OUTDOOR NAVIGATION FOR VISUALLY IMPAIRED

NAME - LAKSHIT SHARMA

ROLL NO – 28700087

BRANCH – CSE

SUBJECT - DIGITAL IMAGE PROCESSING(DIP)

OTHER GROUP MEMBER NAME - DEEPANSHU JAIN

PROBLEM DESCRIPTION

- NAVIGATION THROUGH UNKNOWN ENVIRONMENT IS A CHALLENGING TASK FOR VISUALLY IMPAIRED . MAIN PROBLEMS THAT AFFECT A PERSON'S ABILITY TO NAVIGATE ARE 1.WHAT IS THE OBSTACLES ARROUND THEM , 2.HOW FAR THE OBSTACLE IS AND 3. IN WHICH DIRECTION THE OBSTACLE LIES. SO IF HE/SHE KNOW ABOUT OBSTACLE ARROUND AND ABOUT ITS LOCATION THAN HE/SHE CAN NAVIGATE ARROUND WITH LESS DIFFICULTIES. SO OUR MAIN PROBLEM IS TO RECOGNIZE THE OBJECT , DISTANCE OF OBJECT FROM CAMERA AND ANGLE AT WHICH OBJECT IS LOCATED FROM THE CENTRE OF THE SCREEN. WE CAN SOLVE THE PROBLEM EFFECTIVELY USING VISION LEARNING BUT HERE WE ONLY USES IMAGE PROCESSING AND MACHINE LEARNING .

METHODOLOGY

- FIRSTLY WE HAVE TO DETECT THE OBJECT AND THEN WE HAVE TO FIND THE LOCATION OF THE OBJECT (AT WHAT DISTANCE AND AT WHAT ANGLE THE OBJECT LIES).

OBJECT DETECTION - (CLASSIFICATION , SUPERVISED LEARNING)

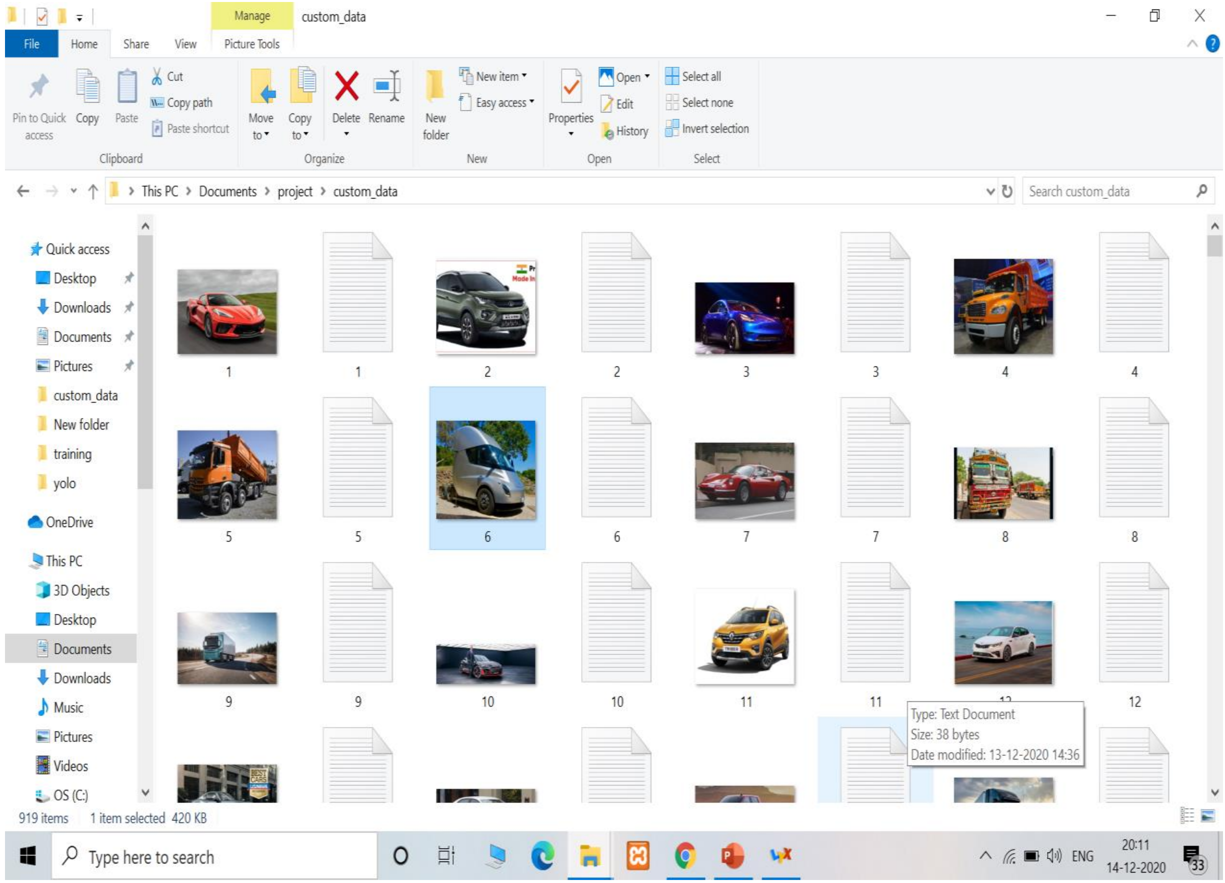
Firstly we trainned the darknet-YOLO_v3 model by some labelled data and

Then after 3-4 hours of trainning yolo.weight file is generated . Then using the .weight file and configuration file we detect the object in real time . After trainning with some labelled data , Now we trainned the model with COCO dataset of 80 images .

- train dataset-
- test dataset

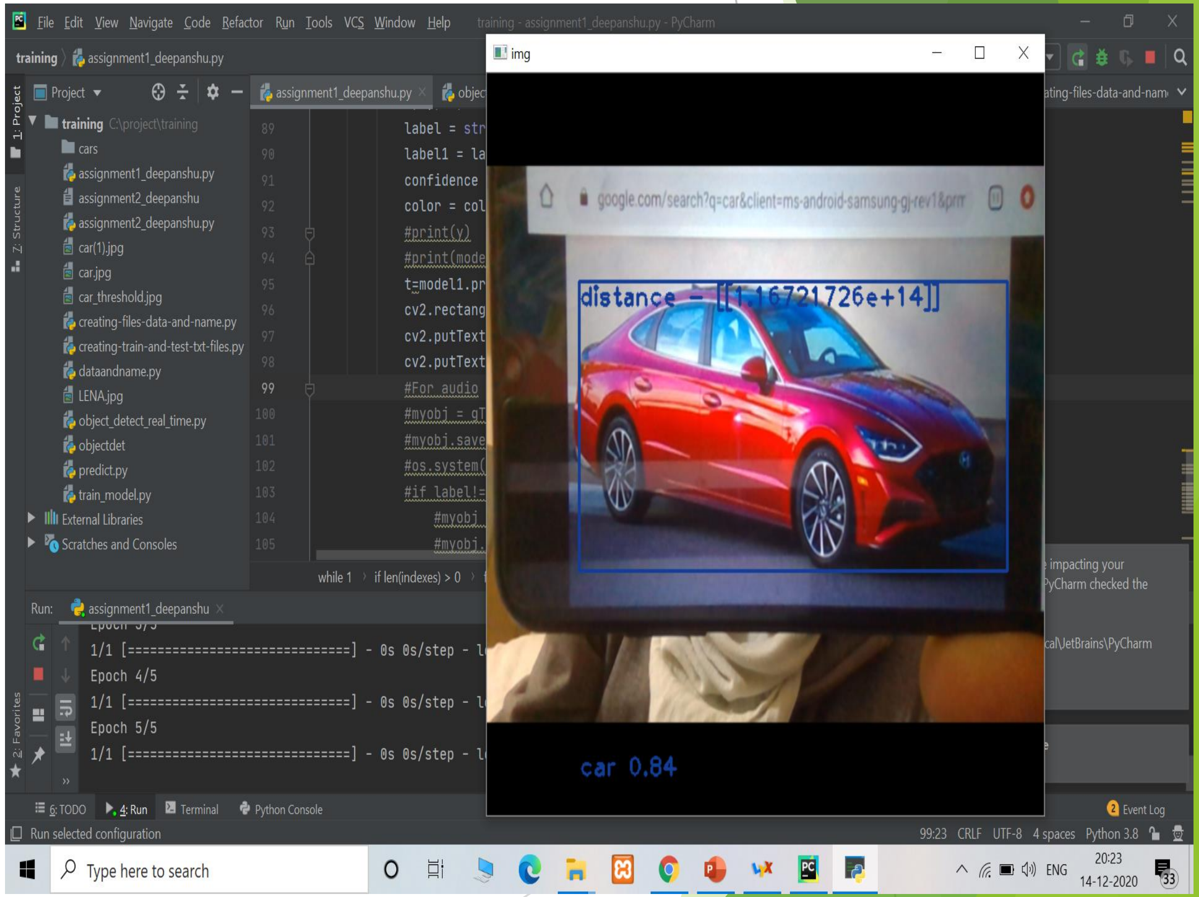
- FOR DISTANCE ESTIMATION
- We need to find the correlation between the pixel position (of the bottom of the detected object) and the actual distance in meteres . So , We train a simple single layer neural network for this and fine the relation between the pixel position of the bottom of the detected obeject and distance in meteres . Basically we need to find the distance in meter per pixel for a plane path .

Labelled dataset



Type: Text Document
Size: 38 bytes
Date modified: 13-12-2020 14:36

Working model



File Edit View Navigate Code Refactor Run Tools VCS Window Help
training - assignment1_deepanshu.py - PyCharm
training › assignment1_deepanshu.py
Project
training C:\project\training
cars
assignment1_deepanshu.py
assignment2_deepanshu
assignment2_deepanshu.py
car(1).jpg
car.jpg
car_threshold.jpg
creating-files-data-and-name.py
creating-train-and-test-txt-files.py
dataandname.py
LENA.jpg
object_detect_real_time.py
objectdet
predict.py
train_model.py
External Libraries
Scratches and Consoles
while 1 › if len(indexes) > 0
Run: assignment1_deepanshu
6: TODO
4: Run
Terminal
Python Console
Run selected configuration
Event Log
99:23 CRLF UTF-8 4 spaces Python 3.8
img
google.com/search?q=car&client=ms-android-samsung-gj-rev1&prm
Type here to search
ENG
20:23
14-12-2020

```
label = str
label1 = la
confidence
color = col
#print(y)
#print(mode
t=model1.pr
cv2.rectang
cv2.putText
cv2.putText
#For audio
#myobj = gT
#myobj.save
#os.system(
#if label!=
    #myobj
    #myobj.
```

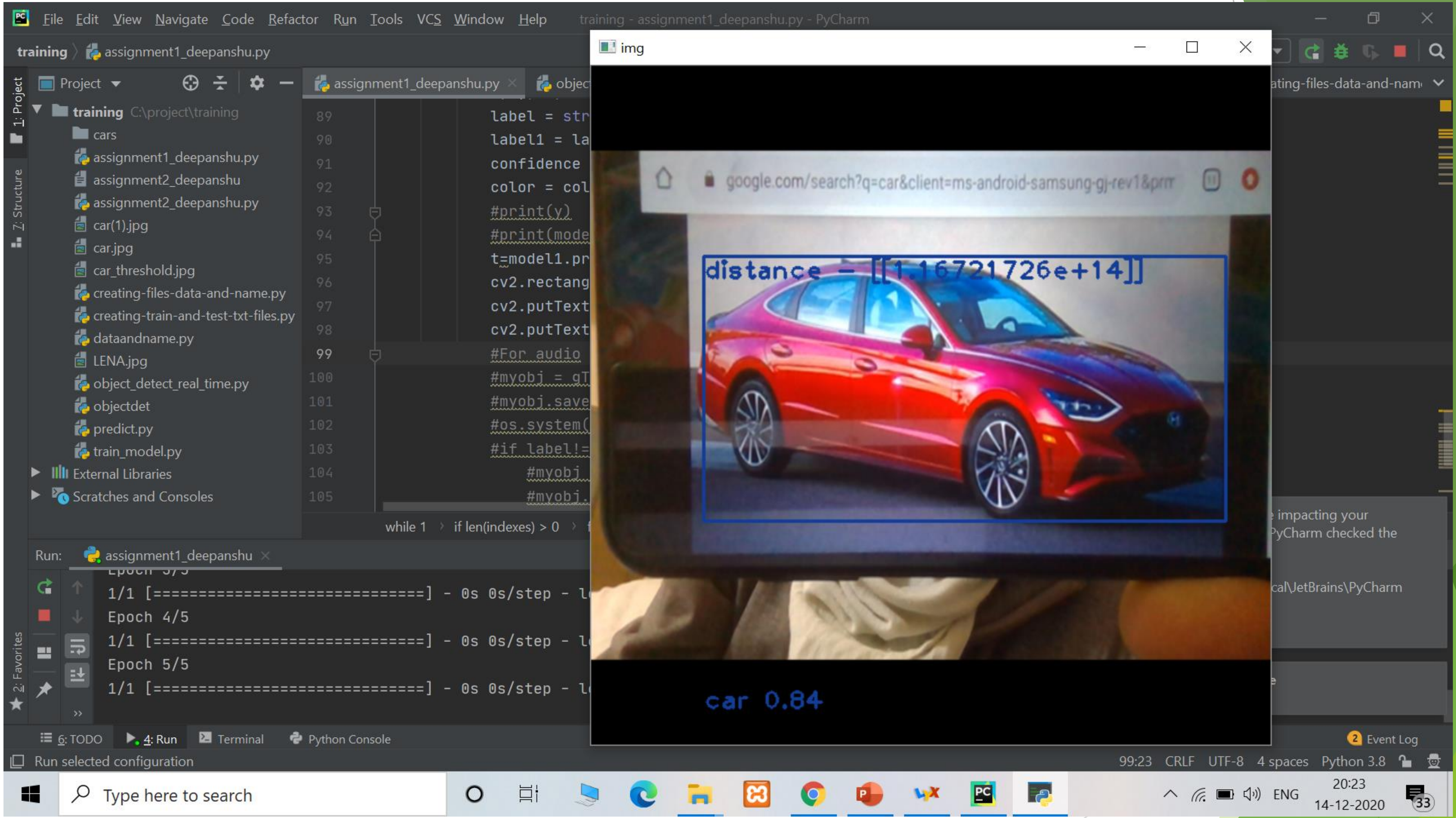
```
Epoch 3/5
1/1 [==============================] - 0s 0s/step - l
Epoch 4/5
1/1 [==============================] - 0s 0s/step - l
Epoch 5/5
1/1 [==============================] - 0s 0s/step - l
```

distance = [[1.16721726e+14]]

car 0.84

impacting your PyCharm checked the

cal\JetBrains\PyCharm


File Edit View Navigate Code Refactor Run Tools VCS Window Help
training - assignment1_deepanshu.py - PyCharm
training 〉 assignment1_deepanshu.py
Project
training C:\project\training
cars
assignment1_deepanshu.py
assignment2_deepanshu
assignment2_deepanshu.py
car(1).jpg
car.jpg
car_threshold.jpg
creating-files-data-and-name.py
creating-train-and-test-txt-files.py
dataandname.py
LENA.jpg
object_detect_real_time.py
objectdet
predict.py
train_model.py
External Libraries
Scratches and Consoles
assignment1_deepanshu.py
objec
ating-files-data-and-nam
img
google.com/search?q=car&client=ms-android-samsung-gj-rev1&prm
while 1 › if len(indexes) > 0
Run: assignment1_deepanshu
impacting your
PyCharm checked the
cal\JetBrains\PyCharm
6: TODO
4: Run
Terminal
Python Console
Event Log
Run selected configuration
99:23 CRLF UTF-8 4 spaces Python 3.8
Type here to search
ENG
20:23
14-12-2020

```
label = str
label1 = la
confidence
color = col
#print(y)
#print(mode
t=model1.pr
cv2.rectang
cv2.putText
cv2.putText
#For audio
#myobj = gT
#myobj.save
#os.system(
#if label!=
    #myobj
    #myobj.
```

```
Epoch 3/5
1/1 [==============================] - 0s 0s/step - l
Epoch 4/5
1/1 [==============================] - 0s 0s/step - l
Epoch 5/5
1/1 [==============================] - 0s 0s/step - l
```

```
distance = [[1.16721726e+14]]
car 0.84
```

- we did object detection , object localization , object recognization by training our model.
- We train model and provide different classes of object like human , car .
- Now After trainning the model a .weight file is generatted . So using this .weight and .cfg file we use the modle for real time object detection .
- And model train our object and identifies its shape by putting rectangle around it. And then we calculate distance of object from camera or person having it by our next model. For calculating distance Once we get the bounding box and midpoint of its bottom edge we can calculate its distance from person having camera. On that image, there is a direct correlation between the pixel position and distance in meters, so the distance between the calculated position of the midpoint and the bottom of the image multiplied by number of meters per pixel represents the distance between our car and the car we have detected

LIBRARIES USED -

TENSORFLOW:-TensorFlow used to create layers of our CNN

NUMPY :- To process the image matrices

OPENCV :- for image processing .

MATPLOTLIB :- To display the result of our predictive outcome.

OS :- os access the file system to read the image from the train and test directory from our machines and then devide the dataset into two train and test .

PANDAS :- For well structured dataset and manipulating data.

- **FOR DISTANCE ESTIMATION :- (REGRESSION , SUPERVISED LEARNING)**

We need to find the correlation between the pixel position (of the bottom of the detected object) and the actual distance in meteres . So , We train a simple single layer neural network for this and fine the relation between the pixel position of the bottom of the detected obeject and distance in meteres . Basically we need to find the distance in meter per pixel for a plane path .

Assumptions-

1.Path on which object is detected should be plane .

2.position of camera should be fixed on the height at which our model is trainned .

- Trainning of distance estimater neural network
- Epoch 1/5
- 1/1 [==============================] - 0s 999us/step - loss: 146409.3438
- Epoch 2/5
- 1/1 [==============================] - 0s 0s/step - loss: 5964812800.0000
- Epoch 3/5
- 1/1 [==============================] - 0s 0s/step - loss: 244149479538688.0000
- Epoch 4/5
- 1/1 [==============================] - 0s 0s/step - loss: 9993436995600252928.0000
- Epoch 5/5
- 1/1 [==============================] - 0s 0s/step - loss: 409047758808036679876608.0000

CHALLENGES

- 1.First of all a good dataset is a big challenge for machine learning projects .
- Firstly we trainned our model by our self-made labelled dataset done by labelImg .Then we trainned the model by COCO dataset of 80 categories .
- Link of COCO dataset is
- http://images.cocodataset.org/zips/train2014.zip
- 2.Trainning time is also a big challenge .
- 3.Accuracy is also a issue .
- 4.For distance estimation , many assumptions should be there .
- Our coding portion for model is on github
- Github link -
- https://github.com/deepan773399/outdoor-navigation-for-visually-blind

CONCLUSIONS AND FUTURE WORKS

- The project will definetly help the visually impaired peron in outdoor navigation .
- Future work – We also need to find the angle at which objct is located .
- Proper use of gTTS libraries for speech .
- We also need to link the poject with location (maps) so that , person can easily navigate arround and go from one place to another .