Android App Development for X-Ray Image Classification using Convolutional Neural Network

Yash G. Bambal[a], Ronit R. Shahu[a], Richa R. Khandelwal[a]
[a] *Department of Electronics Engineering, Shri Ramdeobaba College of Engineering and Management, Nagpur, 440013, India*

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.

Keywords: *pneumonia disease ; X-ray Image; convolutional neural networks; COVID19*
DOINumber:10.14704/NQ.2022.20.11.NQ66032 **NeuroQuantology 2022; 20(11): 287-293**

1.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 of a patient, it could result in a huge loss of life because 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 quantity measurement and transfer learning trials. 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. A CNN based network InceptionNet V3 is used for classification. It decreases the number of parameters and increases training speed. XceptionNet is a modification of InceptionNet. Despite the fact that it has the same parameter size as InceptionNet, it performs somewhat better. The blocks in ResNeXT are replaced with those that follow the divide, transform, and merge technique employed in the inception model. Instead of creating a model from scratch, the writers in [8] used a DarkNet 19 model that was accessible on Github. The DarkNet classifier is the foundation of the YOLO[9] real-time object identification system. They used fewer filters and layers in the beginning and later increased them as per their need.

In [10] the author discusses the use of X-ray imaging to detect Covid-19 infection, normal report, and pneumonia infectivity. COVID-19 X-Ray pictures were composed from internet data organized by European Health Care. The images of Pneumonia X-Rays were taken from an open-source collection. Following the removal of the noisy images, a dataset of 374 images for each of the three labels Covid-19 infection, normal report, and pneumonia infectivity was recovered. 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 they were able to reach an accuracy of with their model of 66.67% with an error rate of 33.33% and in the ResNet-50 model, they were able to attain an accuracyof 72.38% with an error rate of 27.62%.

Work on pre-processing the image which uses filters based on the lateral or frontal projection of X-ray is presented in [11] along with CNN. Dataset balance, medical specialists' picture interpretation, and data augmentation are among the pre-processing processes utilized in [12] to achieve higher performance.

We have applied various algorithms 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.

2.Dataset preparation

To train and validate deep neural networks or CNN models, a large number of images are required. 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. We took three different datasets from Kaggle's challenge. After gathering all the datasets, we created a new dataset 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. The validation has about 1549 images individually making it about 4647 images in total making it 43% of training images. Fig. 1 illustrates the sample of the training images.

Fig 1(a) Covid 19

Fig 1(b) Normal

Fig 1(c) 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, we do not need to worry about the resolution of images. 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.

3. Proposed Approach

The purpose of building the model is to use X-ray images to differentiate Covid-19 patients from pneumonia patients and healthy people. As a result of this work, a customized deep neural network model for detecting the Covid-19 using X-ray images has been built with the maximum accuracy possible. The entire effort is divided 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. Customized CNN (Convolutional Neural Network) is used for training and enhancing the accuracy of the deep learning model to a large extent. The given input image is classified into three categories by this classification model: Covid-19 infection, normal report, and pneumonia infectivity. The user interface of the android application is very simple to be used by new users.

3.1 Model Architecture

The proposed neural network model's design is shown in Fig. 2. Initially, the basic model was developed using three convolutional 2d layers and a 'Relu' activation function. The max-pooling 2d layer came after each convolution layer. For the first layer, the kernel size is set at (5 X 5) and the filters are set to 128. The input to the first layer was (128 X 128 X 3). For the second layer of convolution 2d, kernel size is (3 X 3) and filters = 64 and for the third convolution layer, we used kernel size of (3 X 3) 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, after that, a dense layer with an output size of 64 and the activation 'Relu' is present. 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. With this model, we were able to reach an accuracy of 88.34%. To improve the accuracy furthermore, hyperparameter tuning is used through the Keras tuner for hyperparameter tuning.

Through hyperparameter tuning training accuracy of 95.04% and validation accuracy of 94.14% are achieved. 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 trailed by a max-pooling layer with a pool size of (2, 2). Then there is a flatten layer, a dense layer with an output size of 64 and activation 'Relu' as described in the initial model. In the initial model, we used a dropout rate of 0.5 but the model was overfitting so to overcome this problem we used a dropout rate of 0.2. After that, a dense layer with an output size of 3 and the activation function 'Softmax' was added.This final model consists of total parameters of 879,603 and trainable parameters of 879,603. For the compilation of the model we used the loss function as 'sparse_categorical_crossentropy' and the optimizer as 'Adam'. The model was trained on 50 epochs and a batch size of 32.

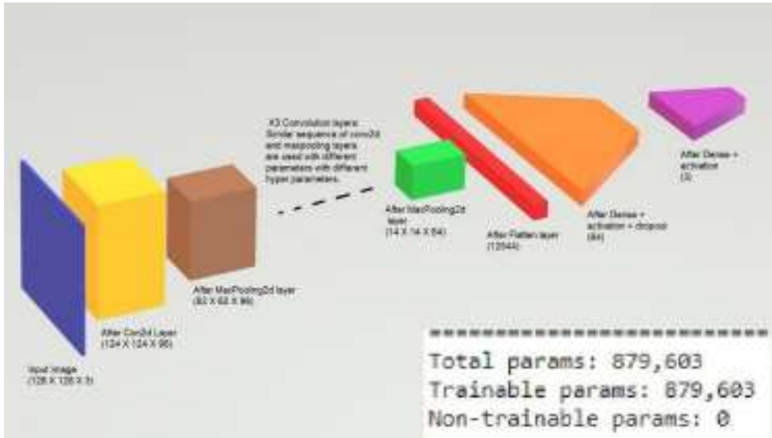

Fig. 2. The proposed architecture of customized CNN

3.2 Development of Android Application

After creating the neural network model with the best prediction accuracy model file is saved. Now for creating the android application with the trained model added to it we have used Android Studio. As android only supports TensorFlow lite, so the '.tflite' file of the model (TensorFlow Lite is the lite version of TensorFlow) is created. The user interface is kept 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 the text 'Prediction' and the other text view will print the prediction of the model if the image of the x-ray belongs to any of the three categories. The 'MainActivity.java' file consists of the backend instruction for the application. For adding the model to the application TensorFlow light model is 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.

4. Result and Discussion

4.1 Performance Analysis of Model

The data set for training and validation consists of 10725 images and 4647 images respectively, belonging to three classes Covid-19 infection, normal report, and pneumonia infectivity. Before the training of the deep neural network model, the images were scaled down to (128 X 128). Keras data generator was used for preprocessing. While training the network, the accuracy and the time required to train the model depends on the factors: optimizer used, number of epochs and correct loss function. Adam optimizer is used here which is a combination of gradient descent with momentum and RMS prop optimizers. The benefits of both the optimizers are in one. The number of epochs used for training determines the accuracy. 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. For the training (blue line) and testing (orange line) datasets, Figs. 3 and 4 depict accuracy of model vs epoch and loss function vs. epoch plots, respectively.

After creating model, training of the model was tested on 300 unique images belonging to three different classes i.e. Covid-19 infection, normal report, and pneumonia infectivity 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 infection, normal report, and pneumonia infectivity .

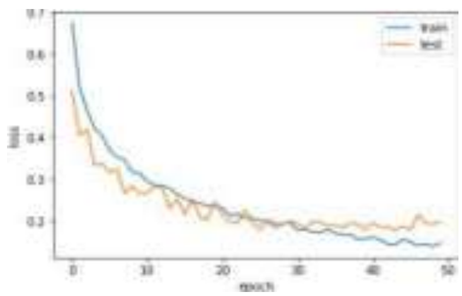

Fig. 3. Loss function vs Epoch

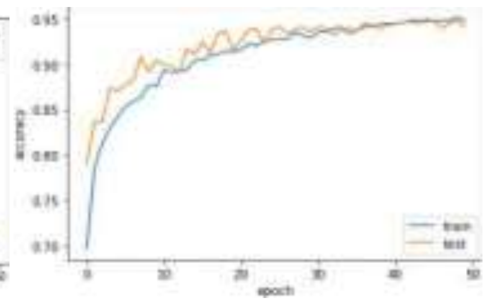

Fig. 4. Accuracy of model vs Epoch

4.2 Recital of Android App

Fig.5 shows Android Application interface, 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.

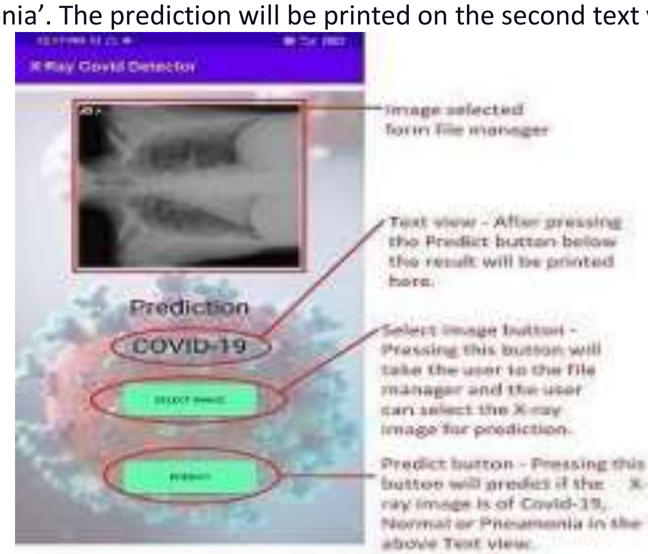

Fig.5. Android application interface

5. Conclusion

When it comes to physical examinations, detecting Covid-19 is difficult due to the significant risk of becoming infected with this highly contagious disease. Because the symptoms of Covid-19 and pneumonia are nearly the same (cough, difficulty breathing, sneezing, fever), patients may be confused. The model gave us 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 you have your chest x-ray in digital form ranging from 128 x 128 to 1024 x 1024

291

pixels you can check that you are suffering from covid 19 or pneumonia with 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] DanielArias-Garzón et al, COVID-19 detection in X-ray images using convolutional neural networks, Machine Learning with Applications, 6, 2021

[12] Aijaz Ahmad Reshi et al, An Efficient CNN Model for COVID-19 Disease Detection Based on X-Ray Image Classification, Complexity and Robustness Trade-Off for Traditional and Deep Models, special issue, Hindawi, 2021

Biography of Authors

	**Mr. Yash Bambal, Graduated from Shri Ramdeobaba College of Engineering and Management is currently employed at Persistent System Limited as Software Engineer in Salesforce Technology. His research and publication interests include Deep Learning.** ***Email: bambaly@rknec.edu***
	**Mr. Ronit Shahu, Graduated from Shri Ramdeobaba College of Engineering and Management is currently employed at Persistent System Limited as Software Engineer in Python FullStack Technology. His research and publication interests include Deep Learning.** ***Email: shahurr@rknec.edu***

	**Dr. (Mrs.) Richa R. Khandelwal, graduated from Madhav Institute of Technology and Science, Gwalior and studied her post graduation in Electronics Engineering from Yeshwantrao Chavan College of Engineering, Nagpur. She was awarded a Ph.D. from Barkatullah University Bhopal. Her employment experience includes 21 years of teaching. She has to her credit many International and National Conference/Journal papers. Her research and publication interests include Image Processing and Machine Learning.** ***Email: khandelwalrr@rknec.edu***