Labellerr

Problem Statement :

Detecting the Face Attributes for a given image.

Data set :

https://www.kaggle.com/jessicali9530/celeba-dataset

SOLUTION :

- As a end to end ML cycle deployment, The Model is created and hosted in the Docker as.
- The dataset is vizualized and preprocessed before feeding into the data.
- Data set Generation
- Model Creation
- Validation
- Docker image creation
- Hosting on Local

Exploratory Data Analysis :

Before feeding data to the model some of the parameter should be consider,Data is balanced or not, Data is normalized ,whether the data contain Valid data, it should not contain NaN values.

By using the Pandas Frame work the data is checked whether its valid or not.

Then the column featues are made into categorical content with 1 and 0 and converted to integer.,using the labelencoder

Then by using the Visualization tool sns the data is viewed and then the Feature and Output Label are decided

Model Creation and Building

Hence the main target is to find the face attributes of the given images, for example a image of a girl who is wearing a glass and looking good with huge lips.

Here the features are huge lips ,wearing a glass and attractive like wise an image can contain N features , in our data set there are 40 features are given.

For handling this kind ofproblem i went to MultiLabel classification Model,Which can give the output features for the given image.

Packages Used :

Frame work : KERAS
Packages : Sklearn,Pandas,Numpy,Pillow

The model is a sequential model where the input image of size 100 X 100 is feed into network and the Output dense layer consist of 40 Output neurons as our output feature.

The Dataset is splited into 3 parts:

Train_dataset [70% of dataset]
Validation_dataset [20% of dataset]
Test_dataset [10% of dataset]

The dataset pipe line is handled by keras Data generator module

Model is build with help of keras layer function by adding sequential layer.

The Optimizer used is RMSprop and the loss function is Binary_Crossentrop function is used to train the model.

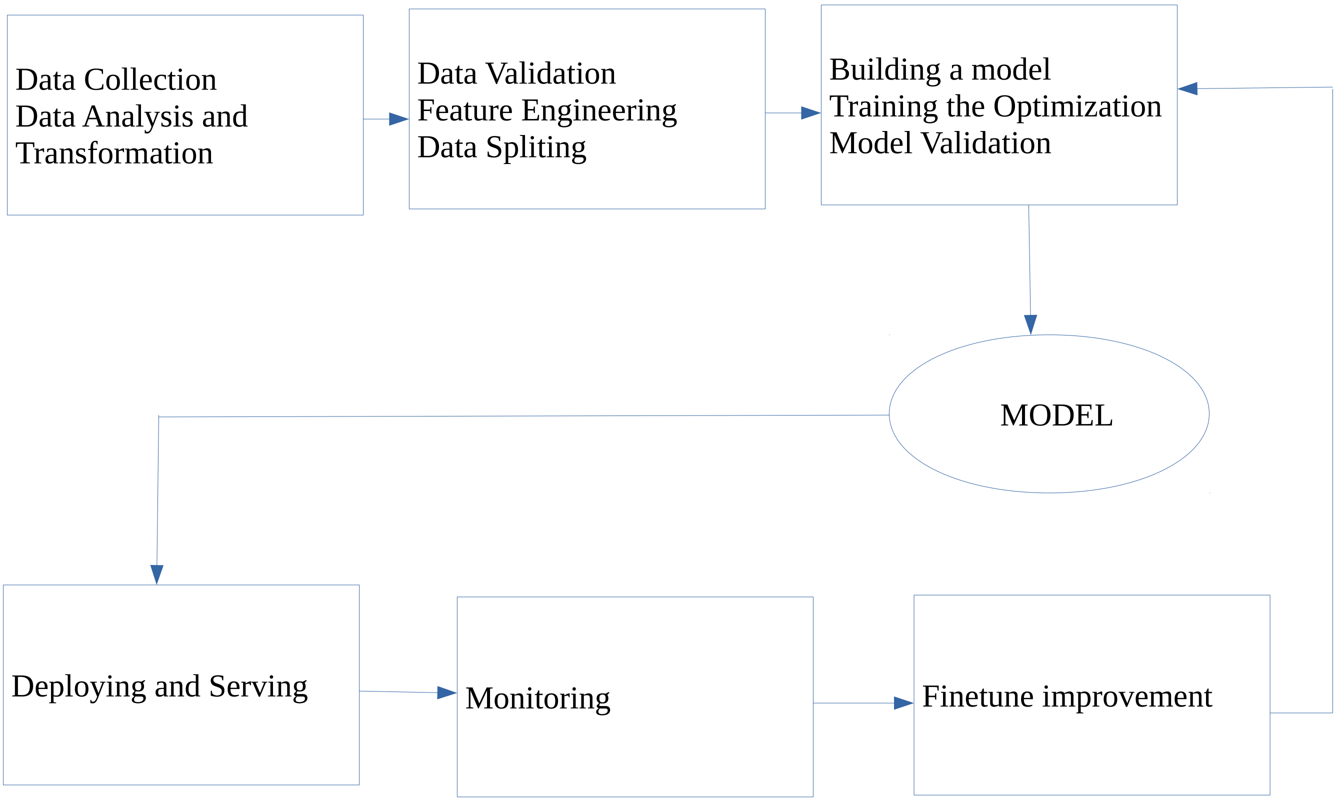
Training is done for 10 Epoch

Inference :

Training Loss : 0.2209
Accuracy : 0.903
Validation_loss : 0.2144
Validation_accuracy : 0.906

A Prediction script is made which will take a Input image as a input or a web url of the image,
Then it gives the output of the particular image thsis script is encripted in the docker container.
The dockerfile is attached in directory.

Architecture for incremental data addition for automatic update of Model :



Source Folder:

The Labellerr Folder contails Model file,Training script and prediction script in Jupyter notebook.

File name : Training.ipynb,prediction.ipynb.

Standalone script to test the image by user :

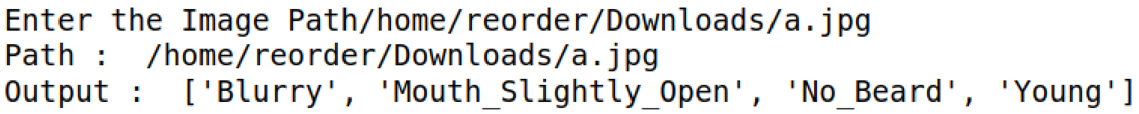
File name : predict_function.py

It has Docker file to create a docker image also .

Example :
python3 predict_function.py

input_image :

output :



```
Enter the Image Path/home/reorder/Downloads/a.jpg
Path :  /home/reorder/Downloads/a.jpg
Output :  ['Blurry', 'Mouth_Slightly_Open', 'No_Beard', 'Young']
```