **Dr. Thirumal P C**, Associate Professor, **Dr. Nandakumar G S,** Associate Professor, Department of Information Technology and guide **Dr. Thirmal P C,** Associate Professor, Department of Information Technology, for their constant and continuous effort, guidance and valuable time.

Our sincere thanks to staff of Department of Information Technology of Kumaraguru College of Technology for their well wishes, timely help and support rendered to us during our project. We are greatly indebted to our family, relatives and friends, without whom life would have not been shaped to this level.

- NAVEETH Z A

SHINTA C

JANANI SRI

ANNA UNIVERSITY: CHENNAI 600 025

TABLE OF CONTENTS

ABSTRACT

It has been more than a year since the first COVID-19 was identified in Wuhan, China. Still, we have not found a faster method which is at the same precision for the detection of COVID-19 disease. There is an urge for new methods because the disease is spreading at a faster pace and it is significant to save the lives of the infected as soon as possible in case of infection. For this reason, we can seek the help of Deep Learning techniques besides the traditional methods for accurate detection of the coronavirus disease at a faster pace. Also, by using these techniques, we can reduce human errors and other medical expenses that occur during the detection of the virus. In this project, we have used the architecture of a deep learning model, the Convolution Neural Network to detect the virus and spread awareness about the virus. This project is an encapsulation of detecting the virus and a place where we can get the overall information about the virus obtained from legitimate sources.

1. INTRODUCTION

1.1 CONCEPTUAL STUDY OF THE PROJECT

This project is about to build a web application wherein the users can upload their X-Ray as an image file and the result is displayed to the user. The output is the classification labels, COVID, Viral Pneumonia or Normal which depends on the X-Ray image uploaded. Also, some information regarding the COVID-19, do's and don'ts and many such awareness is also displayed to the user. Along with this, the last two day's recovered, deceased and new cases are also displayed for the user. This information is obtained from legitimate sources. Users can find all the necessary significant information regarding the virus inside this single web application.

The highly contagious coronavirus is spreading at a faster pace. The respiratory droplet of the infected person is the medium of spread. The person infected with the virus will suffer from respiratory illness and if the infection is acute, it may lead to death. This creates a need to detect the positive cases accurately as soon as possible in order to reduce the mortality rate.

The current methods which are utilized to detect the virus are: RT-PCR (Reverse Transcription Polymerase Chain Reaction) where the RNA is tested for the presence of the virus using a cotton swab, Saliva test where the saliva of a person is taken in a test tube or the sample is collected using a cotton swab and the RNA of the virus is converted to its DNA and then the DNA is amplified for further processes, Antigen Test which is similar to the RT-PCR but it is more likely to miss active infection, Antibody Test which is conducted on blood samples test for proteins or antibodies that our immune system secretes in order to fight off the virus. These are the most commonly used techniques and there are other methods as well. These methods have many severe limitations that put the lives of many people at risk. One such limitation is these methods are not accurate all the time as they produce many false negatives thereby putting lives of the infected at risk. These methods are very slow in detecting the virus and as the virus spread at a faster pace, there is a need for other faster methods for detection. Also, in the above methods, medical specialists are required to conduct the testing thereby putting their lives at a greater risk because they are in close contact with the infected person.

These limitations could be overcome by CXR (Chest X-Ray) images and CT (Computed Tomography) scans as a substitute and by using deep learning methodologies like Convolution Neural Network models to detect the virus in those images. This technique takes a few minutes to issue the result and at the same time it is very accurate thereby, reducing errors caused by humans to a greater extent. These images can be used not only to diagnose COVID-19 virus but also other lung diseases like viral or bacterial Pneumonia that might cause similar imaging as that of COVID. Use of technology in the medical field has improved the accuracy and also it reduces the risk taken by medical practitioners to a greater extent.

For this project, CXR images are collected from various resources to classify the uploaded images as COVID, normal and pneumonia. For the training and testing of dataset, a Convolution Neural Network model is proposed. This deep learning model accurately classifies the images into one of the three categories as mentioned above.

1.2 OBJECTIVES OF THE PROJECT

To develop a fully automatic framework to detect COVID-19 using Chest X-Rays. To develop a model that uses Deep Learning to detect coronavirus disease 2019 (COVID-19) and distinguish it from normal and community acquired pneumonia affected X-Rays.

1.3 SCOPE OF THE PROJECT

- Detect COVID-19 using chest X-Ray images instantly
- Detect COVID-19 without much human intervention
- Additional and necessary information about the disease can be known
- Usage of Machine Learning algorithms in medical field

2. LITERATURE REVIEW

In [1] the authors made use of the transfer learning model for the COVID-19 diagnosis. This model included two pretrained models: ResNet-34 and ResNet-50 which were primarily trained on the ImageNet dataset that consisted of 3.2

million images for the classification purposes. An accuracy of 66.67% on ResNet-34 and an accuracy of 72.38% on ResNet-50 was obtained.

Further, in [2] the authors compared three different deep learning models such as: Inception net V3, Xception net and ResNeXt for the COVID-19 diagnosis. For this purpose, 6432 COVID-19 images and Pneumonia CXR images were used. A 97 percent accuracy rate was achieved using the Xception net model and it is considered to be the best model among the other two models.

In [3] the authors used Ensemble Learning models that were based on a weighted average that contained three pre-trained CNN models which includes: DenseNet201, ResNet50V2 and Inception V3. Adam optimizer was used for faster convergence in all the three models. An accuracy of 95.7% and a sensitivity of 98% were obtained.

Further, in [4], the authors compared the performance of ConvNet models using Experimentation Learning methodology. The experiments conducted include: Transfer Learning experiments, ConvNet, and Statistical measurement. These experiments were sub categorized as: Covid-19/normal/pneumonia, Covid-19/normal, Covid-19/pneumonia and. 94.10% efficiency was obtained using one of the ConvNet architectures.

In [5], the authors proposed a transfer learning model, a fully automated diagnosis method to detect Covid-19 infection. Four pre-trained models were utilized which include VGG16, DenseNet121 [25], ResNet50 [26] and VGG19 [27]. The ImageNet dataset was used to pre-train the models, and the X-Ray dataset was used to further train the models. The number of epochs for each model was set to 30, and the ReLU activation function was used to expand the neutron feature extraction range. VGG19 achieved 100% sensitivity and 99.33% accuracy.

Further, in [6], the authors trained and validated 29 AI-based models using Transfer Learning approach. A number of hyperparameters including Number of epochs, Filter size, Number of filters, fully connected layers, Number of Iterations, Image size, internal validation size and Batch Size. The datasets were augmented using augmentation techniques due to the need for a large amount of data and the restricted availability of COVID-19 CXR images. The model that made use of 101 epochs achieved 93.8% accuracy.

3. PROBLEM DEFINITION

COVID-19 is posed as very infectious and deadly pneumonia type disease until recent time. Novel coronavirus or SARS-COV-2 strain is responsible for COVID-19 and it has already shown the deadly nature of respiratory disease by threatening the health of millions of lives across the globe. Clinical study reveals that a COVID-19 infected person may experience dry cough, muscle pain, headache, fever, sore throat and mild to moderate respiratory illness. At the same time, it affects the lungs badly with virus infections. So, the lung can be a prominent internal organ to diagnose the gravity of COVID-19 infection using X-Ray and CT scan images of chest. Despite having lengthy testing time, RT-PCR is a proven testing methodology to detect coronavirus infection. Sometimes, it might give more false positive and false negative results than the desired rates.

4. PROPOSED SYSTEM

4.1. DATA PRE-PROCESSING AND AUGMENTATION

Data augmentation is a technique that uses random transformation to increase the variance of our dataset. Having a huge dataset is vital to achieve effectiveness in the performance of the deep learning model. So we performed various data augmentation techniques like rescaling, zooming, height and width shifting and flipping since the size of our dataset is very small and also to avoid the problem of model over-fitting. The rescaling function reduces or magnifies the image during the augmentation process. The angle of the image is clipped in an anticlockwise direction with a shear width of 0.2 percent. The zoom range zoom the picture in and out at a rate of 0.2 percent at random. Then the height of the image is shifted with the factor of 0.2 and width of the image is also shifted with the factor of 0.2. Furthermore the images are randomly rotated to 40 degrees. Finally the images are flipped horizontally. After this, the augmented images are resized to the size of 224 X 224.

4.2. DATA SPLITTING

A total of 8252 images are divided into training dataset of 8115 images and the testing dataset of 137 images. Then both the training and testing dataset are segregated into three sets namely normal, covid and viral pneumonia images. In the training dataset, there are 2692 normal CXR images, 1118 covid CXR images and 4305 viral pneumonia CXR images. There are 30 covid CXR images, 65 viral pneumonia CXR images and 42 normal CXR images in the testing dataset. Thus the dataset is set up to train the model which is followed by testing the efficiency of the trained model.

4.3. PROPOSED MODEL

One of the most renowned architects in deep learning is the Convolution Neural Network (CNN). It is now a go-to model for every image analysis and image classification problem. AlexNet is a convolution neural network which had a large impact on the application of deep learning to computer vision. Convolution layers and fully connected layers make up a standard AlexNet architecture. Inspired by the AlexNet architecture, combining one or more such layers, we have designed an automated Covid-19 detector that detects Covid-19 from CXR images.

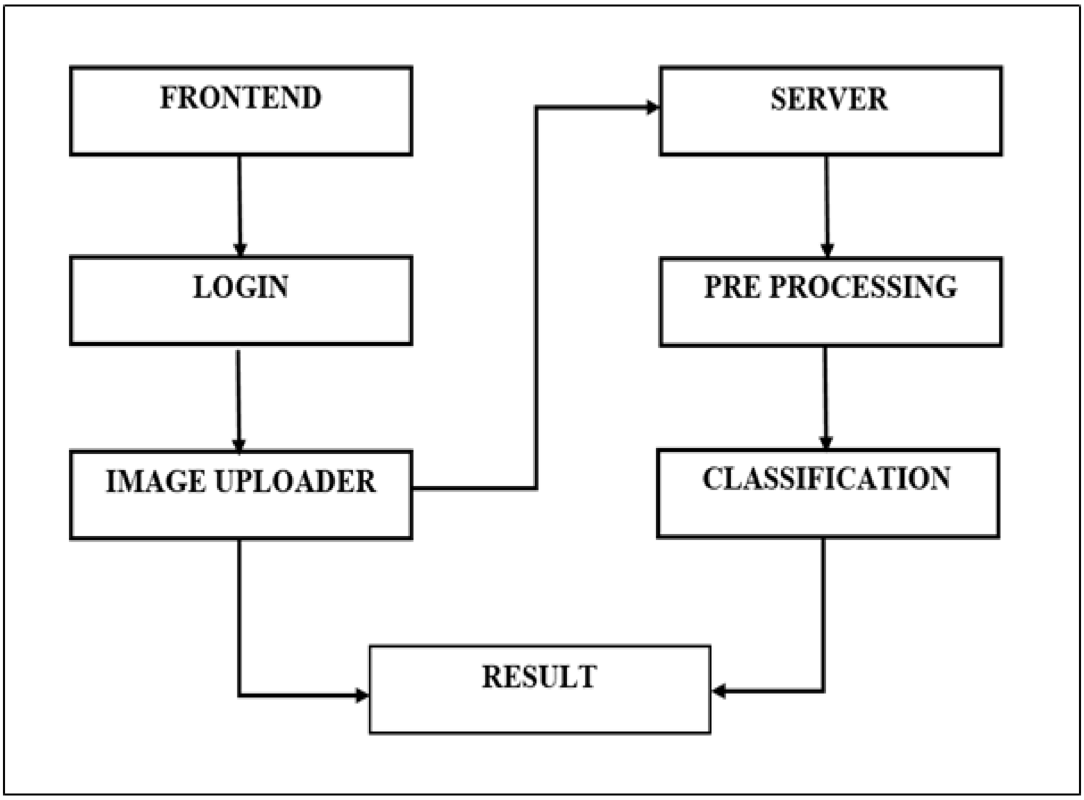
Convolution, max-pooling, normalization, and fully connected layers make up the proposed architecture, with the Relu activation feature added after each convolution and fully connected layer. Each layer accepts the output of the layer immediately preceding it as input. The image size is fixed to 224 X 224 X 3. There are five convolution layers that is conv2D layers with each layer having 96 kernels of size 11 X 11, 256 kernels of size 11 X 11, 384 kernels of size 3 X 3, 384 kernels of size 3 X 3 and 256 kernels of size 3 X 3 respectively. A stride of 4, 1 and 2 is applied to convolution layers. The convolved images are subjected to batch normalization and rectified linear unit (ReLU). Batch normalization standardizes the input. ReLU activation function is used to avoid model over-fitting. Three max pooling layers are then applied to sample the height and width of tensors, keeping the depth the same. We used max pooling windows with dimensions of 2 X 2 and strides of 2 between adjacent windows. Dropout increases the number of iterations required to converge in a fully connected layer by a factor of 0.2 and 0.4 respectively. Finally this ends with a SoftMax layer that produces the output. Weights are generated using SGD optimizer with 0.001 initial learning rates and 0.9 momentum. The model performs the task of classification into three different labels namely covid, viral pneumonia and normal.

Table 1. The specifics and the parameters of the layers of the proposed model

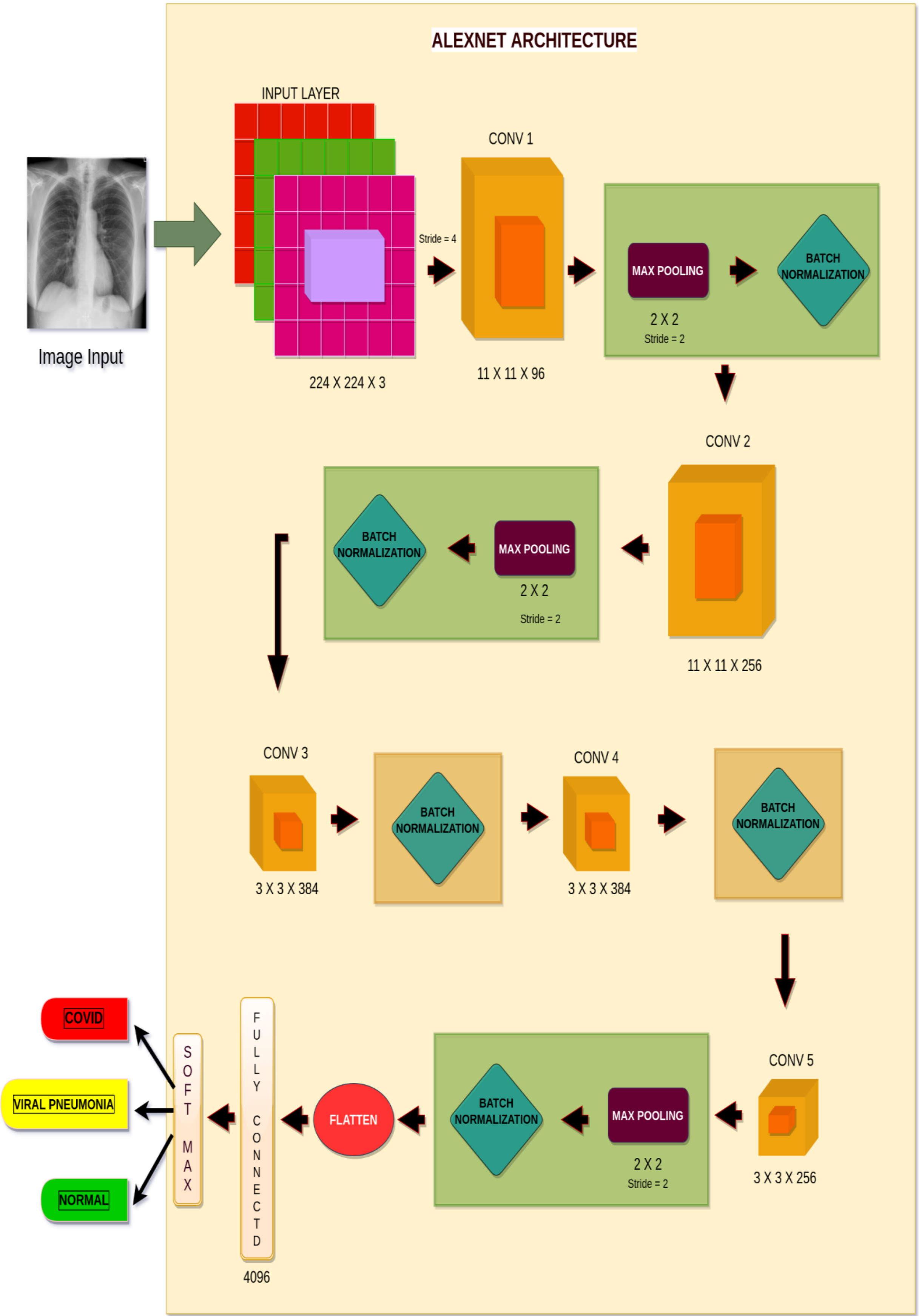
Order of Layer	Type of Layer	Output Shape	Parameters
1	Conv2D	[54, 54, 96]	34944
2	Max Pooling	[27, 27, 96]	0
3	Batch Normalization	[27, 27, 96]	384
4	Conv2D	[17, 17, 256]	2973952
5	Max Pooling	[8, 8, 256]	0
6	Batch Normalization	[8, 8, 256]	1024
7	Conv2D	[6, 6, 384]	885120
8	Batch Normalization	[6, 6, 384]	1536
9	Conv2D	[4, 4, 384]	1327488
10	Batch Normalization	[4, 4, 384]	1536
11	Conv2D	[2, 2, 256]	884992
12	Max Pooling	[1, 1, 256]	0
13	Batch Normalization	[1, 1, 256]	1024
14	Flatten	[256]	0
15	Dense	[4096]	1052672
16	Dropout	[4096]	0

17	Batch Normalization	[4096]	16384
18	Dense	[4096]	16781312
19	Dropout	[4096]	0
20	Batch Normalization	[4096]	16384
21	Dense	[1000]	4097000
22	Dropout	[1000]	0
23	Batch Normalization	[1000]	4000
24	Dense	[3]	3003

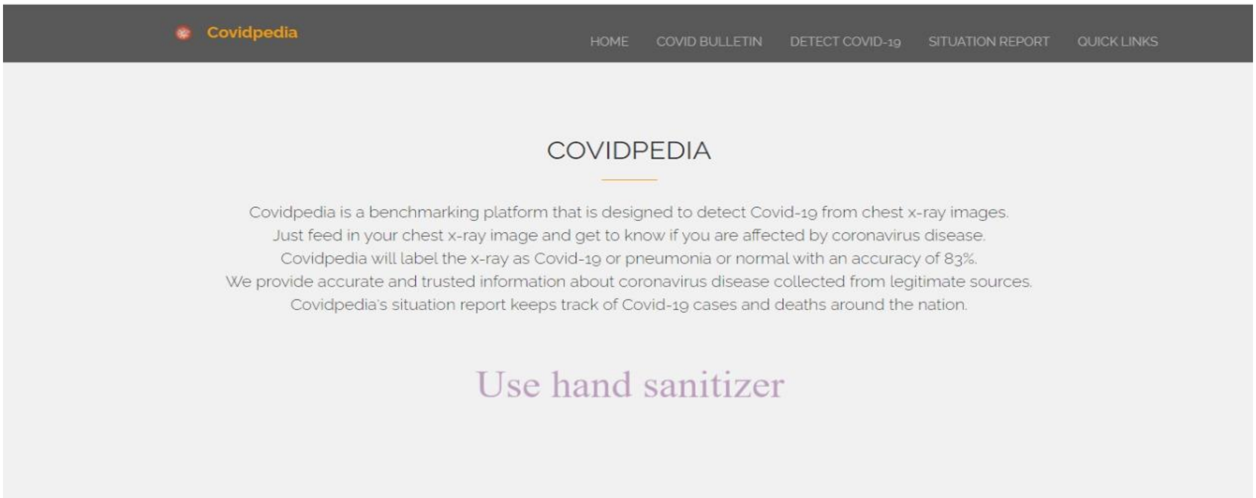
4.4 BLOCK DIAGRAM

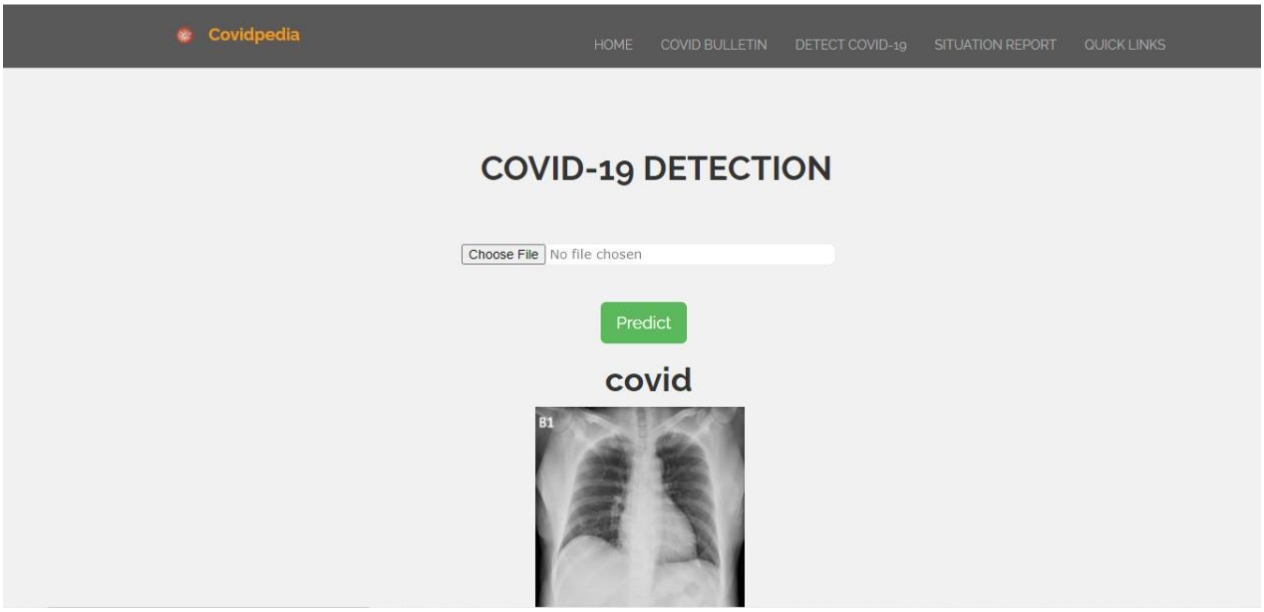


4.5 ARCHITECTURE



4.6 IMPLEMENTATION





5. SYSTEM REQUIREMENTS

OPERATING SYSTEM	Windows 10
SOFTWARE PACKAGE	Pycharm Professional
LANGUAGE	Python
BACKEND FRAMEWORK	Django
FRONTEND	Django Templates
LIBRARIES	Torch vision, Torch, Matplotlib, Numpy, PIL

6. PAPER PUBLICATION

6.1. SURVEY PAPER

A survey paper on the title “SURVEY ON INTELLIGENT COVID-19 DETECTION FROM CHEST X-RAYS” was published in Volume 8 of the International Research Journal of Engineering and Technology (IRJET). In this paper, we have systematically summarized, analysed and compared the datasets and techniques proposed by researchers to detect coronavirus disease using chest X-Rays. Below table tabulates the comparison result of Covid-19 detection strategies that were reviewed in our survey paper.

CITATION	TECHNIQUE	RESULTS
Ravneet et al. [1]	Transfer learning of pre-trained ResNet-34 and ResNet-50 models	66.67% accuracy on using ResNet-34 and 72.38% on using ResNet-50
Rachna et al. [2]	Deep Learning of Inception net V3,Xception Net and ResNeXt	97% accuracy on using Xception Net.
Amit et al. [3]	Ensemble of pre-trained DenseNet201, ResNet50V2 and InceptionV3	95.7% accuracy and 98% sensitivity
Boran et al. [4]	Experimentation Learning of ConvNet, Statistical measurement and Transfer Learning Experiments	94.10% efficiency using one of ConvNet architectures.
Irfan et al. [5]	Transfer learning of pre-trained DenseNet121, ResNet50, VGG19, VGG16.	VGG19 Achieved 100% sensitivity and 99.33% accuracy
Arun Sharma et al. [6]	Transfer learning using AI-based models.	Combined dataset model which used 101 epochs achieved 93.8% accuracy

The below table lists the research databases utilized in our research process.

DATABASES	URL
IEEE Xplore	https://ieeexplore.ieee.org/
Research Gate	https://www.researchgate.net/
Springer Link	https://link.springer.com/
MDPI	https://www.mdpi.com/
Hindawi	https://www.hindawi.com/
SAGE journals	https://journals.sagepub.com/

Throughout the paper, we have studied various ways to detect the virus in X-Ray images thereby reducing the delay in detection and other limitations using the current method. Convolutional Neural Networks, Deep Learning, and Machine Learning algorithms were used to analyze X-Ray images. Majority of the papers that we have surveyed not only detects Covid-19 virus but also differentiates pneumonia and normal X-Rays. This diagnosis is very essential as this helps patients and doctors treat the disease if any, at an early stage. From our survey, it is deductible that since it has only been a few months after the outbreak of the pandemic, the number of datasets available is quite little. With many more X-Ray images, more accurate results can be obtained in the future. In all the papers, performance was measured based on the F1 score, mean accuracy, and other computational complexities of the models.

6.2. RESEARCH PAPER

We published a research paper on the title “INTELLIGENT COVID-19 DETECTION FROM CHEST X-RAYS” in Volume 8 of the International Research Journal of Engineering and Technology (IRJET). This paper included the relevant works, dataset description, proposed approach, performance metrics used, experimental results, tools used, description of the proposed prototype tool and conclusion.

7. CONCLUSION

Thus, using the Convolution Neural Network architecture, we are able to successfully classify the given CXR image file into normal, covid and viral pneumonia. The project is built from scratch using the AlexNet Convolutional Neural Network to train the Chest X-Ray images. Along with classifying the images, the information available on our website will be really helpful to get an overall insight about the spread of the virus and also the necessary safety measures suggested by renowned medical practitioners. This project can also be extended in the future to detect other lung related diseases without much human intervention. Also, the average accuracy can be increased by using more CXR images in both the training and testing phases. With this automation of detecting the presence of Covid, we hope that health care delivery will be improved, saving time as well as the lives of the infected and also the medical practitioners.

8. APPENDIX

Covid_detection.py

```
# importing keras libraries and packages
from keras.models import Sequential
from keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense,
Dropout
from keras.layers.normalization import BatchNormalization
from keras import optimizers
from keras.preprocessing.image import ImageDataGenerator
from keras.callbacks import ModelCheckpoint

#initializing the CNN
classifier = Sequential()
#convolution step 1
classifier.add(Convolution2D(96,11,strides=(4,4),padding='valid',input_shape=(
224,224,3),activation='relu'))

#max pooling step 1
classifier.add(MaxPooling2D(pool_size=(2,2),strides=(2,2),padding='valid'))
```

```
classifier.add(BatchNormalization())

#convolution step 2
classifier.add(Convolution2D(256,11,strides=(1,1), padding='valid',
activation='relu'))
#max pooling step 2
classifier.add(MaxPooling2D(pool_size=(2,2),strides=(2,2),padding='valid'))
classifier.add(BatchNormalization())
#convolution step 3
classifier.add(Convolution2D(384,3,strides=(1,1),padding='valid',activation='rel
u'))
classifier.add(BatchNormalization())
#convolution step 4
classifier.add(Convolution2D(384,3,strides=(1,1),padding='valid',activation='rel
u'))
classifier.add(BatchNormalization())
#convolution step 5
classifier.add(Convolution2D(256,3,strides=(1,1),padding='valid',activation='rel
u'))
#max pooling step 3
classifier.add(MaxPooling2D(pool_size=(2,2),strides=(2,2),padding='valid'))
classifier.add(BatchNormalization())
#flattening step
classifier.add(Flatten())

#full connection step
classifier.add(Dense(units=4096, activation='relu'))
classifier.add(Dropout(0.4))
classifier.add(BatchNormalization())
classifier.add(Dense(units=4096, activation='relu'))
classifier.add(Dropout(0.4))
classifier.add(BatchNormalization())
classifier.add(Dense(units=1000,activation='relu'))
classifier.add(Dropout(0.2))
```

```
classifier.add(BatchNormalization())
classifier.add(Dense(units=3,activation='softmax'))
classifier.summary()
#compiling the CNN
classifier.compile(optimizer=optimizers.SGD(lr=0.001, momentum=0.9,
decay=0.005), loss='categorical_crossentropy', metrics=['accuracy'])

#image preprocessing
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   width_shift_range=0.2,
                                   height_shift_range=0.2,
                                   rotation_range=40,
                                   horizontal_flip=True,
                                   fill_mode='nearest')
test_datagen = ImageDataGenerator(rescale=1./255)
batch_size = 32

train_data_dir = "#training dataset path"
test_data_dir = "#testing dataset path"
training_set = train_datagen.flow_from_directory(train_data_dir,
                                                 target_size=(224, 224),
                                                 batch_size=batch_size,
                                                 class_mode='categorical')
test_set = test_datagen.flow_from_directory(test_data_dir,
                                            target_size=(224, 224),
                                            batch_size=batch_size,
                                            class_mode='categorical')
print(training_set.class_indices)

#fitting images to CNN
history = classifier.fit_generator(training_set,
                                   steps_per_epoch=training_set.samples//batch_size,
```

```
                    validation_data=test_set,
                    epochs=50,
                    validation_steps=test_set.samples//batch_size)

#saving model
filepath = "classify/model.hdf5"
classifier.save(filepath)

# plotting training values
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
acc = history.history['acc']
val_acc = history.history['val_acc']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs = range(1, len(loss)+1)

#accuracy plot
plt.plot(epochs,acc,color='green',label='Training Accuracy')
plt.plot(epochs, val_acc, color='blue', label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()

#loss plot
plt.plot(epochs,loss, color='pink',label= 'Training Loss')
plt.plot(epochs, val_loss, color='red', label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.show()
```

REFERENCES

[1] Ravneet Punia, Lucky Kumar, Mohd. Mujahid and Rajesh Rohilla "Computer Vision and Radiology for COVID-19 Detection". In: 2020 International Conference for Emerging Technology (INCET), Belgaum,India. Doi: 10.1109/INCET49848.2020.9154088.

[2] Rachna Jain, Meenu Gupta,Soham Taneja and D Jude Hemanth "Deep Learning based detection and analysis of COVID-19 on chest X-Ray images".In: Applied Intelligence(2020). Doi: 10.1007/s10489-020-01902-1.

[3] Amit Das, Sayantani Ghosh, Samiruddin Thunder, Rohit Dutta, Sachin Agarwal and Amlan Chakrabarti "Automatic COVID-19

[4] Detection from X-Ray images using Ensemble Learning with Convolutional Neural Network". Doi: 10.21203/rs.3.rs-51360/v1Boran Sekeroglu and Ilker Ozsahin "Detection of COVID-19 from Chest X-Ray Images using Convolution Neural Networks". Doi: 10.1177/2472630320958376.

[5] Irfan Ullah Khan and Nida Aslam "A Deep-Learning-Based Framework for Automated Diagnosis of COVID-19 Using X-ray Images"

[6] In: Information 2020,11,419. Doi:10.3390/info11090419

[7] Arun Sharma, Sheeba Rani and Dinesh Gupta "Artificial Intelligence-based classification of Chest X-Ray images into COVID-19 and other infectious diseases" In: International Journal of Biomedical Imaging. Doi: 10.1155/2020/88890.