COVID-19 DETECTION USING X-RAY IMAGES

B.S.Murugan [1], G.Harshitha[2], A.Chandana[3] ,Ch.Bhuvaneswari[4]

[1]Assistant Professor, CSE, Kalasalingam Academy of Research and Education, Krishnan koil, Tamil Nadu, India

[2]B.Tech, CSE, Kalasalingam Academy of Research and Education, Krishnan koil, Tamil Nadu, India

[3]B.Tech, CSE, Kalasalingam Academy of Research and Education, Krishnan koil, Tamil Nadu, India

[4]B.Tech, CSE, Kalasalingam Academy of Research and Education, Krishnan koil, Tamil Nadu, India

Abstract

The coronavirus 2019(COVID-19), which first occur in wuhan city of China in December 2019, spread quickly around the world and became a plague. It is necessary to detect the positive cases as early as possible so as to prevent the further spread of this disease. Application of convolutional neural networks (CNN) techniques coupled with medical imaging can be helpful for the accurate detection of this disease. In this project a new model for automated COVID-19 detection using raw chest X-ray images is used. The model is developed to supply accurate diagnosis for binary classification (COVID vs No-Findings). Our model produced an accuracy of 98.44%. The sequential model was used to train a model using Keras and Tensorflow. We introduced convolutional layers and implemented different filtering on each layer.

Keywords— Convolutional Neural Networks, Keras, Tensor-flow, Feature Extraction, Sequential Model.

1. Introduction

1.1 Overview

COVID-19 is a transmissible disease that was caused by the Severe Acute Respiratory Syndrome Coronavirus 2(SARS-CoV-2). The disease was first discovered in china and has spread among the world. As we know, on December 12, 2020 more than 12,847384 confirmed cases of COVID-19 and 55,587 confirmed deaths due to disease. Signs of infection include breathing problem, cough, fever. In more serious cases, the infection can cause severe acute respiratory syndrome, septic shock, multi organ failure and death. There is a limited number of COVID-19 test kits available in hospital due to increasing cases daily. Because it is required to implement an automatic detection system as an alternative diagnosis to prevent COVID-19 spreading among people. The sequential model was used to train a model using Keras and Tensorflow. We implemented convolutional neural network layers and implemented different filtering on each layer.

1.2 Convolutinal Neural Networks(CNN)

The objective of this project is to enhance COVID-19 detection accuracy from chest X-ray images. Convolutional Neural Networks (CNN) is one quite deep neural network. The Convolutional neural network structure includes two layers one is feature extraction layer, the input of every layer is connected to the local receptive fields of the previous layer and extracts the local feature. Once the local feature is extracted, the positional relation between it and other features also will be determined.

1.3 Keras Model

Keras model could even be an arrangement about how we stack our neural network layers. We can stack our neural network layers mainly in three ways:

1) Sequential Model

2) Functional Model

3) Subclass Model

During this project we train a neural network using sequential model. A Sequential model is suitable for a transparent stack of layers where each layer has exactly one input tensor and one output tensor. Any of the layers has multiple inputs or multiple outputs. We created a Sequential model by passing an inventory of layers to the Sequential constructor.Once a model is made, you'll call its summary() method to display its contents.

1.4 Tensor Flow

Tensor Flow is an open source library for large-scale deep learning and numerical computation. It uses Python to give a convenient front-end API for building applications with the framework. Tensor Flow can train and run deep neural networks for digit classification, word embeddings, image recognition, sequence-to-sequence models for MT, tongue processing, recurrent neural networks. Over all, Tensor Flow supports production prediction at scale, with an equivalent models used for training.

1.5 Fundamental Steps in Image Processing

Image processing is a technique to transfer a picture into digital form and perform some operations there on it , so as to urge an enhanced image or to extract some useful information from it. It is a kind of signal exemption which input is a picture, like video frame or photograph and output can be image or characteristics associated in that image. Generally, Image Processing system contains treating images as two dimensional signals while applying already set signal processing methods to them.

The usage of image processing is split into 5 groups.

- Visualization - Observe the objects that aren't visible.

- Image sharpening and restoration - to form a much better image.

- Image retrieval and search for image of interest.

- Measuring the pattern – Measures various objects in a picture.

- Image Recognition – Distinguish the objects during a picture.

1.6 Feature Extraction

Feature extraction contains reducing the amount of variables required to explain an outsized set of knowledge. When performing analysis of complex data among the main problems from the amount of variables involved. Analysis with an outsized number of variables generally requires an outsized amount of memory and computation power, also it's going to cause a classification algorithm to overfit to training samples and generalize poorly to new samples.

2. Related Works

Liu, S., Satapathy, S.C., Zhang, YD. et al [1] proposed a deep learning model that can treat COVID-19 on chest X-ray images more accurately. The accuracy of 93.64% is achieved in identifying COVID-19.

Ashad, Chowdhury, Md. Muhtadir & Kabir, Nihad & Rahman [2] proposed a parallel-dilated CNN based COVID-19 detection system from chest x-ray images, named as Parallel-Dilated COVIDNet (PDCOVIDNet). The accuracy, precision, recall and F1 scores reach 96.58%, 96.58%, 96.59% and 96.58%, respectively, which is comparable or enhanced compared with the state-of-the-art methods.

Choi, W., Jeon, E., Kim, J., Lee, K. [3] proposed a model on transfer learning with a deep convolutional neural network (CNN)-based COVID-19 screening to identify efficient transfer learning strategies. The accuracy value of this experiment is 95% concerning COVID-19 classification.

Boran Sekeroglu1 and Ilker Ozsahi2 [4] proposed a deep learning for the detection of COVID-19 using chest X-ray images. Totally, 1583 healthy, 4292 pneumonia, and 225 confirmed COVID-19 X-ray images were used in their experiment, which involved the training of deep learning and machine learning classifiers. It uses minimized layers without preprocessing which is capable of detecting COVID-19 in a limited number of images

3. System Analysis

3.1 Existing System

The following are the methods used in the existing system:

- Swab Test – It is a test in which a special swab is used to take sample from your nose and throat.

- Nasal aspirate – Here, a saline solution will be injected into your nose and then a sample is taken with a light suction.

- Tracheal aspirate – In this case, bronchoscope which is a thin tube with a torch is put into your mouth to reach your lungs from where a sample is collected.

- Sputum Test – Sputum is a thick mucus that gets

accumulated in the lungs and comes out with a cough that will be collected in a special cup or a swab to test.

3.2 Proposed System

We have proposed an automatic prediction of covid-19 using a deep convolutional neural networks and Chest X-ray images. For this implementation, python programming language used with Tensor Flow Keras and OpenCV modules. We have deployed the previously trained CNN model in a web application using a python backend with a Flask web development framework. HTML and JavaScript are used as the frontend of the website. The proposed models will have end-to-end structure without manual feature extraction and selection methods. Chest X-ray is the best tool for the detection of covid-19.

3.3 Required Analysis and specification

The framework is used for the prediction of COVID-19 where the data of X-ray images is given to the code, preprocesses and trains the CNN model.

3.4 Hardware Specifications

Minimum hardware requirements are

- Processor: intel i5
- Hard drive: 500 GB
- RAM: 4 GB

3.5 Software Specifications

Minimum software requirements are

- Operating systems: Linux, windows, mac…
- Frontend: HTML and JavaScript
- Backend: Python 3.6 and Flask 1.0.2
- Anaconda software
- Jupyter Notebook API
- Keras 2.3.1
- Pillow 5.4.1
- Numpy
- Scikit-learn 0.20.0
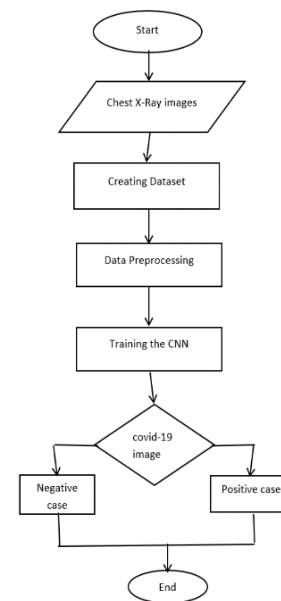- Tensor-flow 2.2.0
- Matplot lib 3.0.2

4. System Design

4.1 Basic System Design

A basic system design represents the simplistic representation of the design of the project. It shows the processes involved in the sequential model in CNN that were followed in the project.

4.1.1 Data Flow Diagram

Figure 4.1.1 Dataflow diagram

The above figure represents the data flow diagram. Data flow diagram represents the flow of data from the start of the process to the end of the process and finally returns to the user.

4.2 Design Constraints and Standards

Social Impact

The constraint of the project can be namely the image clarity. As the data set of the data set of the images can be blurry and undistinct sometimes may cause the testing yield the different results. However this problem can be compromised by getting the images suitable for testing and training the data. We should not reveal the patient names of the dataset collected. Time and cost of the people will be reduced by this system.

Design Standards

This project is based on IEEE 3333.2.1-2015 –Medical modeling and visualization. This standard focuses on the project demands arising when scientific results in the field of medical visualization are applied for the construction of a system. It is targeted to aid the clinical work of medical professionals. This standard includes visualization techniques by the automated medical shape detection and reconstruction of three-dimensional (3D) models from two-dimensional (2D) medical images.

5. Module Description

Convolutional Neural Network (CNN) might seem intimidating for a beginner. However, this project will provide an overview of how to build a model to detect COVID-19 using Tensorflow and Keras.

The content of the project is listed as follows:

- Creating Dataset
- Data Preprocessing
- Training the CNN
- Webpage Creation

5.1 Creating Dataset

This experiment leveraging the data from Covid Chest X-Ray Dataset and Pneumonia dataset by Praveen from GitHub and Kaggle respectively. Using these two datasets, we have created a customized dataset as COVID19 Negative and COVID19 Positive. The two classes of dataset can be seen in the figure below:
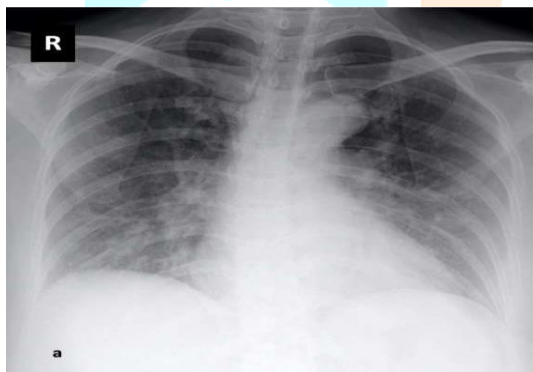

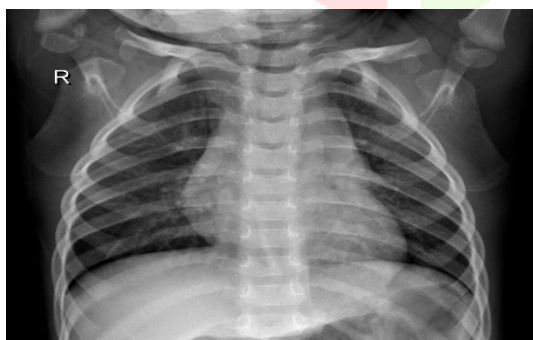**Figure 5.1** i) COVID-19 Positive X-ray Image

Figure 5.1 ii) COVID-19 Negative X-ray Image

5.2 Data Preprocessing

- In this process, we are using two categories as 0 for COVID-19 Negative and 1 for COVID-19 Positive shown below:

  ['Covid19 Negative', 'Covid19 Positive']

  [0, 1]

- Let us set the original image set to be S_1, which is composed of n images as

  $S_1=\{s_1(1), s_1(2),\ldots, s_1(i),\ldots\}$

- First, we compress the three channel color image to gray image, and get the grayscale image set S_2 as

  $S_2=G(S_1|BGR \rightarrow Grayscale)$

  $=\{s_2(1), s_2(2),\ldots, s_2(i),\ldots\}$

- Resizing the gray scale into [100 100], since we need a fixed common size for all the images in the dataset and then appending the image and the label(categorized) into the list (dataset).

- Now, it will be normalized and saved as data and target.

5.3 Training the CNN

Before we thought of training the CNN, we will load the data and target.

5.3.1 Model Development

The following are the several components that our model comprises of:

- **tf.keras.layers.Conv2D():** The convolution layer which improves image recognition by isolate images features.
- **tf.keras.layers.MaxPooling2D():** a layer to reduce the information in an image while maintaining features.
- **tf.keras.layers.Flatten():** flatten the result into 1-dimensional array.
- **tf.keras.layers.Dense():** add densely connected layer.

Here, Conv2D() and MaxPooling2D() will stack together as one layer. Then, the output of the final convolutional layer will be flattened and fit to fully connected neurons. In addition to this layer, dropout, activation and concatenate layers are used. In addition to this, before the model is fitted for training, it is necessary to configure the specifications as follows:

- loss: Covid-19 detection is using sigmoid activation in the final step, which resulted in either 0 or 1 (covid-19 positive or covid-19 negative). Therefore, categorical_cross entropy is the most suitable loss function to the training that measure the loss between the probability of the class predicted

from the soft max activation function and through probability of the category.

- optimizer: Adam optimizer to train 15 epochs for each model with a learning rate of 0.001 will be used.
- metrics: Accuracy is the metric which is used for measurement to obtain the prediction on each epoch.

The following is the overview of the model that we have built:

Convolutional Neural Network Architecture
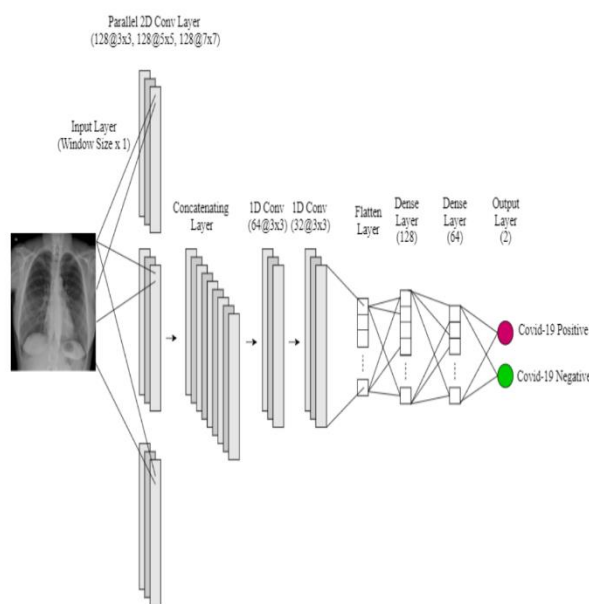
Figure 5.3.1 ii) Convolutional Neural Network Architecture

5.3.2   Model Training

Before training the model, we will set up data generators to read images from source folders instead of labeling the image one by one. Basically, ImageDataGenerator will label the images based on the directory the image is contained. It points to the sub-directory of the data.

- Grayscale normalization should be done in order for CNN to converge faster by using rescale parameter. The normalization transforms the image pixel from [0…255] to [0…1].
- Create two generators namely **train_data** and **train_target**. Where train_data points to the sub-directory of train data and train_target points to the sub-directory of evaluation data.
- Then, train the model for 15 epochs with 15 steps on every epoch.
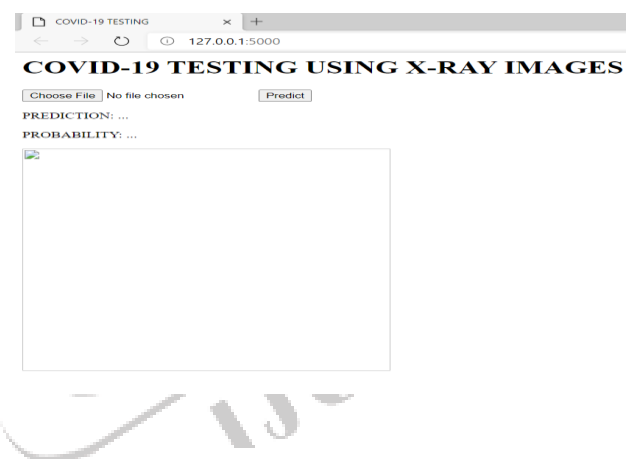
5.3.3   Model Evaluation

After the model is trained completely, the test dataset will be used to predict the results. This model evaluation step is useful to measure how well the trained model predicts unseen data. The accuracy of the trained model reaches 98.44% as below:

Figure 5.3.3 Loss and Accuracy of the model

5.4 Webpage Creation

The deployment of the previously trained CNN model have done in a web application using a python back end with a flask web development frame work. The front end of the website is created with HTML and Java script.

Figure 5.4: Webpage

6.   Evaluation and Results

6.1 Evaluation Method

The evaluation was carried out in the progress of knowing the performance of the proposed covid-19 detection process. The sequential model used in this project can identify two classes of covid-19. This can be done in several layers and the output contain two categories as 0 and 1.

| Category	Output
0	COVID-19 Negative
1	COVID-19 Positive

Table 6.1 Categories of COVID-19

6.2 Final Prediction

At this stage, the model is ready to make a prediction in this process the front end and back end codes of this project allows the users to input an image and to determine whether the X-ray image is diagnosed with covid-19. It will give the prediction and probability when the user clicks the predict button after importing the input as X-ray image.

The following are the two prediction of this project:
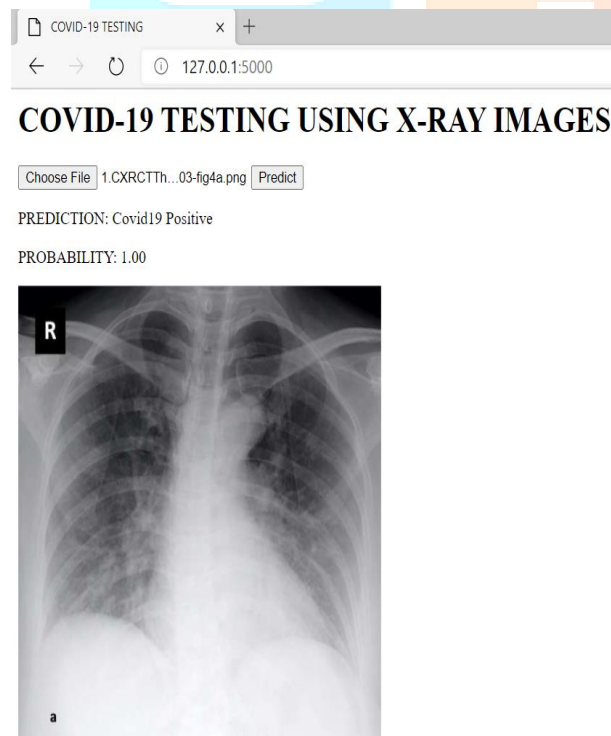
Figure 6.2.1 i) Prediction of COVID-19 Positive in Webpage
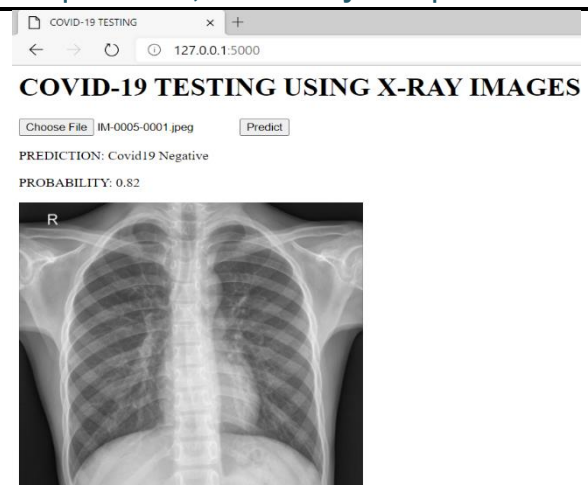
Figure 6.2.1 ii) Prediction of COVID-19 Negative in Webpage

7. Conclusion

In this project, we proposed a deep learning CNN model called sequential model for detecting corona virus disease(covid-19) from chest X-ray images. As we have seen, this model can effectively capture covid-19 features in the parallel layers of convolutional network, so it has an excellent performance compared to some well-known CNN architecture. Our experimental evaluation clearly shows that this model gives an accuracy of 98.84%. Using our proposed model, user can get the results of COVID-19 through webpage instantly. Therefore, this model can help to stop the spread of the pandemic with less cost and time.

8. References

1. Liu, S., Satapathy, S.C., Zhang, YD. et al (2020). A five-layer deep convolutional neural network with stochastic pooling for chest CT-based COVID-19 diagnosis. Machine Vision and Applications 32, Article number: 14-2021.

2. Ashad, Chowdhury, Md. Muhtadir & Kabir, Nihad & Rahman (2020). PDCOVIDNet: A Parallel-Dilated Convolutional Neural Network Architecture for Detecting COVID-19 from Chest X-Ray Images.

3. Choi, W., Jeon, E., Kim, J., Lee, K. (2020). Evaluation of Scalability and Degree of Fine-Tuning of Deep Convolutional Neural Networks for COVID-19 Screening on Chest X-ray Images Using Explainable Deep-Learning Algorithm. J. Pers. Med. 10, 213.

4. Boran Sekeroglu1 and Ilker Ozsahi2 (2020). Detection of COVID-19 from Chest X-Ray Images

Using Convolutional Neural Networks SLAS TECHNOLOGY: Translating Life Sciences Innovation Volume 25, Issue 6.

5. Arman Haghanifar, Mahdiyar Molahasani Majdabadi, Younhee Choi (2020). COVID-CXNet: Detecting COVID-19 in Frontal Chest X-ray Images using Deep Learning on 16 Jun 2020.

6. Cesar, P., Soares, Lucas, P., Soares (2020). Automatic Detection of COVID-19 Cases on X-ray images Using Convolutional Neural Networks on on 2 Jul 2020.

7. Yi Zhong (2020). Using Deep Convolutional Neural Networks to Diagnose COVID-19 From Chest X-Ray Images on 19 Jul 2020.

8. Chaimae Ouchicha, Mohammed Meknassi, Ouafae Ammor (2020). CVDNet: A novel deep learning architecture for detection of coronavirus (Covid-19) from chest x-ray images, Chaos, Solitons & Fractals, Volume 140, 110245, ISSN 0960-0779.