Training with SSD Mobilenet v1:

We are using pretrained model from tensorflow keras that is SSD Mobilenet. We are training this model on our GPU and testing with the CPU.

Dependencies:

1. Numpy
2. Pandas
3. OS
4. Tensorflow
5. Keras
6. Io
7. OpenCV as cv2
8. Tensorflow object detection api

We have done the image processing for the input data. Based on the dimensional information given in the name of the file we are have done the string manipulation and got the
Xmin, Xmax, ymin, and ymax.

Data Preparation:

- I have prepared the data based on the previous data preprocessing results.
- Based on the 10 classes we have in the brand names we have one-hot encoded the classes.
- Created data frames based on the data preprocessing. Train_df, eval_df/.

Creating the TensorFlow records on the data:
I have created TensorFlow records on the data which i will be using in the training process.

Records are 1. Train. record, 2. Eval.record.

Training process:
Now after getting the records for training i have followed the following procedure.

1. Arranged the data in the local system and followed the procedure.
2. Used the code to train the model on GPU:
```
python3 train.py --logtostderr \
--train_dir=pack_detector/models/ssd_mobilenet_v1/train/ \
--pipeline_config_path=pack_detector/models/ssd_mobilenet_v1/ssd_mobilenet_v1_pack.co
nfig
```
3. Then i have started testing the model on the CPU:
```
CUDA_VISIBLE_DEVICES="" python3 eval.py \
--logtostderr \
--checkpoint_dir=pack_detector/models/ssd_mobilenet_v1/train \
--pipeline_config_path=pack_detector/models/ssd_mobilenet_v1/ssd_mobilenet_v1_pack.co
nfig \
--eval_dir=pack_detector/models/ssd_mobilenet_v1/eval
```
4. ON the testing data i have got an accuracy of 85%.
5. After training and testing the model I have run the model on the tensorboard to generate the inference graph:
```
tensorboard --logdir=pack_detector/models/ssd_mobilenet_v1
```

6. After getting the frozen inference graph i have started the implementation of the model using the inference graph.

Implementation:

1. After getting the frozen graph i have implemented it using the following dependencies:
    - Numpy
    - Pandas
    - Tensorflow
    - Tensorflow.compat
    - Keras
    - Collections
    - OpenCV
    - Object detection utils

2. Then we have prepared the images for our manual testing using OpenCV.
3. We have created both detection boxes and the fields of detection i have used the python functions to perform this task.
4. Totally i got an accuracy which is more than 85 % on the testing data and while implementing also the model is decently performing.