**MLOps Task**

**Face Detection using MobileNet**

This is a program for Face Detection.

In this I have used a pre-trained MobileNet model.

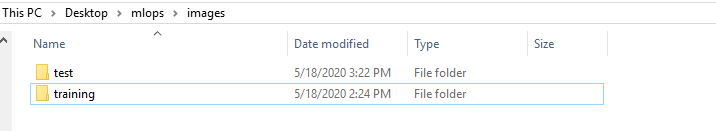
I used this code to detect some cricketers by their image.

We will show our model, the images of cricketers and then, our model will predict their name.

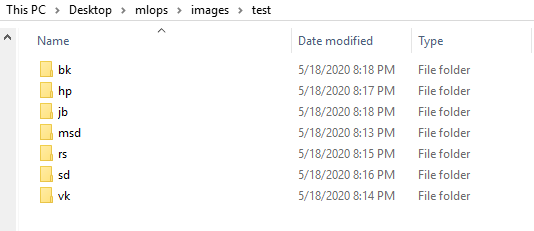
We have collected images from google and then classify those images in two folders test folder and training folder. Training folder contains images to train our model and test folder contains images to test our model. Then we have also made 7 classes in these training and test folders to denote images of 7 cricketers.

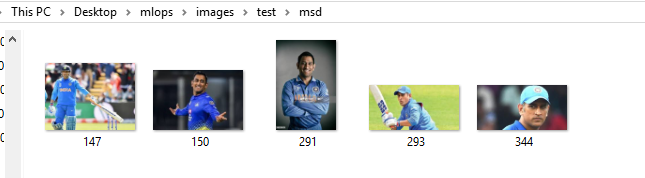
Folder structure is as shown:

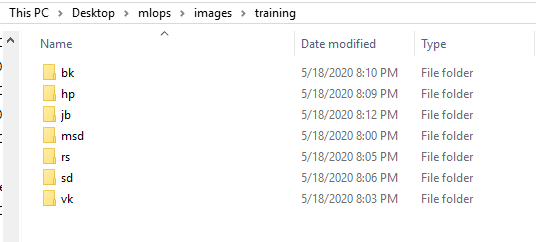
Image folder:



Test folder inside image folder containing 7 folders for different cricketer images like msd for MS Dhoni, vk for Virat Kohli, rs for Rohit Sharma, sd for Shikhar Dhawan, etc.

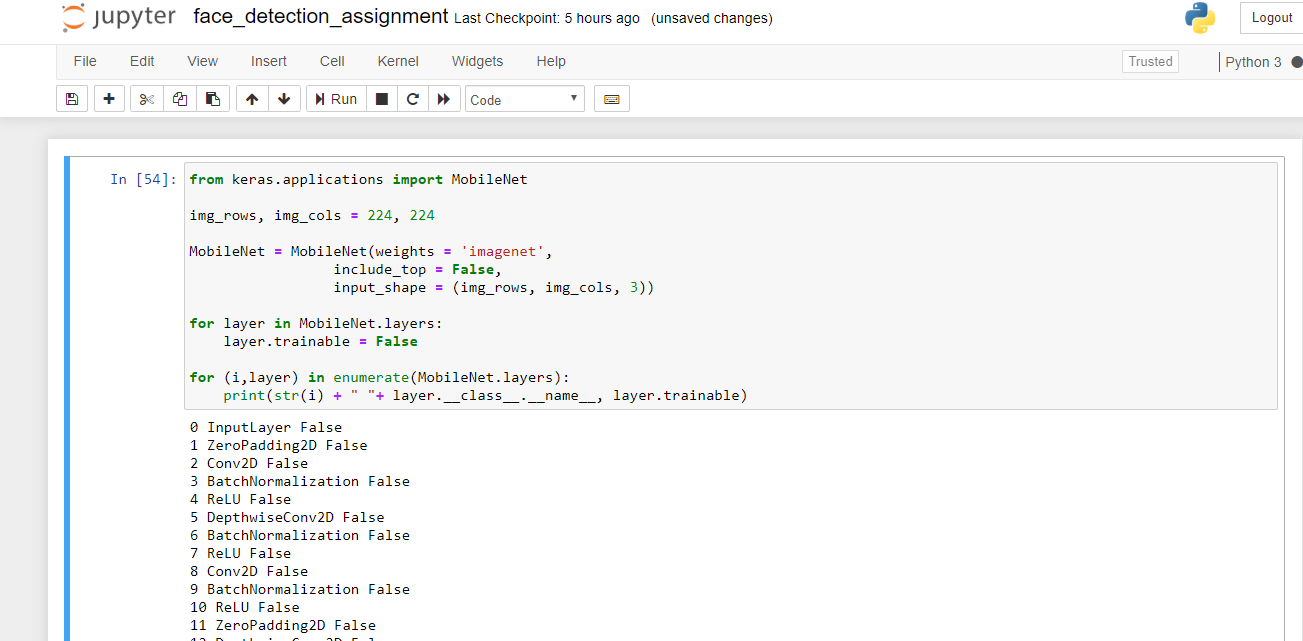


Msd folder containing imaged of MS Dhoni for test dataset.

Training dataset folder inside image folder containing different folders for different cricketer images similar to test dataset folder.

Msd folder inside training dataset folder containing images for MS Dhoni for training our model

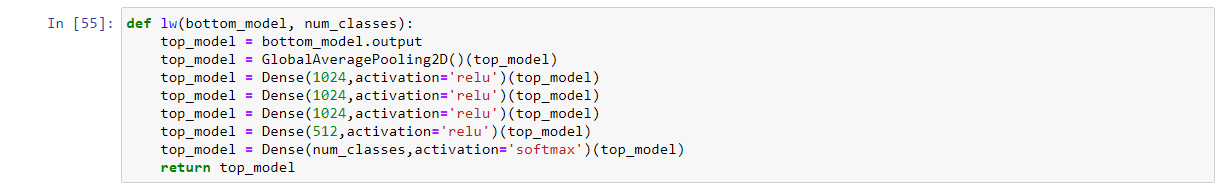




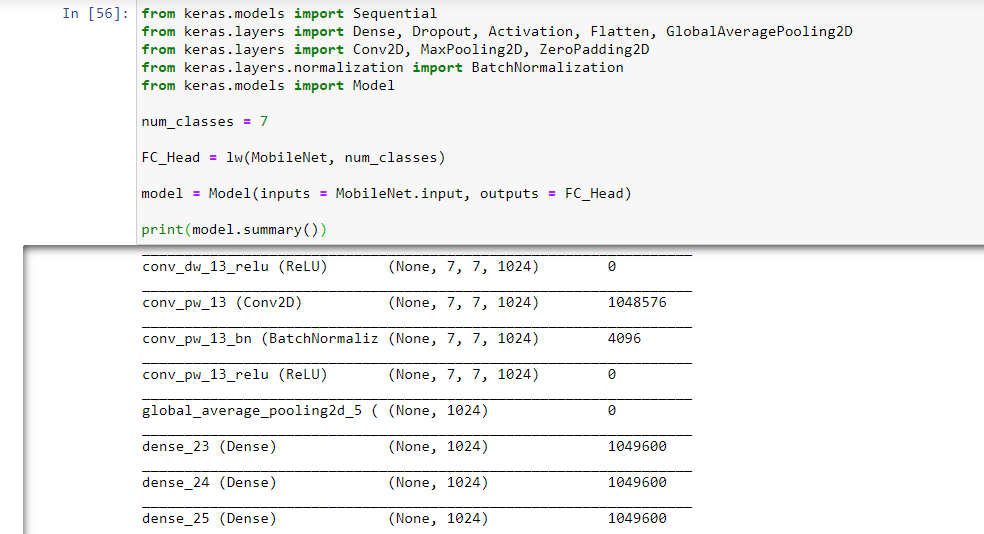
Here we have define the size of image as 224 X 224 which will be provided as input to our model.

We used MobileNet model. We have downloaded the predefined weights from imagenet and also we have not included the topmost or end layer of this model as we have written “include\_top=False”.

Then we have freezed all the layers of the model by setting trainable flag of all the layers as False.

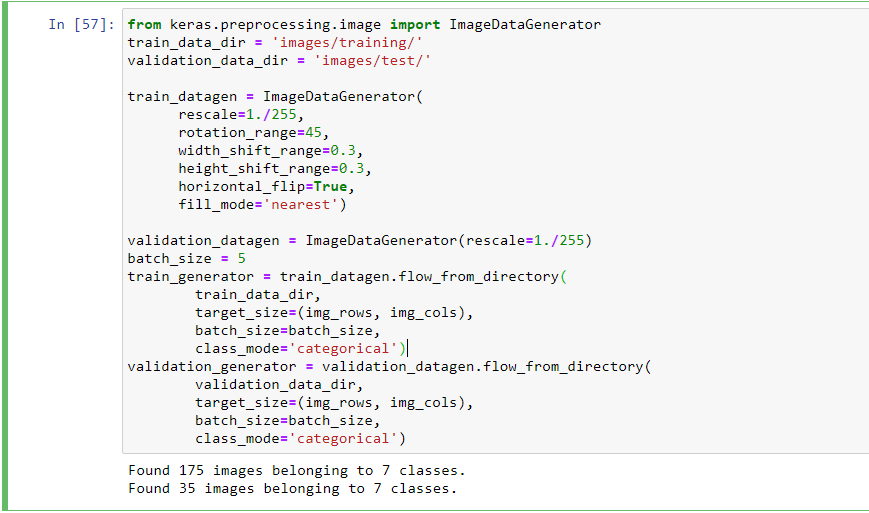


Then here we have created a function lw() and defined our output layer. We have used a Pooling2D layer and 5 Dense layers in which 4 have ‘relu’ as activation function and last layer has ‘softmax’ as the activation function.



Then we have imported the required libraries from the keras.layers and keras.model libraries.

Since I have taken 7 classes of objects, i.e. I have trained my model with pictures of 7 cricketers so we have assigned num\_classes as 7 which is passed in lw() method which is used here to create a Fully Connected layer ‘FC\_Head’. Then we have added this layer as the output layer to our model.



Then we have used ImageDataGenerator for further processing.

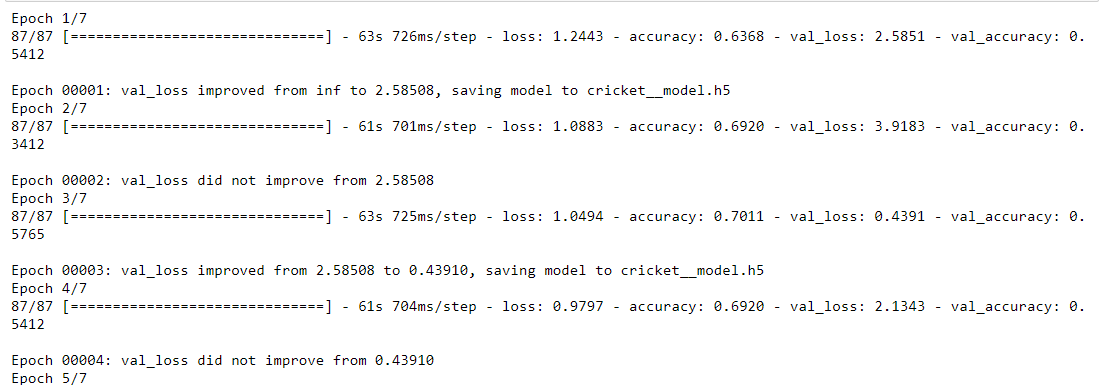
We have provided the test data and the training data. We have provided 175 images to train of 7 calsses and 35 images of 7 classes to test.

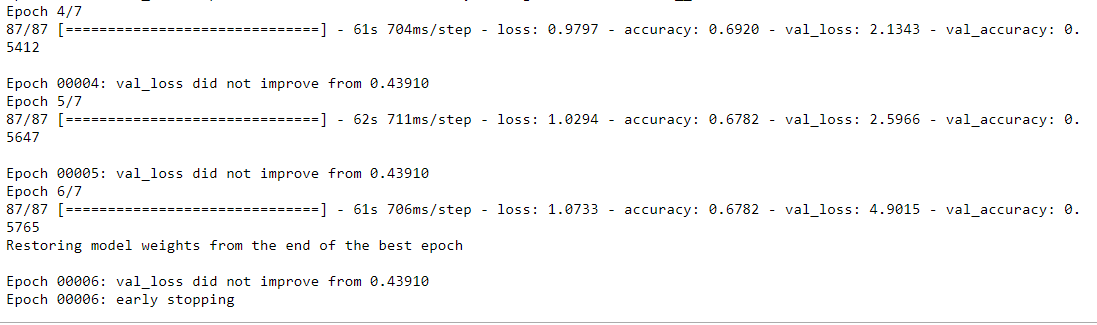


Here we have used RMSprop, ModelCheckpoint,EarlyStopping.We have saved our model named as ‘cricket\_\_model.h5’.

We have taken batch\_size equal to 2 and epochs equal to 7 after performing some hit and trials.

Following is the output when we will run this block

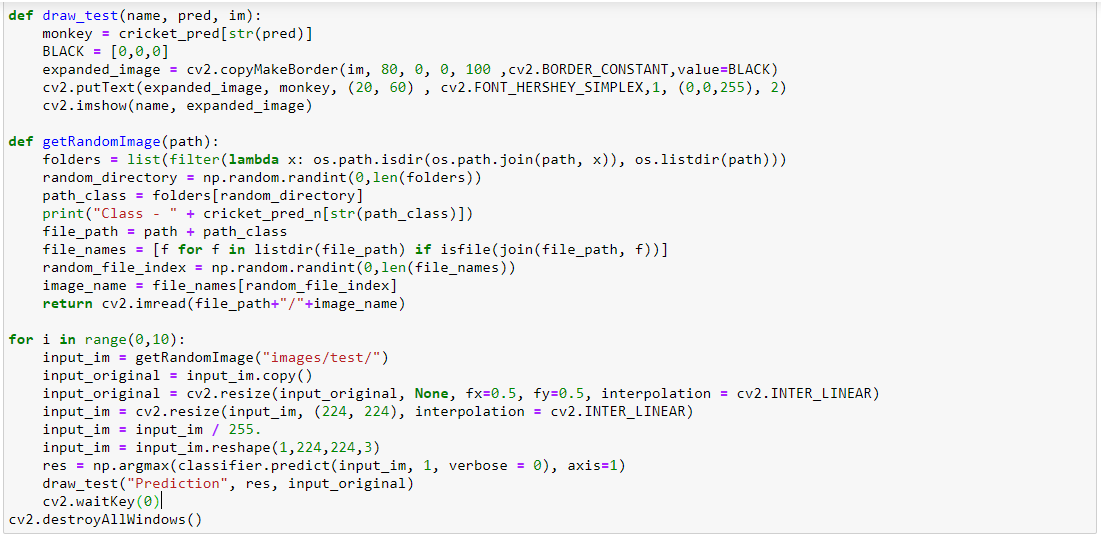


Here we get accuracy of almost 68 percent.



Here, we used load\_model to load our model ‘cricket\_\_model.h5’ which we had saved in last step



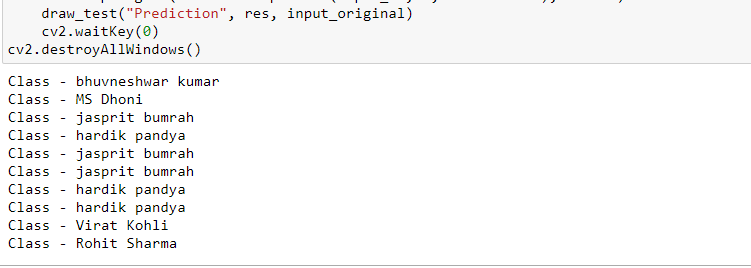


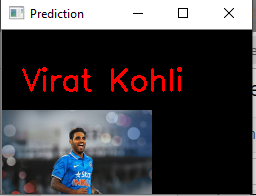
Here in cricket\_pred and in cricket\_pred\_n we have associated the required values with their classes.

Then we have made a function getRandomImage to make a random choice and selecting images from required path.

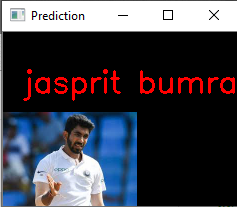
Then we have used cv2 to show our images and their respective predictions done by our model.

Following is the random values selected by our code. Then

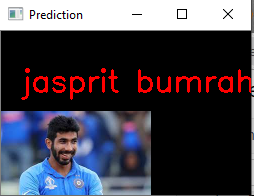


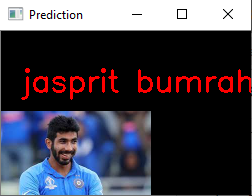
 **INCORRECT**

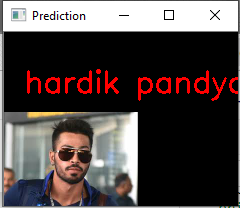
 **CORRECT**

 **CORRECT**

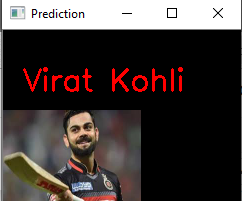
 **INCORRECT**

 **CORRECT**

 **CORRECT**

 **CORRECT**

 **INCORRECT**

 **CORRECT**

 **CORRECT**

**7 out of 10 are correct.**

This again shows that our model is **almost 70 percent accurate**.

And also we still require some further efforts to make our model more accurate.