CNN ON CT SCAN IMAGES DATASET
19th july 2020
Guide: Dr UTHAYAKUMAR GS

This PDF clearly explains what has been done related to the Jupyter notebook for Image classification on covid data set.

DATA SET

The data set has two folders : “CT_COVID” and “CT_NonCOVID”. The first folder contains 349 images of lungs containing Corona disease. The second folder has 397 images of normal lungs.
The data is downloaded from github repository online.

RESOURCES

The entire program is written using the Python language. We have used interactive python shell called Jupyter Notebook.
The libraries used are Tensorflow and keras for ML, cv2 for image analysis and numpy for array conversion(linear algebra).

STEPS AND EXPLANATION

The concept of Convolutional Nueral Network has been used in order to classify the given images as Covid or Non-Covid.

Step 1: We first concatenate both the directories containing covid and non covid images of the data set into one single variable.

Step 2: We then label the images containing covid as 1 and the ones without covid as 0. Then we concatenate it into variable called data_target.

Step 3: We now read every image in both the folders using cv2(tool for computer vision). We resize the image into 32 x 32 size. We then identify the r g b pattern of the image so that we can convert the image into an array. We then append all the arrays(of images) into a list.

Step 4:We now will squeeze the list and then we will normalize the data.

Step 5:We then map the arrays to the labels.

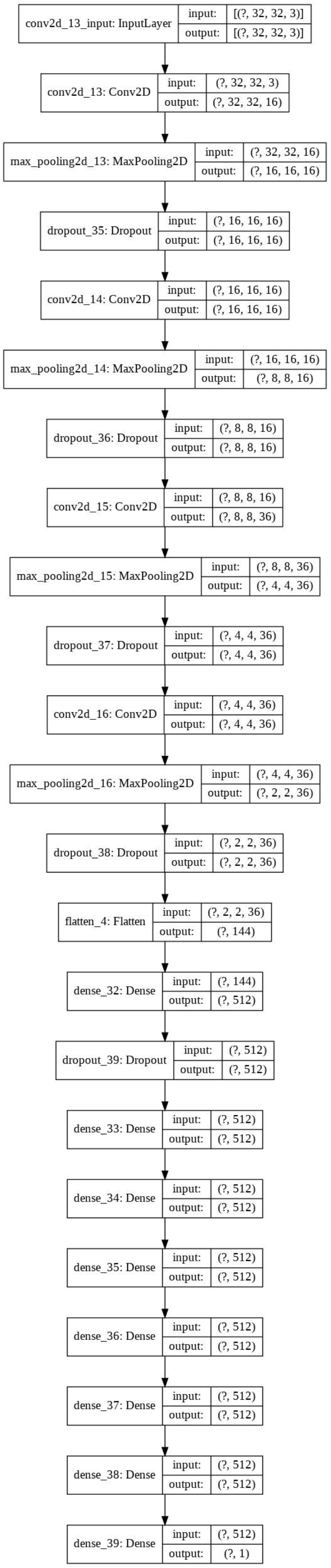
Step 6: We then split the data into training and testing data (80 percent training and 20 percent testing). We also randomize the splitting.

Step 7: We finally create the model as below: The ? in the image will be filled by the input data .

Conv2D: 2D convolution layer (e.g. spatial convolution over images).

Maxpool: Max pooling operation for 2D spatial data.

Dropout: In order to reduce overfitting.

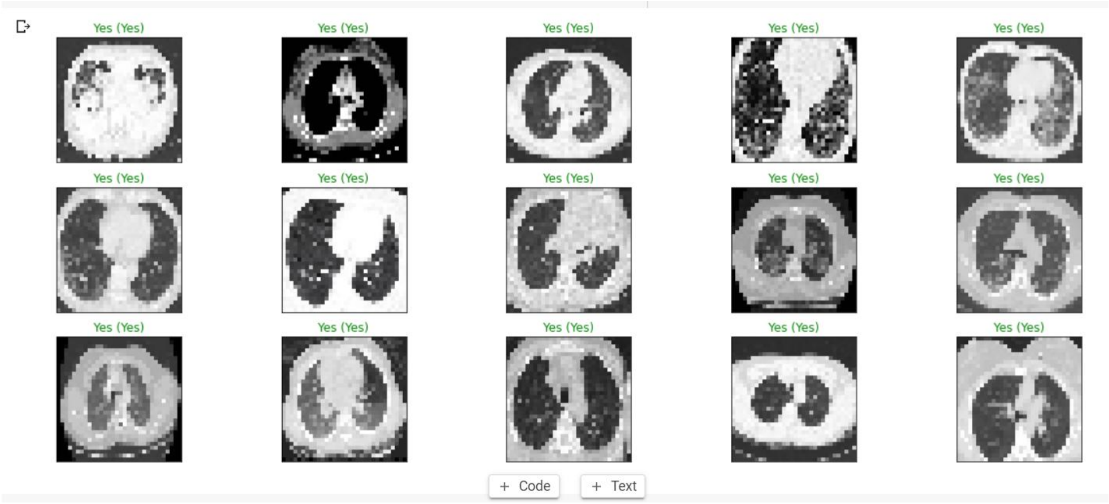

conv2d_13_input: InputLayer | input: | [(?, 32, 32, 3)] | output: | [(?, 32, 32, 3)]
conv2d_13: Conv2D | input: | (?, 32, 32, 3) | output: | (?, 32, 32, 16)
max_pooling2d_13: MaxPooling2D | input: | (?, 32, 32, 16) | output: | (?, 16, 16, 16)
dropout_35: Dropout | input: | (?, 16, 16, 16) | output: | (?, 16, 16, 16)
conv2d_14: Conv2D | input: | (?, 16, 16, 16) | output: | (?, 16, 16, 16)
max_pooling2d_14: MaxPooling2D | input: | (?, 16, 16, 16) | output: | (?, 8, 8, 16)
dropout_36: Dropout | input: | (?, 8, 8, 16) | output: | (?, 8, 8, 16)
conv2d_15: Conv2D | input: | (?, 8, 8, 16) | output: | (?, 8, 8, 36)
max_pooling2d_15: MaxPooling2D | input: | (?, 8, 8, 36) | output: | (?, 4, 4, 36)
dropout_37: Dropout | input: | (?, 4, 4, 36) | output: | (?, 4, 4, 36)
conv2d_16: Conv2D | input: | (?, 4, 4, 36) | output: | (?, 4, 4, 36)
max_pooling2d_16: MaxPooling2D | input: | (?, 4, 4, 36) | output: | (?, 2, 2, 36)
dropout_38: Dropout | input: | (?, 2, 2, 36) | output: | (?, 2, 2, 36)
flatten_4: Flatten | input: | (?, 2, 2, 36) | output: | (?, 144)
dense_32: Dense | input: | (?, 144) | output: | (?, 512)
dropout_39: Dropout | input: | (?, 512) | output: | (?, 512)
dense_33: Dense | input: | (?, 512) | output: | (?, 512)
dense_34: Dense | input: | (?, 512) | output: | (?, 512)
dense_35: Dense | input: | (?, 512) | output: | (?, 512)
dense_36: Dense | input: | (?, 512) | output: | (?, 512)
dense_37: Dense | input: | (?, 512) | output: | (?, 512)
dense_38: Dense | input: | (?, 512) | output: | (?, 512)
dense_39: Dense | input: | (?, 512) | output: | (?, 1)

The dense layers uses relu activation.
The last dense layer uses sigmoid activation.

Step 8: We then specify that we use Adam optimizer and we also use binary crossentropy for loss detection.

Step 9: We then fit the model with 1000 epochs and 128 batch size.

Step 10: We see the accuracy and then we randomly take 10 images out of testing images and see if predicted is same original images nature of covid or not.



RESULTS

The accuracy of test data varies from 75 to 82 percent. This variance is due to the factor that each time training is done randomly in each epoch.

SOLUTION

The solution is to increase the data set of images to a minimum total of 50,000 images so that we can reduce the variation.