Company overview comprehended

Labellerr is an AI training and Data Annotation platform for Computer Vision, Voice and NLP solutions. The goal of the company is to provide organizations a tool so that they can focus more on building AI models quickly rather than waiting on any third party services specially designed for application requiring labelling huge amounts of data. Labellerr also specializes in providing ready-to-deploy pre trained AI solution in various domains like Retail and E-commerce and deploy various models from customer tracking to autonomous checkouts. The company also actively participates in enhancing the field of AI by open sourcing datasets, publishing blogs on recent feats and open sourcing the research plan of their products. The company also provides data labelling services and actively works towards Ai assisted labelling techniques.

---------------------Deliverable on stage 1---------------------------

1. The dataset has an imbalance problem. There are some unique/rare attributes like Bald and Mustache with a very low frequency ,and a couple of very common attributes like No Beard and Young with a very high frequency. This means the data is biased towards young and shaved people .this can cause the data to over fit. Although this can be solved by tweaking the hyper parameters of the data and the loss function.

2.The data needs to be one hot encoded ; i.e it should be represented in binary(0,1) rather than -1 and 1 to get absolute frequency of data. thus ensuring simplified analysis of data.

3.The over fitting of this biased data can be mitigated using data augmentation. This can be achieved through the latest TensorFlow 2 architecture which simplifies this procedure with a few lines of code. This increases the amount of data we have thus removing the bias. Data augmentation on the training set leads to a more generalized/smarter model.

4.overfitting of the data can also be reduced using early stopping methodology, Here we call a callback() function during the training process of the model when it reaches a certain validation accuracy thus stopping the model to train before it starts to over fit.

5.Hyper parameters to tweak can be the use of the binary cross entropy loss function because in my cases it works the best on models to minimize information loss during loss estimation. And in terms of optimizers, since face data is not a primitive pattern we can go with the Adam optimizer (which changes learning rate dynamically) providing a much better accuracy on this non primitive patter recognition of faces.

-----------------------Deliverable on stage 2--------------------------------

I have created an facial feature extraction model from the dataset. This model scans a face and auto identifies top 10 facial features of the given face regardless of gender. The model used was mobilenetV2 for its fast inference speed, adequate mAP(mean average precision) and light weight. This model took 6 hours worth of hyper parameter fine tuning along with 8 hours of training for 5 epochs leading to 90% accuracy on the validation dataset. The model is then turned into an API and Dockerised and its image is attached below.

building the Docker image file

step 1:compress the facialAtrr folder to facialAtrr.rar

step 2: change the extension of the folder from facialAtrr.rar to facialAttrr.tar

Commands to run the image

```
docker load -i <path to image tar file>/facialAtrr.tar
```

```
eg:docker load -i D:/project/facial_attr/test/facialAtrr.tar
```

```
docker run -it -p 8888:8888 -v {path to this extracted folder}/Docker:/tf/ {image id}
```

```
eg: docker run -it -p 8888:8888 -v D:/project/facial_attr/data/Docker:/tf/ 72db6d35eef4
```

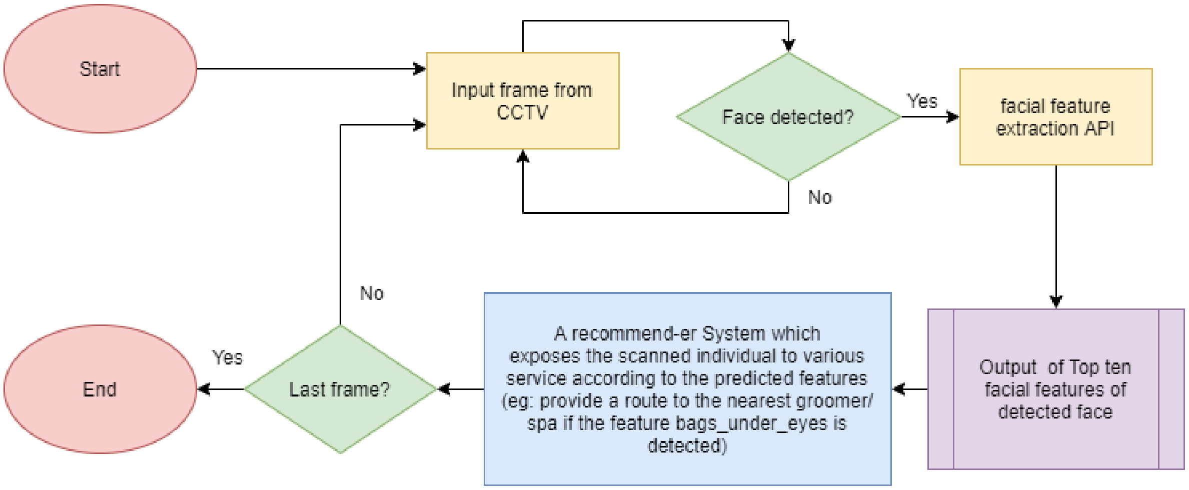
jupyter notebook can be explored for checking out the training model

or you can exit out of the jupyter notebook and run the API by

run API cmd: python run.py -i "image_path"

eg:python run.py -i "000005.jpg" <--while in image directory, this will run the model on a single image

--------------------------------Deliverable on stage 3--


Start
Input frame from CCTV
Face detected?
Yes
No
facial feature extraction API
Output of Top ten facial features of detected face
A recommend-er System which exposes the scanned individual to various service according to the predicted features (eg: provide a route to the nearest groomer/ spa if the feature bags_under_eyes is detected)
Last frame?
No
Yes
End

The model API can be easily adapted to CCTV input mode by running the API in a loop with CCTV frames as its input. When a face is detected in the CCTV input, our model starts inferencing and predicting the facial features of the detected face. The top 10 features are extracted and a recommender system uses these predictions to provide services to the detected individual(eg: provide a route to the nearest groomer/ spa if the feature bags_under_eyes is detected). This process is done in loop till the last frame is detected i.e the CCTV are turned off.

This system can provide with personalized services towards the user and can help them narrow down their services and options. This model can be extremely profitable to service providers, by providing them customers and exposure by this ML model.

The model works on the light weight MobilenetV2 architecture thus its computational requirements are very low, thus reducing the capital cost by setting up the machine up on a low powered system. Furthermore the model has high inference speed and accuracy, thus providing efficient processing with less power consumption and good accuracy. Since the model is completely autonomous it dowsnt require any long term maintenance and can work around the clock with no interruption and without constant supervision. Thus this architecture can be established on minimum capital and very low monthly power and maintenance cost.