The chest x-ray data has been collected from 2 two sources: GitHub and Kaggle. The GitHub Dataset has various kind of x-rays but and a metadata file giving information about the patients’ name, age, sex, diagnosis and various distinct features. The Kaggle Dataset consists of chest x-ray images with the diagnosis of pneumonia or no infection. Both these datasets are downloaded and uploaded into a Jupyter Notebook. For the GitHub dataset, the data is filtered, considering only the patients that have been tested positive for COVID-19 and having Post Anterior (PA) View. The total number of COVID-19 chest x-ray images is 206. For the Kaggle Dataset, only normal chest x-rays are chosen, which have no infections. In order to keep the dataset balanced, we choose 206 Normal Chest X-rays. The dataset is further divided into train and validation having 80:20 proportion.

|  |  |  |  |
| --- | --- | --- | --- |
| Images | COVID-19 | Normal | Total |
| Train | 165 | 165 | 330 |
| Validation | 41 | 41 | 82 |
| Total | 206 | 206 | 412 |

The dataset contains images of COVID-19 and Normal Chest X-rays. A CNN model is to be built having two classes in order to distinguish between the Chest X-rays of patients infected with COVID-19 and Normal Chest X-rays. Here are sample Chest X-ray images of COVID-19 and Normal patients respectively.

(COVID-19 & Normal X-rays)

In order to create the CNN model, Google Colaboratory, or Colab, is used in which Python codes can be written and executed on the browser. The benefits of Colab are no configuration required, free access to GPUs and the file can be easily shared with others. Image datasets can be easily uploaded into Google Colab with a few lines of code which is executed on the power of Google hardware (including GPUs and TPUs), regardless of the power of your machine. Google Colab is widely used for getting started on TensorFlow and making neural network models.

The dataset is converted into a zip file to take up less space and is then uploaded onto Dropbox. The Dropbox link is shared on Google Colab, which quickly downloads the zip file. We unzip the file, and the dataset has successfully been uploaded into Google Colab. In order to build the model, Numpy, matplotlib, Keras in-built libraries are used. TensorFlow is running in the backend.

The CNN model consists of two types of layers: Convolutional layers and Fully Connected Layers. A nine layered CNN model is created, consisting of three sets of stacked convolutional and pooling layers followed by two fully connected layers for classifying COVID-19 and normal x-ray images. There are four convolutional layers with filter sizes of 3 x 3 but having increasing filter numbers (32, 64,64,128) over the layers.

(MODEL IMAGE)

The initial layers are small in the beginning because the lower layers detect features in small parts of the images and can find small patterns in the image as we go deeper into the layers, the receptive field of the CNN layer increases. The kernel size is 3 x 3, which is a standard choice. Activation of Relu layer is used in the convolutional layers for non-linearity. Since this is the first layer, we specify the input size as 224 x 224 x 3. There are three pooling layers with filter size 2 x 2 each which is the default size, by using Max Pooling, the receptive field of the layer increases.

The first convolutional layer is reshaped into 224 x 224 with three channels because the x-rays are RGB images. On carefully noticing the images, it is observed that the chest x-rays are not greyscaled images but RGB in nature because some pictures have a tone of blue or yellow. In the first set of a stacked convolutional layer, two convolutional layers of 3 x 3 kernel size have been used instead of one convolutional layer of 5 x 5 kernel size. Using two layers is advantageous to the model as it increases non-linearity in the model, and there are fewer parameters which in-turn reduce overfitting. The model can detect a higher level of features in images with the model layers deepening. After every pooling layer, a dropout layer is added to reduce the risk of overfitting with the dropout rate of 25%.

After inputting the convolutional and pooling layers, the output shape has to be changed in order to go forward with the fully connected layers. Hence, the model is converted from a two-dimensional layer to one dimension by using a flattening layer and then connected to a fully connected layer. The fully connected layer uses the ReLu function for activation. For the output layer, there is only one filter applied since we have to classify the images between COVID-19 and Normal chest x-ray images. Due to this reason, the activation function of Sigmoid is used.

The model is compiled with binary entropy loss and adam optimiser, which is the default optimiser function using accuracy metrics. This model uses 56 lakh parameters in total, which is trained from scratch. The input shape at the beginning is 224 x 224, which converts into 222 x 222 after the first convolutional layer. The shape is changed to 26 x 26 or 26 x 128 before the flattening layer.

The in-built Keras Image Generator library is used to train the dataset. In order to aid data convergence, the dataset is rescaled for normalisation of RGB images by 1/255. Moreover, usage of sheer and zoom augmentation is inculcated, allowing the model to take random crops from images and zooming into the images with 20% magnitude of the image. Vertical flipping of the image has been restricted in order to get accurate results. For the validation dataset, the Image Generator library is used to rescale the images for normalisation by 1/255.

For the training dataset, the flow from directory function is applied to reshape the image with a target size of 224 x 224, having a batch size of 32 and using binary classification to distinguish between COVID-19 and Normal Chest x-rays. The same has been done for the validation dataset. The input size of 224 x 224 is a standard choice used by data scientists. Most of the ImageNet problems are solved using this input size. The model would be difficult to train if the input size would be big, and it would be challenging to capture fine-grained features if the input size would be small. In the training process, 10 epochs are used with 8 steps per epoch.

This model is simpler than other models like VGG16, ImageNet and TransferNet as it uses few parameters. The other models will not give good results since they contain over a million parameters. Since the dataset contains chest x-rays, the models would have to be carefully fine-tuned with several modifications which Moreover, the dataset is too small and would hence lead to overfitting. The other models require at least 3000 images for training in order to give a good accuracy score.