Object detection using OpenCV and Python

Project Report submitted in partial fulfillment

for the award of the degree of

B-TECH CSE

Submitted By

1.Vaibhav Maheshwari (18scse1010414)

2. Ravikant kumar (18scse1010675)

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Under the Supervision of-

Dr. Dileep Kumar Yadav

GALGOTIAS
UNIVERSITY

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE'S DECLARATION

I/We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"OBJECT DETECTION USING OPEN CV AND PYTHON"** in partial fulfilment of the requirements for the award of the B.Tech(CSE) submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of january,2022 to May,2022 under the supervision of DR. Dileep Kumar Yadav, Department of Computer Science and Engineering/Computer Application and Information and Science, of School of Computing Science and Engineering , Galgotias University, Greater Noida

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Vaibhav

Maheshwari(18SCSE1010414)

Ravikant kumar (18SCSE1010675)

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr. Dileep Kumar Yadav

CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Vaibhav Maheshwari (18SCSE1010414) and Ravikant Kumar (18SCSE1010675) has been held on _________________ and his/her work is recommended for the award of B.Tech

Signature of Examiner **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: 13 May, 2022

Place: Greater Noida

Statement of Project Report

Preparation

1.Thesis title: **Object detection using OpenCV and Python**

2.Degree for which the report is submitted: **B. Tech CSE**

3.Project Supervisor was referred to for preparing the report.

4.Specifications regarding thesis format have been closely followed.

5.The contents of the thesis have been organized based on the guidelines.

6.The report has been prepared without resorting to plagiarism.

i.All sources used have been cited appropriately.

ii. The report has not been submitted elsewhere for a degree.

(Signature of the student)

Name: vaibhav maheshwari
Roll No: **18021011645**

Statement of Project Report

Preparation

1. Thesis title: **Object detection using OpenCV and Python**

2. Degree for which the report is submitted: **B. Tech CSE**

3. Project Supervisor was referred to for preparing the report.

4. Specifications regarding thesis format have been closely followed.

5. The contents of the thesis have been organized based on the guidelines.

6. The report has been prepared without resorting to plagiarism.

   i. All sources used have been cited appropriately.
   ii. The report has not been submitted elsewhere for a degree.

(Signature of the student)

Name: ravikant kumar
Roll No: **18021011899**

TABLE OF CONTENT

AKNOWLEDGEMENT

The success and result outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to thank them.I respect and thank our guide: Dr. DK.Yadav Sir for providing me an opportunity to do the project work under their guidance and give us all support and guidance which made me complete the project duly. I am extremely thankful to my college Galgotias University for providing such nice support and guidance.

ABSTRACT

This paper covers topics starting from how computing and machine learning algorithms facilitate object detection to how OpenCV is such a great tool for beginners who want to be told however real time object detection and pursuit is done. It conjointly shows the flexibleness of a pursuit system to a moving camera, ideal for automotive safety applications. the item pursuit algorithmic program is termed centre of mass pursuit because it depends on the geometrician distance between (1) existing object centroids (i.e., objects the centre of mass huntsman has already seen before) and (2) new object centroids between ensuant frames in an exceedingly video.

LIST OF FIGURE

INTRODUCTION

The ASCII text file laptop Vision library (Ooen CV) could be a laptop vision library that gives many computers and machine learning algorithms and thousands of functions composing and supporting those algorithms. The library comes with C++, Java, and Python interfaces and supports all common desktop and mobile operating systems. OpenCV includes numerous modules, like a picture process module, object detection module, and machine learning module. exploitation of this library, we will acquire, compress, enhance, restore, and extract information from pictures. The rising algorithms, that represent the computer code side of the technology, are divided into totally different classes, like feature-based, learning-based, and hybrid (i.e., feature and learning-based). However, this pursuit features a sizable amount of advantages, few of them area unit traffic watching, golem vision, police work and security and video communication, public areas like underground stations, airports, mass events and animation. Thus, the actual application desires the best trade-off among computing, communication, and accuracy over the network. The revenue associated with computing and communication depends on the quantity and sort of cooperation dead among cameras for information assortment, dispensing and process to verify selections and to cut back the estimation errors and ambivalency.

Real-World Cases of Object Detection: Now, video object detection is being established across a good vary of industries. Its usage ranges from video police work, sports broadcasting, electrical cars, golem navigation and plenty of additional. How Video Object Detection works: So, however, can we find a moving object? The objective is to capture the coordinates of a moving object and spot that object within the video.

Main objective of the paper is described as follows.

- The primary objective of this work is to develop a framework for object detection and tracking in a video using various approaches.
- Based on the framework developed, it is also extended for the detection of other features of a human face like eyes and nose.

CASCADE OBJECT DETECTOR

The traincascadeobjectdetector could be a classifier operate for vision. cascadeobject detectors system object is gift in matlab’s laptop vision tool cabinet. it's accustomed produce a custom classifier for sleuthing classes of objects whose ratio doesn't amendment (fixed), like faces, human full-body, cars, body options, plate numbers.

cascade object detector could be a cluster of cascade classifiers organized little by little. every stage consists of call stumps (weak learners). every stage selects smaller numbers of options employing a technique referred to as boosting (i.e., adaboost). adaboost creates associate in nursing correct complicated classifier by combining the typical weight of the choice taken by the weak classifiers. for example, allow us to think about the adaboost classifier’s learning as a suggestion of a gaggle of consultants. The classification result by every professional nut for the input xi is expressed as en(xi). to differentiate between the coaching vector of 2 outcomes, en(xi) will solely settle for 2 results that area unit portrayed as +1 or −1, i.e., en(xi) ∈ .the collective opinion of the consultants is denoted as k(xi). it represents the linear combination of the weighted add of professional suggestions, which might be expressed as follows: k(xi) = w1e1(xi) + w2e2(xi) + · · · + wnen(xi) wheree1(xi), e2(xi), . . ., en(xi) represents the selections from n consultants and w1, w2, . . ., wn area unit the weights given to every professional suggestion. The procedural steps of the adaboost algorithmic program for classifier learning of options area unit given as follows: 1. given a coaching information with n pairs of pictures (xi, yi), wherever xi could be a positive or negative image, and yi is that the label assigned to every image. the worth for positive pictures is one, which of negative pictures is zero. 2. initializing weights w1, i= 1/2m, 1/2l for positive and negative pictures, wherever m and l area unit the amount of negative and positive pictures, severally. 3. for t = 1, . . ., t, wherever t is that the variety of stages of coaching and n pairs of pictures. 4. update the load wt+1, i = wt, iβ 1−ei t, wherever ei = zero once xi is assessed properly, ei = one otherwise and βt = et/ (1 − et). image recognitionimage recognition

- Image recognition (with AI) uses techniques like object detection, visual perception, and segmentation. The use of AI and mil enhances the speed of information process and therefore the quality of the ultimate result. as an example, by exploitation AI platforms, we will terribly simply accomplish complicated tasks. The additional typical laptop vision and mil approach as hostile deep learning, the subsequent are useful: determining the foremost acceptable options (HAAR options, image color bar chart, etc.) inspecting and learning regarding numerous techniques provided by OpenCV applying grid search to look the acceptable classifier Also, the subsequent issues might arise: To decide correct position of our slippy windows To make positive that they don’t overlap Recognizing the brink for overlapping detection Using acceptable frame rate Keeping min detection count over multiple frames
- Collecting combined window dimensions for avoiding overlapping detections

1.1 Keywords for the Project:

1.Pedestrian pursuit
2.Moving vehicle
3.Intelligent vehicles
4.Unattended driving systems
5.Intelligent driving selections

REQUIREMENT & SCOPE OF THE PROJECT

2.1 Required tools and skills:

- Visual Studio Code
- Open CV
- Python
- Haar Cascade classifier
- XML

2.2 Scope of the Project:

Computer vision is still a developing discipline, it has not been matured to that level where it can be applied directly to real life problems.

After a few years" computer vision and particularly object detection will not be any more futuristic and will be ubiquitous. For now, we can consider object detection as a sub-branch of machine learning.

Some common and widely used application of object detection are:

1 Face Detection

Have you ever wondered how Facebook detects your face when you upload a photo? Not only it detects, it remembers the face too. This is a simple application of object detection that we see in our daily life.

2 Counting objects/peoples

Object detection can be also used for counting purpose, it is used for keeping a count of particular or all objects in an image or a frame. For e.g. from a group photograph it can count the number of persons and if implemented smartly you may also find out different people with different dresses.

3 Vehicle detection

Similarly, when the object is a vehicle, object detection along with tracking can be used for finding the type of vehicle, this application may be extended to even make a traffic calculator.

4 Industries

Object detection is also used in industrial processes for the identification of different products. Say you want your machine to only detect objects of a particular shape, you can achieve it very easily. For e.g. Hough circle detection transform [6] can be used for detecting circular objects.

5 Security

Identification of unwanted or suspicious objects in any particular area or more specifically object detection techniques are used for detecting bombs/explosives. It is also used for personal security purposes.

6 Biometric recognition

Biometric recognition uses physical or behavioral traits of humans to recognize any individuals for security and authentication purposes [1]. It uses distinct biological traits like fingerprints, hand geometry, retina and iris patterns etc.

7 Surveillance

Objects can be recognized and tracked in videos for security purposes. Object recognition is required so that the suspected person or vehicle can be tracked

8 Medical analysis

Object detection is used to detect diseases like a tumor, stones, cancer in MRI images

9 Optical character recognition

Characters in scanned documents can be recognized using object recognition

10 Human computer interaction

Human gestures can be stored in the system and can be used for recognition in a dynamic environment by computers to interact with humans.

Object detection‟s scope is not yet limited here. You can use it for any purpose you can think of. For e.g. for solving number puzzles by just giving their images as input and applying some proper algorithms after detecting different numbers and their places from the input image.

ANALYSIS OF THE PROJECT & ACTIVITY TIME SCHEDULE

3.1 Problem Formulation:

Safety standards and accident prevention systems in cars have gained notable development. Advanced driver assistance systems are commercially available now and mostly found pre-installed by car manufactures. The aim of these systems is to provide automobiles with sensors capable of detecting and acting when threatening situations are faced, so that the driver would be able to avoid a collision. Detecting pedestrians, launches an alert and then acts upon their unpredictable behavior.

Thus, the full potential of ADAS systems can be achieved by including AI car and pedestrian tracing. To utilize such strong ML with computer vision methods, we employ a cascade of classifiers. Each classifier is trained to model a pedestrian. The targets are detected Haar features. This process is done to be able to operate in real-time and will help in future, especially in self-driving cars.

This was a tricky project, as we opted for the more conventional computer vision and machine learning approach as opposed to deep learning. The following steps were quite time consuming:

- determining the most suitable features (HAAR features, image color histogram, and other pre-requisites)
- exploring and learning at the same time
- applying grid search to find the most suitable classifier

Moreover, in our pipeline we struggled with the following:

- Determining correct position of our sliding windows and the overlap
- Identifying suitable *threshold* for overlapping detection

- Adopting suitable frame sampling rate
- Finding a good enough minimum detection count over multiple frames
- Aggregating the combined window dimensions for overlapping detections

MERITS

While driving, if you take your eyes off of the road to reach for your coffee, turn to tell your kids to quiet down, and then suddenly you see a pedestrian is crossing the road right in front of you. You push the brakes—but it might be late.

It is a very scary yet usual scenario. One out of three crashes involve a vehicle going straight as a pedestrian crosses the road. To prevent these mis-happenings, automakers now provide a "pedestrian detection" system in some of their models. The ability to reliably detect vehicles provides a huge advantage to everyone.

To prevent these crashes, automakers now offer a "pedestrian detection" system in some models. If the system detects that a pedestrian could be in the vehicle's travel path, it alerts the driver or employs automatic emergency braking, preventing what could be a fatal crash.
The ability to reliably detect vehicles offers significant advantages for asset management, resource allocation, site safety, and traffic control. Identifying the right technology for your vehicle-detection application can be challenging, and many factors must consider, including task, size of target, sensing range, sensor mounting, and whether the application is primarily indoor or outdoor.

COMPARATIVE STUDY

A. INTRODUCTION:

To acknowledge people during a real video closed-circuit television, we tend to spot humans mechanically. Pedestrian detection detects areas wherever somebody's gift is.

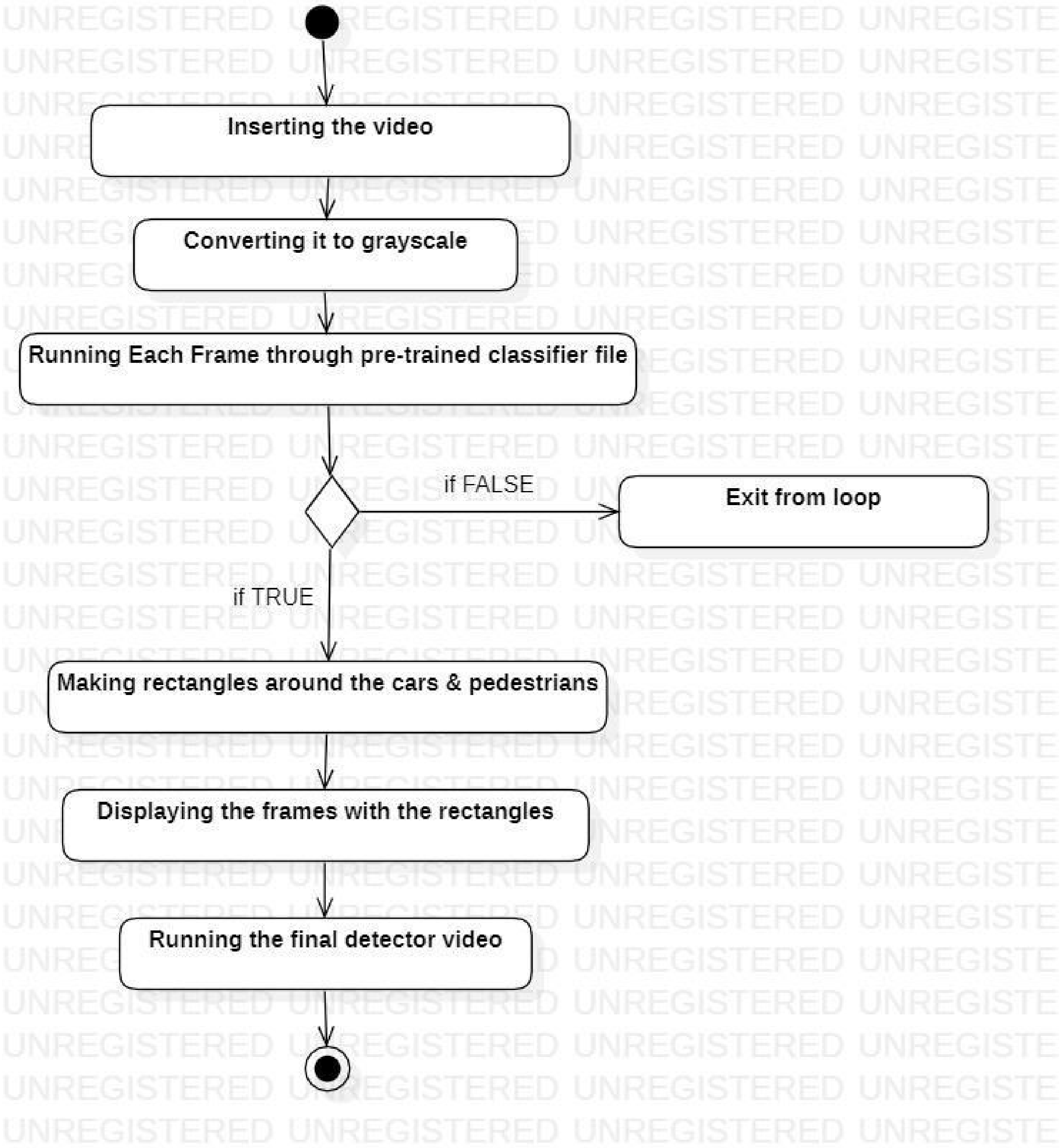
Person detection is the method of predicting and localizing all and sundry within the image, diagrammatic during a rectangular box that's created round the image.

B. METHOD OF PEDESTRIAN TRACING:

It mainly has 2 models:

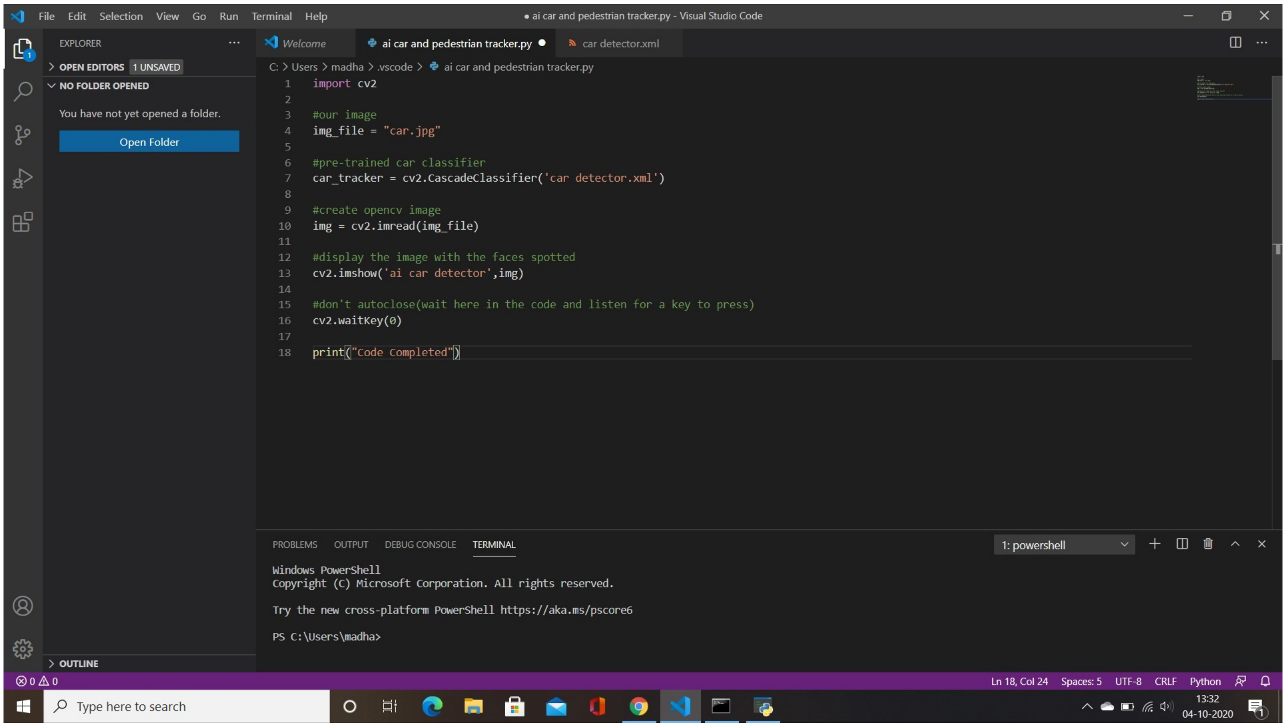
1. Hand-crafted model which is based on hand-crafted features and is widely used for object detection.
2. Deep learning models use convolutional neural networks to focus on improved performance

Architecture Diagram for Proposed method:


Inserting the video
Converting it to grayscale
Running Each Frame through pre-trained classifier file
if FALSE
Exit from loop
if TRUE
Making rectangles around the cars & pedestrians
Displaying the frames with the rectangles
Running the final detector video

Implementation and Description of Project Modules:

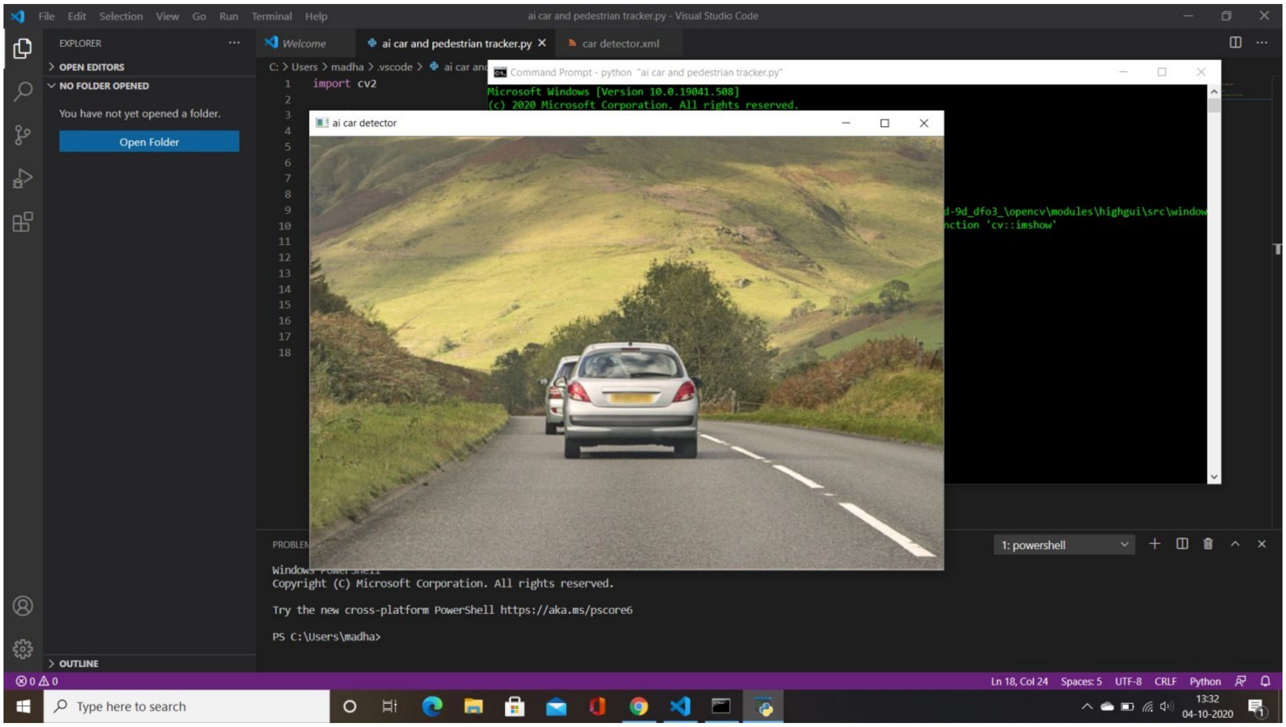
Module 1:


File Edit Selection View Go Run Terminal Help
● ai car and pedestrian tracker.py - Visual Studio Code
EXPLORER
OPEN EDITORS 1 UNSAVED
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and pedestrian tracker.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
Ln 18, Col 24 Spaces: 5 UTF-8 CRLF Python
Type here to search
13:32
04-10-2020

```python
import cv2

#our image
img_file = "car.jpg"

#pre-trained car classifier
car_tracker = cv2.CascadeClassifier('car detector.xml')

#create opencv image
img = cv2.imread(img_file)

#display the image with the faces spotted
cv2.imshow('ai car detector',img)

#don't autoclose(wait here in the code and listen for a key to press)
cv2.waitKey(0)

print("Code Completed")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

Fig 1


File Edit Selection View Go Run Terminal Help
ai car and pedestrian tracker.py - Visual Studio Code
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and
Command Prompt - python "ai car and pedestrian tracker.py"
ai car detector
PROBLEN
1: powershell
Ln 18, Col 24 Spaces: 5 UTF-8 CRLF Python
Type here to search
13:32
04-10-2020

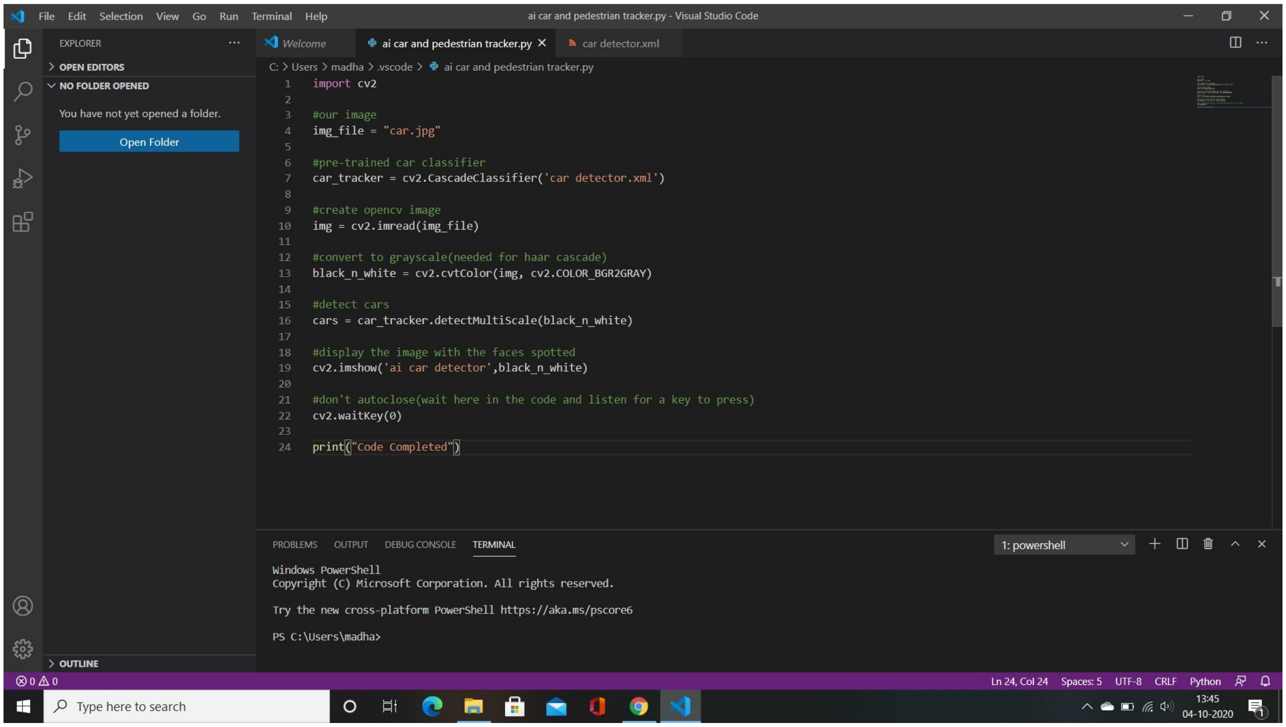
```python
import cv2
```

```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

d-9d_dfo3_\opencv\modules\highgui\src\window
nction 'cv::imshow'
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

fig 2

Module 2:

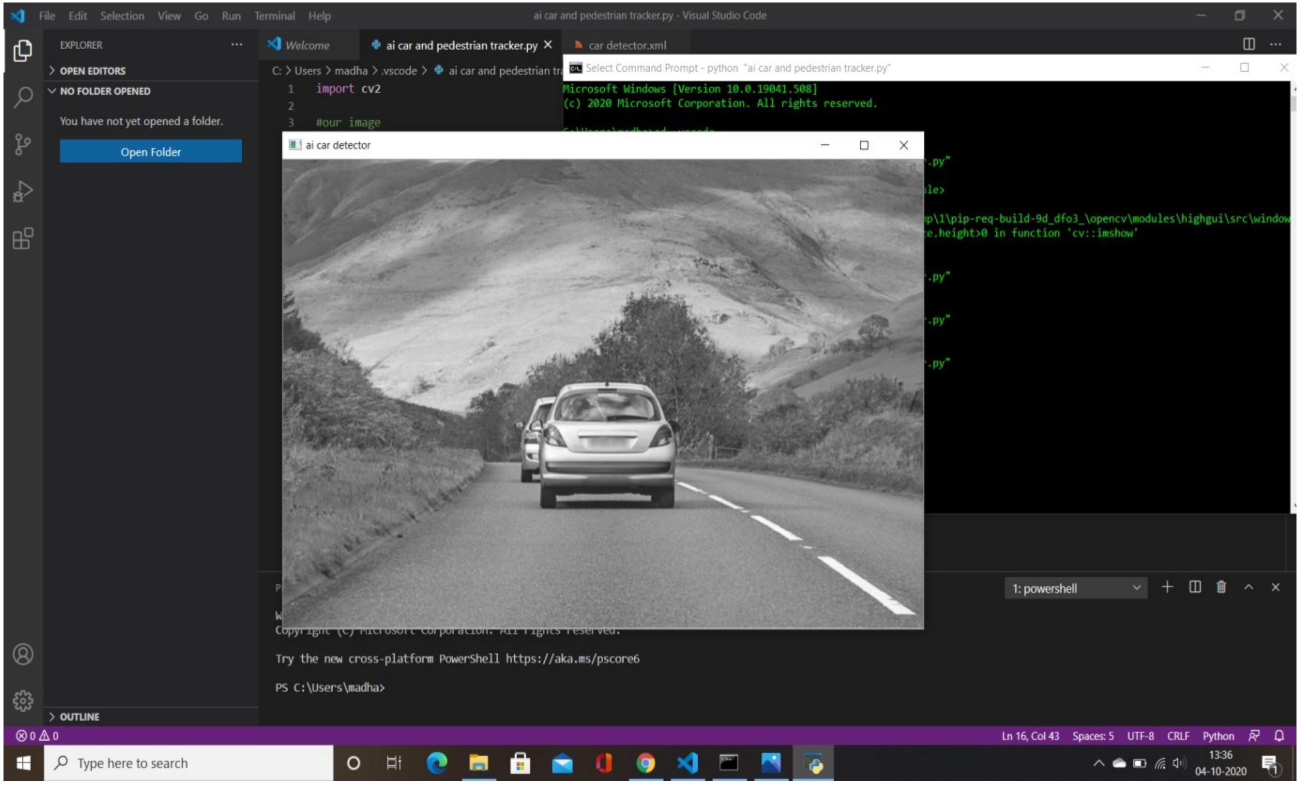


```
import cv2

#our image
img_file = "car.jpg"

#pre-trained car classifier
car_tracker = cv2.CascadeClassifier('car detector.xml')

#create opencv image
img = cv2.imread(img_file)

#convert to grayscale(needed for haar cascade)
black_n_white = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

#detect cars
cars = car_tracker.detectMultiScale(black_n_white)

#display the image with the faces spotted
cv2.imshow('ai car detector',black_n_white)

#don't autoclose(wait here in the code and listen for a key to press)
cv2.waitKey(0)

print("Code Completed")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

ai car and pedestrian tracker.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and pedestrian tracker.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
Ln 24, Col 24 Spaces: 5 UTF-8 CRLF Python
Type here to search
13:45
04-10-2020

Fig 3



```
import cv2

#our image
```

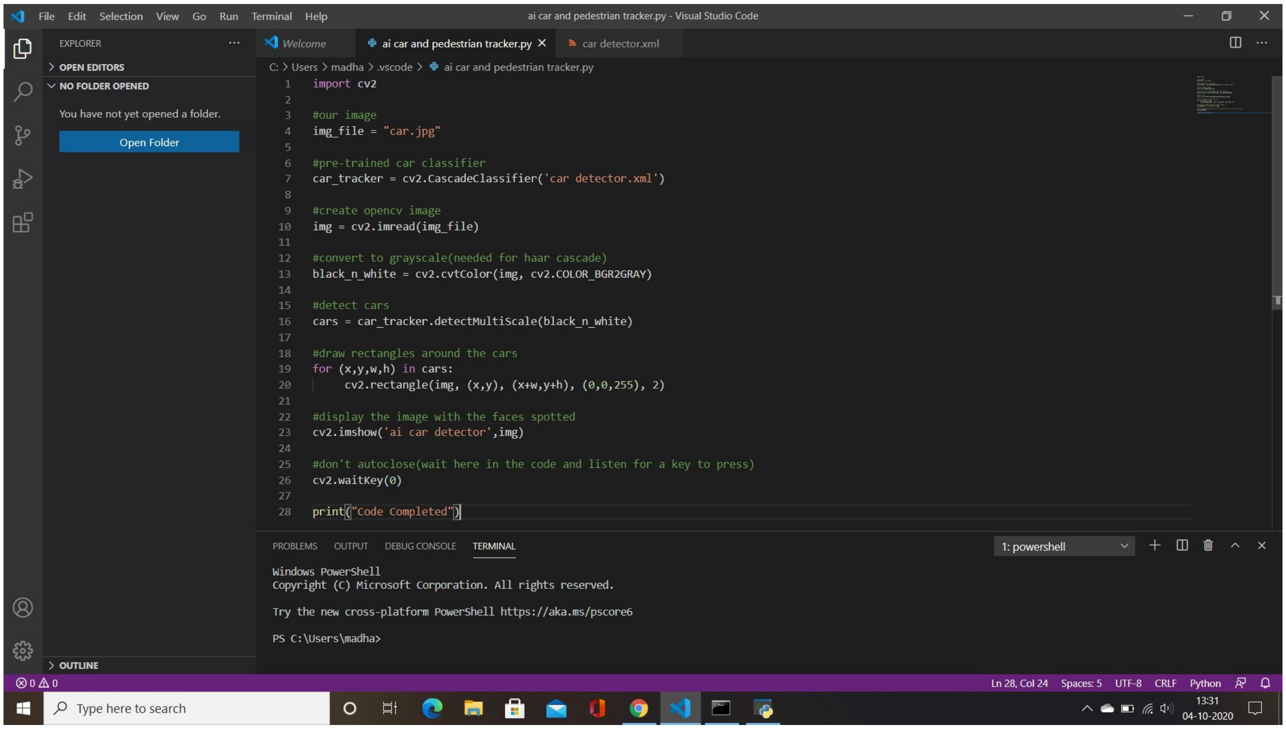
```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

.py"

le>

p\1\pip-req-build-9d_dfo3_\opencv\modules\highgui\src\window
e.height>0 in function 'cv::imshow'

.py"

.py"

.py"
```

```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

ai car and pedestrian tracker.py - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Welcome
ai car and pedestrian tracker.py
car detector.xml
Select Command Prompt - python "ai car and pedestrian tracker.py"
ai car detector
1: powershell
Ln 16, Col 43 Spaces: 5 UTF-8 CRLF Python
Type here to search
13:36
04-10-2020

Fig 4

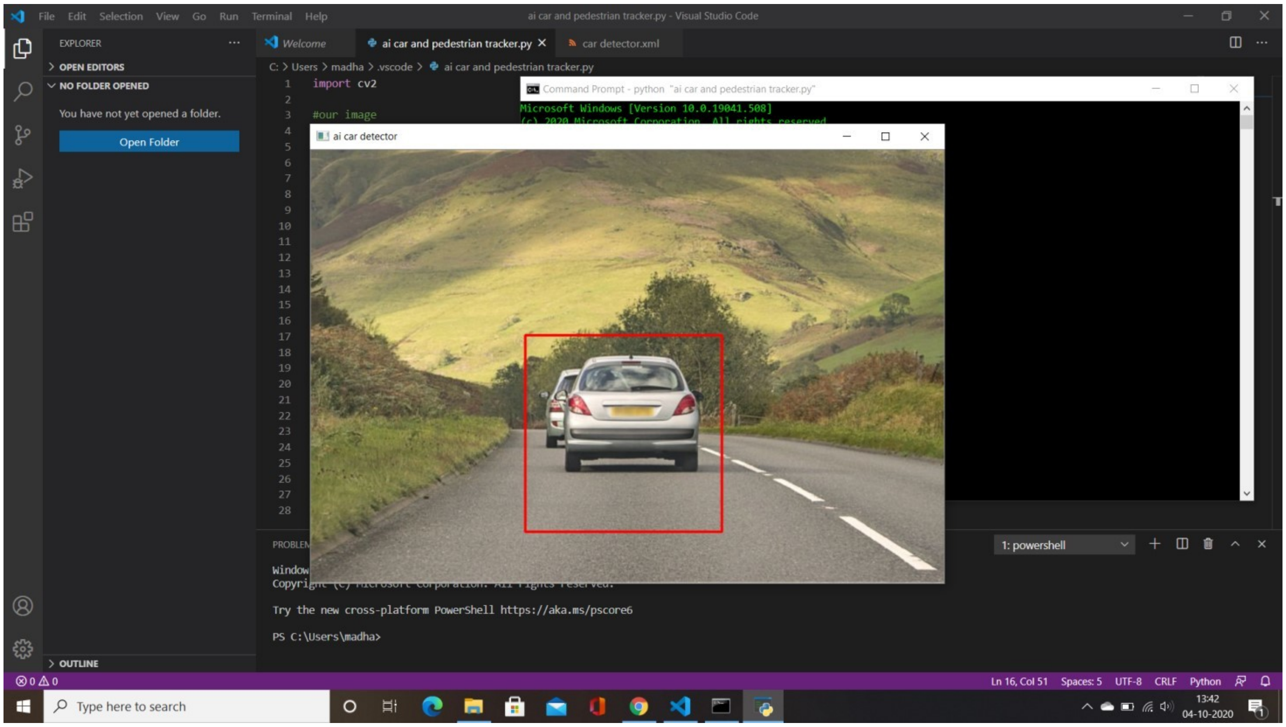
Module 3:


File Edit Selection View Go Run Terminal Help
ai car and pedestrian tracker.py - Visual Studio Code
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and pedestrian tracker.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
OUTLINE
Ln 28, Col 24 Spaces: 5 UTF-8 CRLF Python
Type here to search
13:31
04-10-2020

```python
import cv2

#our image
img_file = "car.jpg"

#pre-trained car classifier
car_tracker = cv2.CascadeClassifier('car detector.xml')

#create opencv image
img = cv2.imread(img_file)

#convert to grayscale(needed for haar cascade)
black_n_white = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

#detect cars
cars = car_tracker.detectMultiScale(black_n_white)

#draw rectangles around the cars
for (x,y,w,h) in cars:
     cv2.rectangle(img, (x,y), (x+w,y+h), (0,0,255), 2)

#display the image with the faces spotted
cv2.imshow('ai car detector',img)

#don't autoclose(wait here in the code and listen for a key to press)
cv2.waitKey(0)

print("Code Completed")
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

Fig 5

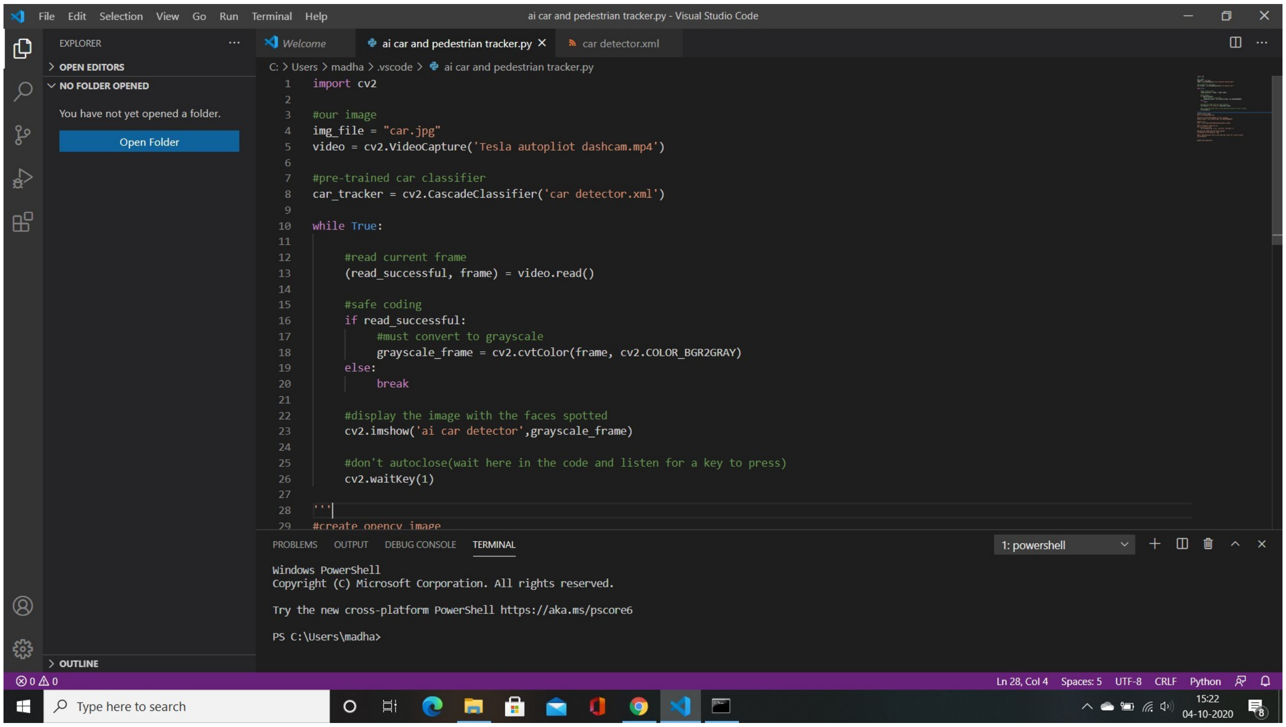

File Edit Selection View Go Run Terminal Help
ai car and pedestrian tracker.py - Visual Studio Code
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and pedestrian tracker.py
Command Prompt - python "ai car and pedestrian tracker.py"
ai car detector
1: powershell
OUTLINE
Ln 16, Col 51 Spaces: 5 UTF-8 CRLF Python
Type here to search
13:42
04-10-2020

```
Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

Fig 6

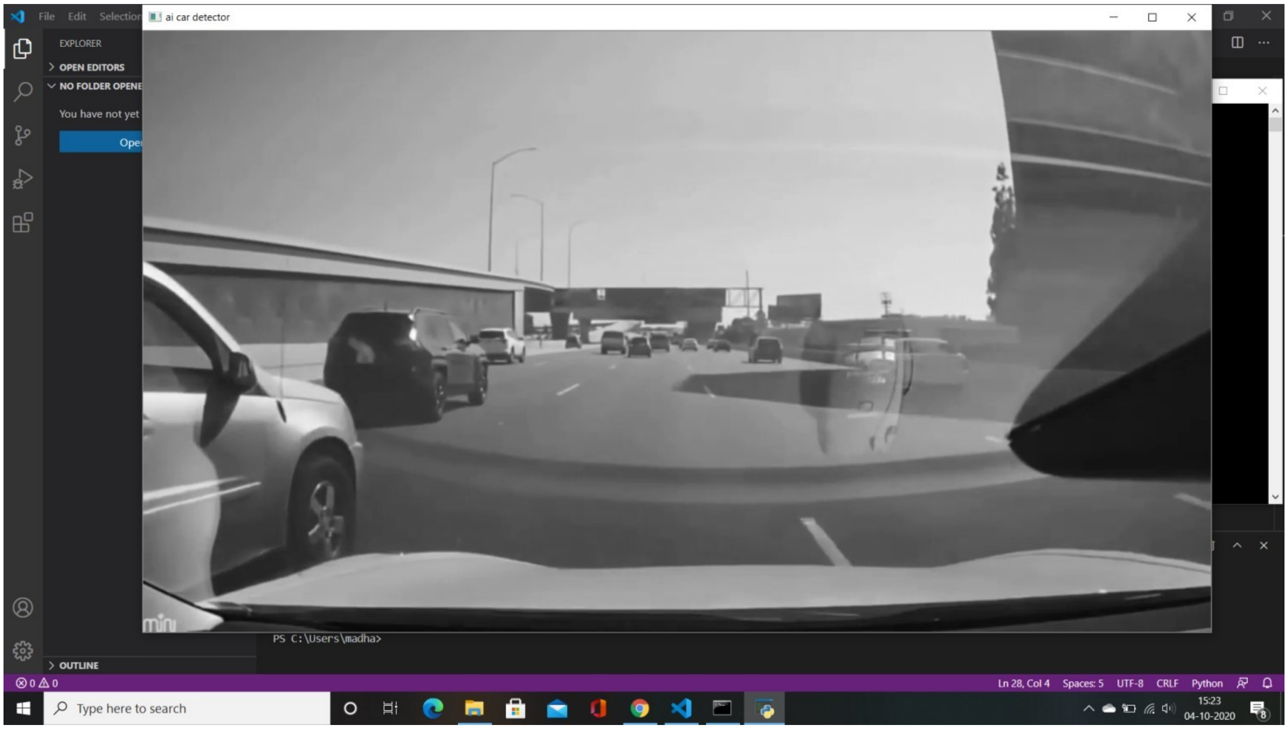
For Example:


File Edit Selection View Go Run Terminal Help
ai car and pedestrian tracker.py - Visual Studio Code
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and pedestrian tracker.py
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
1: powershell
Ln 28, Col 4 Spaces: 5 UTF-8 CRLF Python
Type here to search
15:22
04-10-2020

```python
import cv2

#our image
img_file = "car.jpg"
video = cv2.VideoCapture('Tesla autopliot dashcam.mp4')

#pre-trained car classifier
car_tracker = cv2.CascadeClassifier('car detector.xml')

while True:

     #read current frame
     (read_successful, frame) = video.read()

     #safe coding
     if read_successful:
          #must convert to grayscale
          grayscale_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
     else:
          break

     #display the image with the faces spotted
     cv2.imshow('ai car detector',grayscale_frame)

     #don't autoclose(wait here in the code and listen for a key to press)
     cv2.waitKey(1)

'''
#create opencv image
```

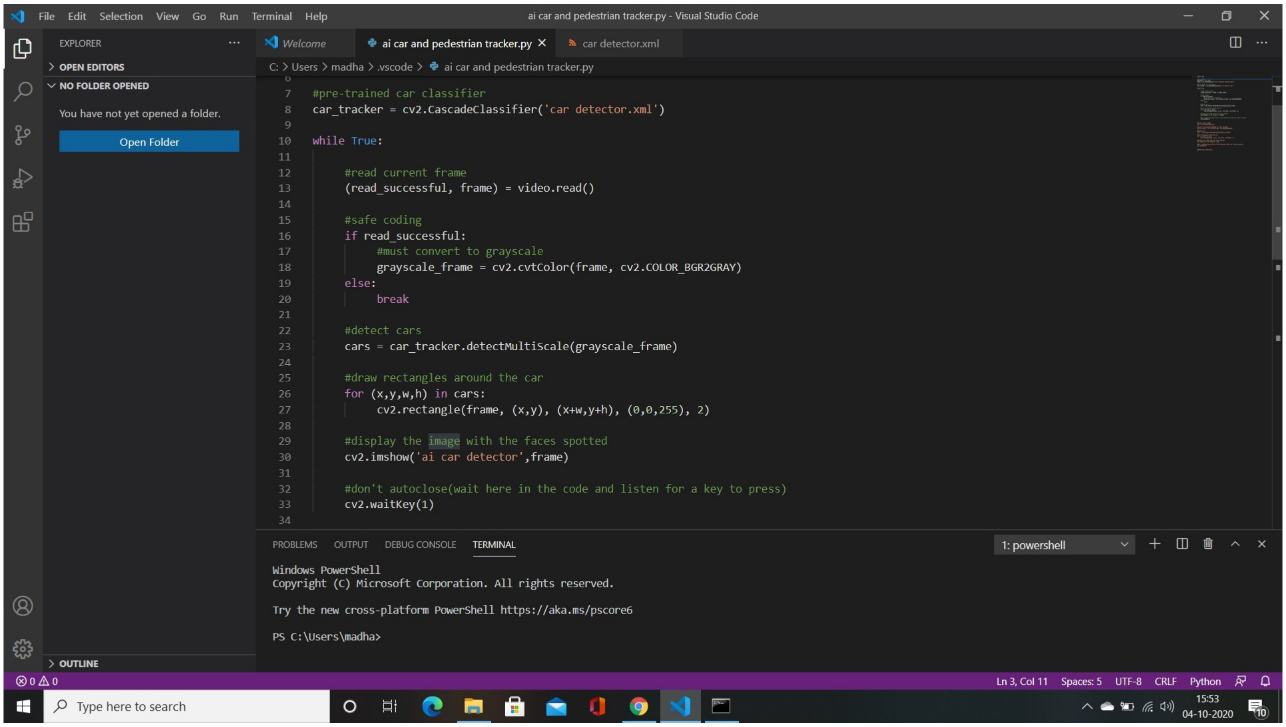
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

fig 7



```
PS C:\Users\madha>
```

Fig 8

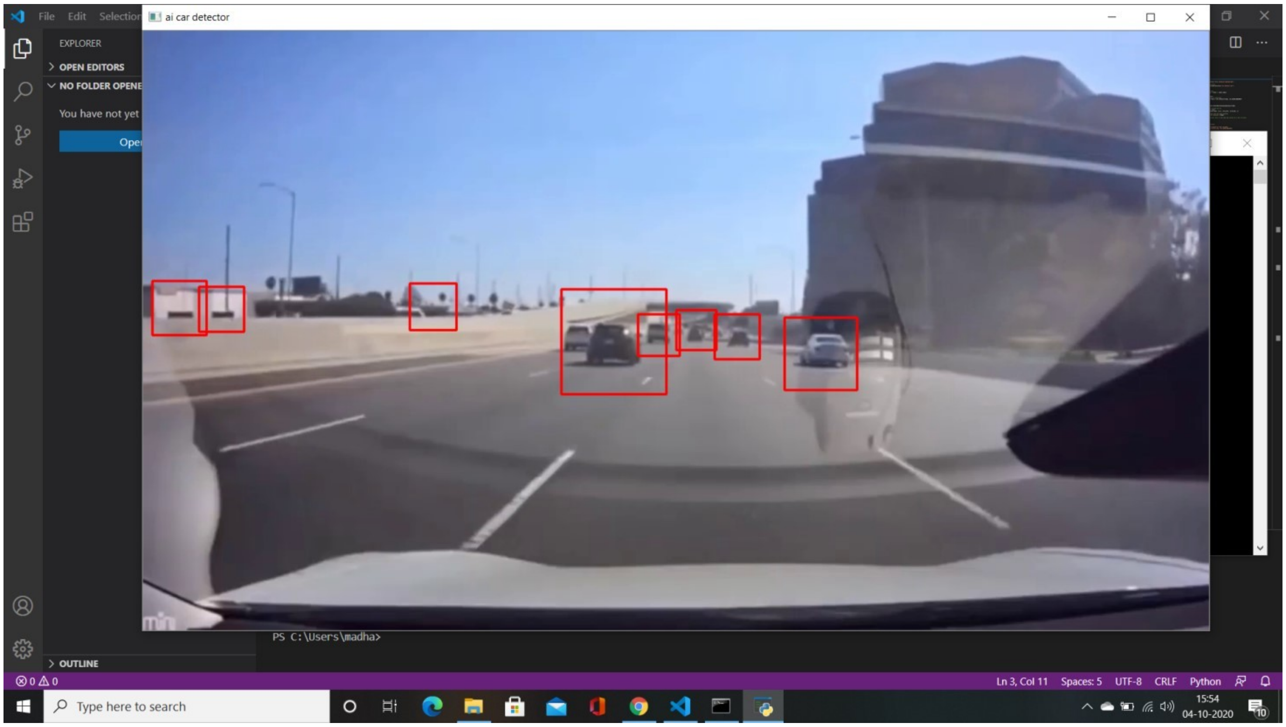


File Edit Selection View Go Run Terminal Help
ai car and pedestrian tracker.py - Visual Studio Code
EXPLORER
OPEN EDITORS
NO FOLDER OPENED
You have not yet opened a folder.
Open Folder
OUTLINE
Welcome
ai car and pedestrian tracker.py
car detector.xml
C: > Users > madha > .vscode > ai car and pedestrian tracker.py
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: powershell
Ln 3, Col 11 Spaces: 5 UTF-8 CRLF Python
Type here to search
15:53
04-10-2020

```
#pre-trained car classifier
car_tracker = cv2.CascadeClassifier('car detector.xml')

while True:

    #read current frame
    (read_successful, frame) = video.read()

    #safe coding
    if read_successful:
        #must convert to grayscale
        grayscale_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    else:
        break

    #detect cars
    cars = car_tracker.detectMultiScale(grayscale_frame)

    #draw rectangles around the car
    for (x,y,w,h) in cars:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,0,255), 2)

    #display the image with the faces spotted
    cv2.imshow('ai car detector',frame)

    #don't autoclose(wait here in the code and listen for a key to press)
    cv2.waitKey(1)
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\madha>
```

fig 9



```
PS C:\Users\madha>
```

Fig 10

REFERENCES:

[1] Viola, P & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR, 2001), December 8-14, 2001, Kauai, HI, USA.

[2] Liao, S., Jain, A.K., Li, S. Z. (2016). A fast and accurate unconstrained face detector. IEEE Transaction of Pattern Analysis and Machine Intelligence, Vol 38, No 2.

[3] Luo, D., Wen, G., Li, D., Hu, Y., and Huna, E. (2018). Deep learning-based face detection using iterative bounding-box regression. Multimedia Tools Applications.

[4] Zhang, Y., Wang, X., and Qu, B. (2012). Three-frame difference algorithm research based on mathematical morphology. Proceedings of 2012 International Workshop on Information and Electronics Engineering (IWIEE), pp. 2705 – 2709.

[5] Canny, J. (1986). A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume: PAMI-8, No: 6, pp. 679-698, November 1986.

[6] Li, J. and Ding, S. (2011). A research on improved Canny edge detection algorithm.

Proceedings of the International Conference on Applied Informatics and Communication, pp. 102 – 108, Communications in Computer and Information Science (CCIS), Vol 228, Springer-Verlag. [7 Lucas, B. D. & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. Proceedings of Imaging Understanding Workshop

[8] Ren, Z., Yang, S., Zou, F., Yang, F., Luan, C., and Li, K. (2017). A face tracking framework based on convolutional neural networks and Kalman filter. Proceedings of the 8th IEEE International Conference on Software Engineering and Services Science, pp. 410-413. [9] Mingxing, J., Junqiang, D., Tao, C., Ning, Y., Yi, J., and Zhen, Z. (2013). An improved detection algorithm of face with combining AdaBoost and SVM. Proceedings of the 25th Chinese Control and Decision Conference, pp. 2459-2463.

[10] Altun, H., Sinekli, R., Tekbas, U., Karakaya, F. and Peker, M. (2011). An efficient color detection in RGB space using hierarchical neural network structure. Proceedings of 2011 International Symposium on Innovations in Intelligent Systems and Applications, pp. 154-158, Istanbul, Turkey.