Detection of COVID 19 using Deep Neural Network and Android App Development

***Abstract*—**Automated detection of highly communicable diseases such as COVID-19 will reduce the risk of the disease spreading. Detection of such disease requires a great amount of accuracy to ensure the safety of the patient and doctor as well. To provide an efficient solution to the problem, this model has proposed a robust solution to detect and classify diseases such as Covid-19 and Pneumonia for testing personnel which include doctors, different pathology laboratory personnel, etc. The work is divided into 2 parts, first is image classification into the above-mentioned categories using deep neural networks and the second is adding the obtained model for the said deep neural network to an android application using Android Studio. This algorithm classifies the given image into 3 categories namely Covid-19, Pneumonia, and normal. The classifier architecture is created using CNN (Convolutional Neural Network) and it is trained on 10725 images, each of size 128 x 128. This dataset is a collection of datasets taken from Kaggle repositories in their open challenges. The customized neural network model was able to provide training accuracy of 95.04% and testing accuracy of 94.14%.

***Keywords*—**COVID-19,pneumonia,X-ray,convolutional neural networks, coronavirus**.**

I. Introduction

Automated detection of diseases requires great accuracy and the system should be efficient as well. Any application made from such a system should be able to classify the given input image into an infected or healthy one based on the given input training parameters. A lot of diseases are present in the world. Out of such diseases, some are curable and some are not. Due to this factor, if the system makes an incorrect prediction about the image from a patient, it could result in a huge loss of life cause it may lead to a worldwide pandemic.

There are various types of disease classification systems present in the world. In [1], the authors divided their experiments into 3 categories namely ConvNet, statistical measurement, and transfer learning experiments. In the ConvNet experiment, they used different convolutional layers consisting of different filters for classification. In the statistical measurement experiment, they obtained basic statistical information and preprocessed characteristics from the images. The concept behind doing so was that each image has some basic statistical information hidden in them which was useful for training machine learning models. After the first 2 experiments, the images were given to a pretrained network such as VGG16[2], RESNET50[3], inceptionV3[4], DenseNet50[5], MobileNet-V2[6] (Transfer Learning Experiment). The system used the Convolutional layer of different kernel sizes and filters.

In the [7], the authors used 3 different networks namely InceptionNet V3, XceptionNet, and ResNeXT. InceptionNet V3 is a CNN-based network for classification. It decreases the number of parameters and increases training speed. XceptionNet is a modification of InceptionNet. Though it has the same parameter size as that of inception net it performs slightly better than InceptionNet. In ResNeXT, the blocks are replaced with such blocks that follow a split, transform, merge strategy which is used in the inception model.

In [8], the authors have used a DarkNet 19 model which was available on Github instead of building a model from scratch. The DarkNet classifier forms the base of a real-time object detection system called YOLO[9]. YOLO stands for you only looking once. They used fewer filters and layers in the beginning and later increased it as per their need.

In [10], Work on detection of Covid-19, Normal and Pneumonia using X-ray images have been done by students of the Electronics and Communication Department, Delhi Technological

University, Delhi, India. They published the paper in the IEEE with the name 'Computer Vision and Radiology for COVID-19 Detection'. In this paper, the X-Ray images of COVID-19 were extracted from online hosted data by an Italian research organization[11], European Health Care. The Pneumonia X-Ray Images were collected from the open-source dataset. After removing the noisy images, a dataset with images for three labels COVID-19, PNEUMONIA, and NORMAL with 374 images in each category was extracted. They performed rotation, zooming, height shifting and resized to 224 X 224.

For creating a deep learning model they had considered two architectures one was ResNet-34 and the other was ResNet-50. According to the data presented in the paper in the ResNet-34 model they were able to achieve an accuracy of 66.67% with an error rate of 33.33% and in the ResNet-50 model, they were able to achieve an accuracy of 72.38% with an error rate of 27.62%.

Various algorithms were applied to the dataset out of which customized CNN algorithms have yielded the maximum accuracy among ResNet-34, ResNet-50. The trained convolutional neural network model was then added to an android application for easy and simple prediction using x-ray images.

II. Dataset

All deep neural networks or CNN models require a huge amount of images to train and validate. If the images are not enough and running a large number of iterations can result in overfitting of the model. The dataset is a huge collection of the dataset from a Machine Learning website called Kaggle. Three different datasets from Kaggle's challenge were used**.** After gathering all the datasets, a new dataset was created comprising all the gathered datasets, but all the repeated images were removed using a simple block of code. The new dataset consisted of unique images only. This dataset consisted of 3 major subcategories namely COVID 19, Normal, and Pneumonia. Each sub-category has about 3575 images comprising a total of 10725 images for training and validation have about 1549 images individually making it about 4647 images in total making it to 43% of training images. A sample of the training image is shown below.

Fig 1.1 Covid 19

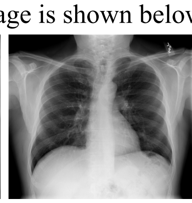

Fig 1.2 Normal

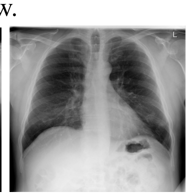

Fig 1.3 Pneumonia

The images for training have resolutions ranging from 705 x 370 to 1024 x 1024. Since the architecture samples down the image into 128 x 128, now the resolution of images was not a problem. The detection model is trained on the above-mentioned dataset. Classification models of CNN or Deep neural networks required a huge amount of images to classify into different categories with suitable or acceptable accuracy.

III. Proposed Approach

The intuition behind developing the model is to create a solution that solves the issue of detecting the Covid-19 patients from the patients having pneumonia and normal people using the X-ray images. As in the above section of Related work the paper of IEEE on 'Computer Vision and Radiology for COVID-19 Detection' was able to reach a maximum accuracy of 72.28% with the architecture such as ResNet. So the plan was to create a customized deep neural network model for the detection of the Covid-19 using X-ray images. The aim was to create a model with the best accuracy possible. The entire project was partitioned into two parts. First is the development and training of an efficient deep neural network algorithm for the classification of X-ray images and the second part is the development of an android application with the trained deep neural network model added to it. For the development of the deep learning model, the customized CNN (Convolutional Neural Network) was used for training and improving the accuracy to a good extent. This classification model classifies the given input image into three categories i.e. Covid-19, Normal, Pneumonia. The user interface of the android application is very simple to be used by new users.

Architecture and Implementation of Deep Neural Network

Various models were implemented, among which the customized deep neural network model has the highest accuracy among the other models such as ResNet-34 and ResNet-50. The ResNet-34 and ResNet-50 had a training accuracy of 71.62% and 79.47% respectively. The Final customized deep neural network model has a training accuracy of 95.04% which is far better than the ResNet-34 and ResNet-50. For the training purpose of the model, the Jupyter notebook was used as a development platform and Python 3 as the programming language for the development of the deep neural network.

The data set for training and validation consists of 10725 images and 4647 images respectively, belonging to three classes Covid-19, Normal, and Pneumonia. Before the training of the deep neural network model, the images were scaled down to (128 X 128). The batch size was 32. Keras data generator was used for preprocessing. First, the basic model was built having 3 convolution 2d layers with activation function as 'Relu. Each convolution layer was followed by the max-pooling 2d layer. For the initial phase, the kernel size was (5 X 5) and filters = 128 for the first layer. The input to the first layer was (128 X 128 X 3). For the second layer of convolution 2d, kernel size of (3 X 3) was used and filters = 64 and for the third convolution layer, kernel size of (3 X 3)was used and filters = 32. After these 3 convolution 2d layers there were four more layers added. Flatten layer was used to unstack all this multidimensional tensor into a very long 1D tensor, followed by a dense layer with an output size of 64 and activation ‘ Relu’. Then a dropout layer was used with a rate = 0.5. Then again a dense layer with an output size of 3 with activation function ‘Softmax’. This was the primary model for training. This model was able to reach an accuracy of 88.34%. For improving the accuracy furthermore, hyperparameter tuning was used. So various changes were implemented and Keras tuner was used for hyperparameter tuning.

After hyperparameter tuning, the model was able to reach a training accuracy of 95.04% and validation accuracy of 94.14%. The improved model consists of 3 convolution 2d layers. The first layer consists of (5 X 5) kernel size, filters = 96, and the activation function is ‘Relu’. The second convolution 2d layer consists of kernel size of (3 X 3), filters = 48, and activation function is ‘Relu’. The third convolution 2d layer consists of (3 X 3) kernel size and filters = 64 with activation function ‘Relu’. All the three convolution 2d layers were followed by a max-pooling layer with a pool size of (2,2). Then there is a flatten layer, dense layer with an output size of 64, and activation ‘Relu’ as described in the initial model. In the initial model dropout rate of 0.5 was used, but the model was overfitting so to overcome this problem a dropout rate of 0.2 was used. It was followed by a dense layer with an output size of 3 and activation function ‘Softmax’. This final model consists of total parameters of 879,603 and trainable parameters of 879,603. For the compilation of the model, the loss function was ‘sparse_categorical_crossentropy’ and optimizer as ‘Adam’. The model was trained on 50 epochs and a batch size of 32.

A Neural Network was trained by using a Dataset which is shown in this paper. While training the Network, the time required to train and the accuracy of the model depends on the optimizer used, the number of epochs and the correct loss function.

Adam Optimizer was used in this case. Adam optimizer is a combination of Gradient descent with momentum and RMSprop optimizers. The benefits of both the optimizers are into one. The accuracy is determined by no epochs given for training. The accuracy can be increased by training with large data and more epochs. But after some particular number of epochs, the accuracy moves towards saturation and increases very slowly. This is the ideal point. The figures below show accuracy vs epochs and loss vs epoch plot for training (blue line) and testing (orange line) dataset.

After creating training of the model, it was tested on 300 unique images belonging to three different classes i.e. Covid-19, Normal, and Pneumonia with 100 images belonging to each class. The model successfully identified 96 Covid-19 images out of 100 covid x-Ray images, 91 Normal images out of 100 normal x-ray images, and 95 Pneumonia images out of 100 pneumonia x-ray images. The total accuracy of the prediction of the model is 94%. The final model has great accuracy for the prediction of the X-ray images belonging to any of the three categories i.e. Covid-19 or Normal or Pneumonia.

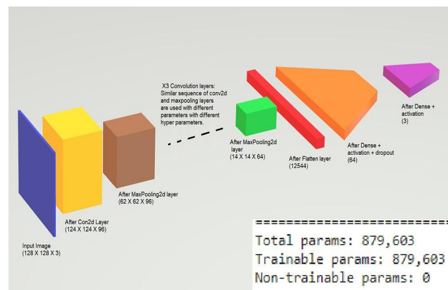

Fig 2 Proposed customized CNN architecture.

Android Application for Developed Model

After successfully creating the deep neural network model with the best prediction accuracy the model file was saved. Now for creating the android application with the trained model added to it Android Studio was used. As android only supports TensorFlow lite, so a '.tflite' file of the model (TensorFlow Lite is the lite version of TensorFlow) was created. The user interface is very simple. The ‘.xml’ file was created in a relative layout with two buttons and two text views. The buttons are ‘SELECT IMAGE’ for selecting an image from the file manager and ‘PREDICT’ for predicting the category in which the image lies (Covid-19 or Normal or Pneumonia). The first text view has text ‘Prediction’ and the other text view will print the prediction of the model if the image of x-ray belongs to either of the three categories. The ‘MainActivity.java’ file consists of the backend instruction for the application. For adding the model to the application the TensorFlow light model was imported which was created previously. So on clicking the ‘SELECT IMAGE’ button on the application, the backend code will take the user to the file manager and if the user selects any of the images then the user will be redirected to the main activity. If the user pressed the ‘PREDICT’ button then the model will give an array of [1 X 3] as an output. If the array[0] is 1 then the image is predicted to be 'Covid-19', if the array[1] is 1 the image is predicted to be 'Normal' and if the array[2] is 1 the image is predicted to be 'Pneumonia'. The prediction will be printed on the second text view.

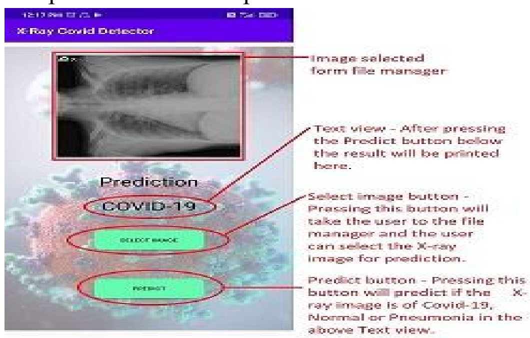

Fig 3 Android application interface

IV. Result and Discussion:

Different algorithms were used for detection on Covid-19, Normal, Pneumonia using the ResNet-34 and ResNet-50 had a training accuracy of 71.62% and 79.47% respectively, but the customized neural network model was able to provide training accuracy of 95.04% and testing accuracy of 94.14%. Below Fig.4.1 and 4.2 show the loss vs epoch graph and accuracy vs epoch graph respectively. Fig 1.8 and 1.9 show the accurate prediction provided by the android application. The user interface of the android application is kept simple for any user to use it efficiently. The model successfully identified 96 Covid-19 images, 91 Normal images, and 95 Pneumonia images from 100 images provided from each category. The accuracy achieved is the best result so far in the multi-class prediction scenario. The early prediction of Covid-19 using x-ray images may help to early diagnosis and treatment of the patients who may have been infected just using an X-ray.

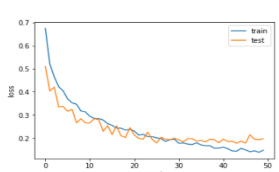

Fig 4.1 Loss Vs Epoch

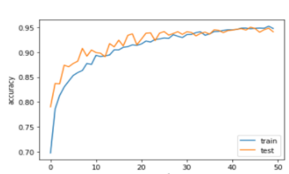

Fig 4.2 Accuracy Vs Epoch

Fig. 1.8 COVID 19 Prediction

Fig. 1.9 NORMAL Prediction

V. Conclusion

Detection of Covid 19 is not an easy process when it comes to physical tests since the chance of getting infected with this highly contagious disease is high. Since the symptoms of both covid 19 and pneumonia are almost the same (Cough, Difficulty in breathing, sneezing, fever), the patients may get confused between the two. The model gave a prediction accuracy of 94%. This means that maximum images were being classified correctly. An android application was made from this model. The reason behind doing so was that if a patient has an x-ray in a digital form one can check that if they are suffering from covid 19 or pneumonia with a 94% accuracy. In this way, the disease won't spread from patient to patient in a testing facility and from patient to doctor or any other testing personnel in the same way.

References:

[1] Boran Sekeroglu, Ilker Ozsahin "Detection of COVID-19 from Chest X-Ray Images Using Convolutional Neural Networks." DOI: September 18, 2020. https://doi.org/10.1177/2472630320958376

[2] Simonyan, K., Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv. 2015, arXiv:14091556.

[3] Kaiming, H., Xiangyu, Z., Shaoqing, R.; et al. Deep Residual Learning for Image Recognition. ArXiv. 2015, arXiv:1512.03385.

[4] Szegedy, C., Vanhoucke, V., Ioffe, S.; et al. Rethinking the Inception Architecture for Computer Vision. ArXiv. 2015, arXiv:1512.00567v3

[5] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In CVPR, 2017.

[6] Howard, A. G., Zhu, M., Chen, B.; et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv. 2015, arXiv:17040486.

[7] Jain, R., Gupta, M., Taneja, S., et al. "Deep learning-based detection and analysis of COVID-19 on chest X-ray images." Appl Intell 51, 1690–1700 (2021). https://doi.org/10.1007/s10489-020-01902-1

[8] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U. Rajendra Acharya ``Automated detection of COVID-19 cases using deep neural networks with X-ray images" DOI: April 28, 2020. 10.1016/j.compbiomed.2020.103792

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi "You Only Look Once: Unified, Real-Time Object Detection" https://arxiv.org/abs/1506.02640

[10] R. Punia, L. Kumar, M. Mujahid, and R. Rohilla, "Computer Vision and Radiology for COVID-19 Detection." 2020 International Conference for Emerging Technology (INCET), 2020, pp. 1-5, DOI: 10.1109/INCET49848.2020.9154088.

[11] "SIRM- Italian Society of Medical Radiology (SIRM)", https://www.sirm.org/category/senza-ategoria/covid-19/.