TRAFFIC CONTROL SYSTEM

A Course Project Report submitted to the

VNR VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY ,HYDERABAD

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING-CYBER SECURITY

Submitted By
CH.VARUN TEJA-21071A6213
K.SAI NIKHIL-21071A6225

Under the Guidance of
MRS. E. LALITHA

TAMASOMA JYOTIRGAMAYA
Estd. 1995

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade (CGPA: 3.73/4.0)
NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes
Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as “College with Potential for Excellence” by UGC
VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India
2023

VALLURUPALLI NAGESWARA RAO VIGNANAJYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with ‘A++’ Grade (CGPA: 3.73/4.0)NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B.Tech. Programmes Approved by AICTE, New Delhi, Affiliated to JNTU-H, Recognised as “College with Potential for Excellence” by UGCVignanaJyothi Nagar,
Pragathi Nagar, Nizampet (S.O), Hyderabad TS 500 090 India

CERTIFICATE

This is to certify that **CH.VARUN TEJA(21071A6213), K.SAI NIKHIL (21071A6225)** have successfully completed their Course Based Project work at **Computer Science & Engineering Department** of **Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology,** Hyderabad entitled **“TRAFFIC CONTROL SYSTEM ”** in partial fulfillment of the requirements for the award of **B.Tech** during the academic year 2022-2023**.**

This work is carried out under my supervision and has not been submitted to any otherUniversity/ Institute for award of any degree/ diploma.

DECLARATION

This is to certify that the project work entitled **“TRAFFIC CONTROL SYSTEM”** submitted in VNR Vignana Jyothi Institute of Engineering & Technology in partial fulfillment of requirement for the award of Bachelor ofTechnology in Computer Science and Engineering is a bonafide report of the work carried out by us under the guidance and supervision of Mrs.E. LALITHA, AssistantProfessor, Department of CYS,DS,AND AI&DS VNRVJIET. To the best of our knowledge, this report has not been submitted in any form to any university or institution for the award of any degree or diploma.

ACKNOWLEDGEMENT

An endeavor over a long period can be successful only with the advice and support of many well-wishers. We take this opportunity to express our gratitude and appreciation to all of them.

First of all we thank the lord almighty who has been with us from the beginning tothe end of our project. We are indebted to our venerable principal **Dr. C. D. Naidu** for thisunflinching devotion, which led us to complete this project. The support, encouragement given by him and his motivation lead us to complete this project.

We wish to express our profound gratitude to **Dr M.RAJASEKHAR HOD OF CSE-CYS DS AND AI &DS** Department**, VNR Vignana Jyothi Institute of Engineering and Technology** for their constant and dedicated service to brighten our career.

With great pleasure we express our gratitude to the internal guide **Mrs.E.LALITHA** for his timely help, constant guidance, cooperation, support and encouragement throughout this project.

Finally we wish to express our deep sense of gratitude and sincere thanks to our parents, friends and all our well-wishers who have technically and non-technically contributed for the successful completion of our course based project.

ABSTRACT

Traffic light control systems are widely used to monitor and control the flow of automobiles through the junction of many roads. They aim to realize smooth motion of cars in the transportation routes. However, the synchronization of multiple traffic light systems at adjacent intersections is a complicated problem given the various parameters involved. Conventional systems do not handle variable flows approaching the junctions. In addition, the mutual interference between adjacent traffic light systems, the disparity of cars flow with time, the accidents, the passage of emergency vehicles, and the pedestrian crossing are not implemented in the existing traffic system. This leads to traffic jam and congestion. We propose a system based on PIC microcontroller that evaluates the traffic density using IR sensors and accomplishes dynamic timing slots with different levels. Moreover, a portable controller device is designed to solve the problem of emergency vehicles stuck in the overcrowded road.

CHAPTER 1

INTRODUCTION

Now a days ,we want technology to fast ,our everyday today activities to be completed fast but when we come to traffic .We know how much time it spend in traffic ,a lot of time .Here is a project which we have done on traffic control system to reduce the time and spend our previous time on good.

About our project

We programmed a traffic control system at a junctions which has four different paths ,we assumed it as north ,east, south and west. In this program we are checking the direction in which we have more vehicles and the traffic signal will allow the vehicles to go and if there is to much traffic in outway then the traffic signal will not allow vehicles to go that side as it will increase the traffic .We have designed a program which checks the density of vehicles which can be seen in red rectangles . We have included the timer also ,so that the traffic signal changes. If our program did not detect any vehicles the signal will be RED and it will move to the next direction and perform the same.

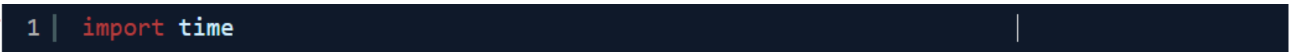
CHAPTER 2

SOFTWARE REQUIREMENT ANALYSIS

The Python Time Module

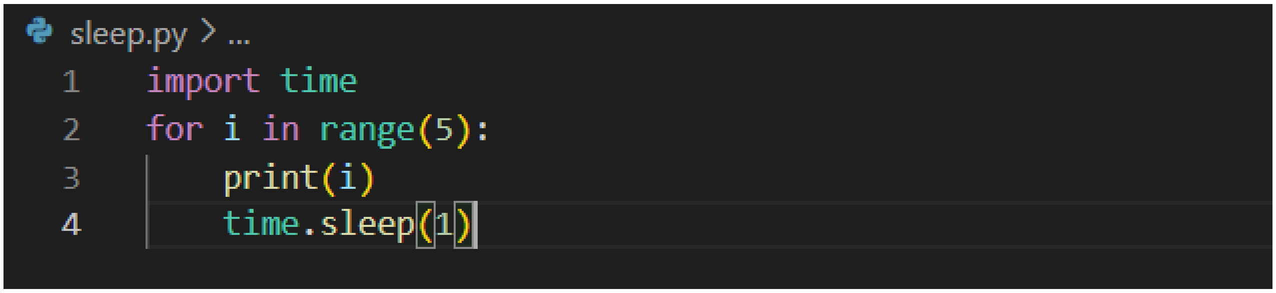
One of the 200 modules in Python's standard library is the **time module**. This module contains the functions we'll need to build a simple timer in Python.

To use the time module in Python, we first import it into our program:



```python
import time
```

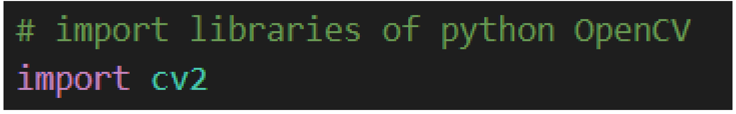
- **The time.sleep() function:**
  We call the `time()` function, located in the *time* module, as `time.sleep()`. The first `time` references the module, whereas the second is the function itself. The `sleep()` function makes the execution of timed loops possible.



```python
import time
for i in range(5):
    print(i)
    time.sleep(1)
```

The Python OpenCV Module

To use the time module in Python, we first import it into our program:



```python
# import libraries of python OpenCV
import cv2
```

CHAPTER 3

Hardware Requirements:

Processor: Intel core i3 3740

OS: Windows(Preferred, but any OS that can run python)

Python version: Any version above 2.9x

Minimum Ram: 4GB

Camera: HD(not required if processing is done internall

CHAPTER 4

IMPLEMENTATION

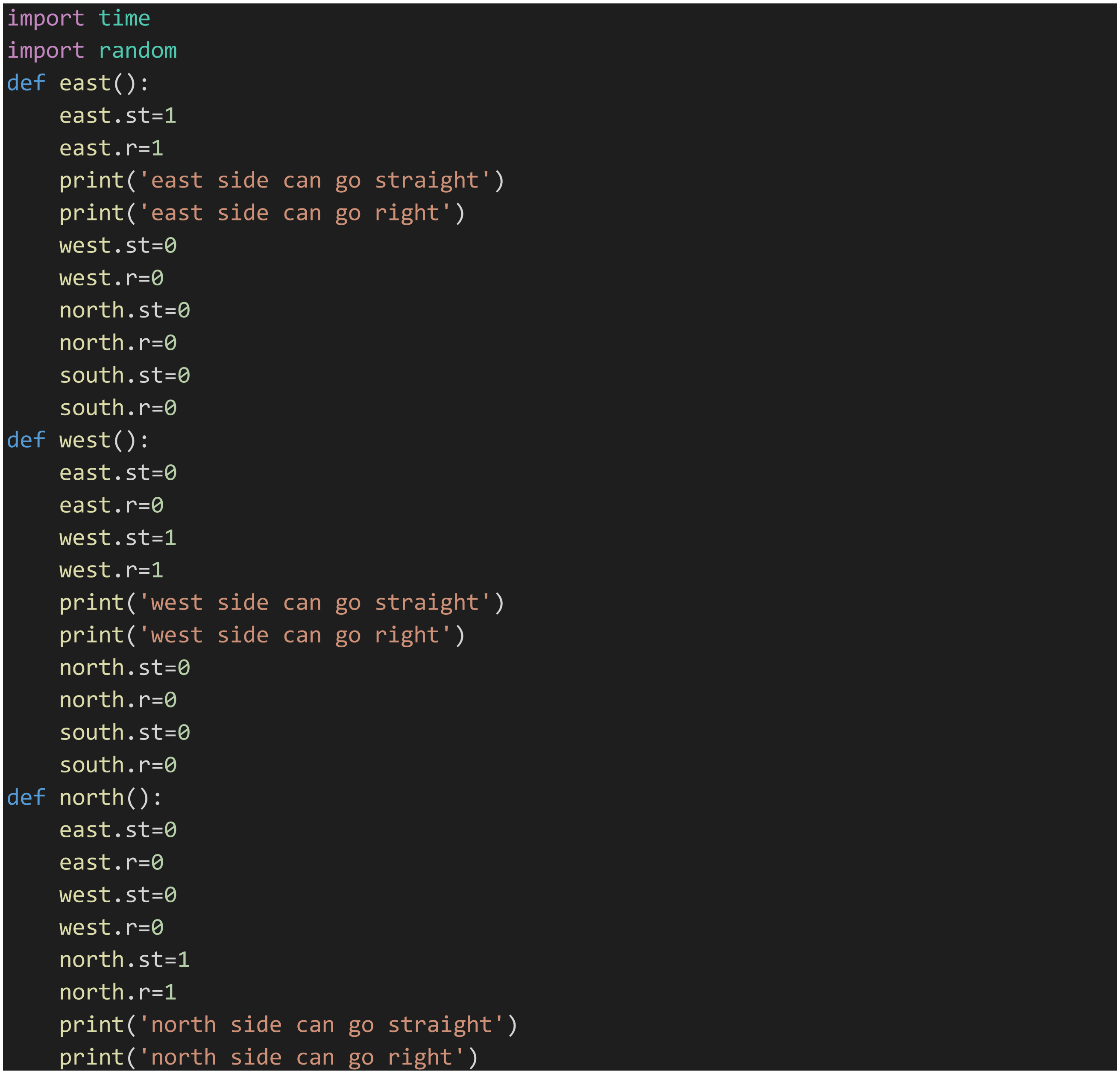
Vehicle Detection

```
import cv2

# capture frames from a video
cap = cv2.VideoCapture('video.avi')

car_cascade = cv2.CascadeClassifier('cars.xml')

# loop runs if capturing has been initialized.
while True:
    # reads frames from a video
    ret, frames = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)


    # Detects cars of different sizes in the input image
    cars = car_cascade.detectMultiScale(gray, 1.1, 1)

    for (x,y,w,h) in cars:
        cv2.rectangle(frames,(x,y),(x+w,y+h),(0,0,255),2)

    cv2.imshow('video2', frames)
     #opens the camera window for 30 seconds or until esc is pressed.
    if cv2.waitKey(33) == 27:
        break

cv2.destroyAllWindows()
```

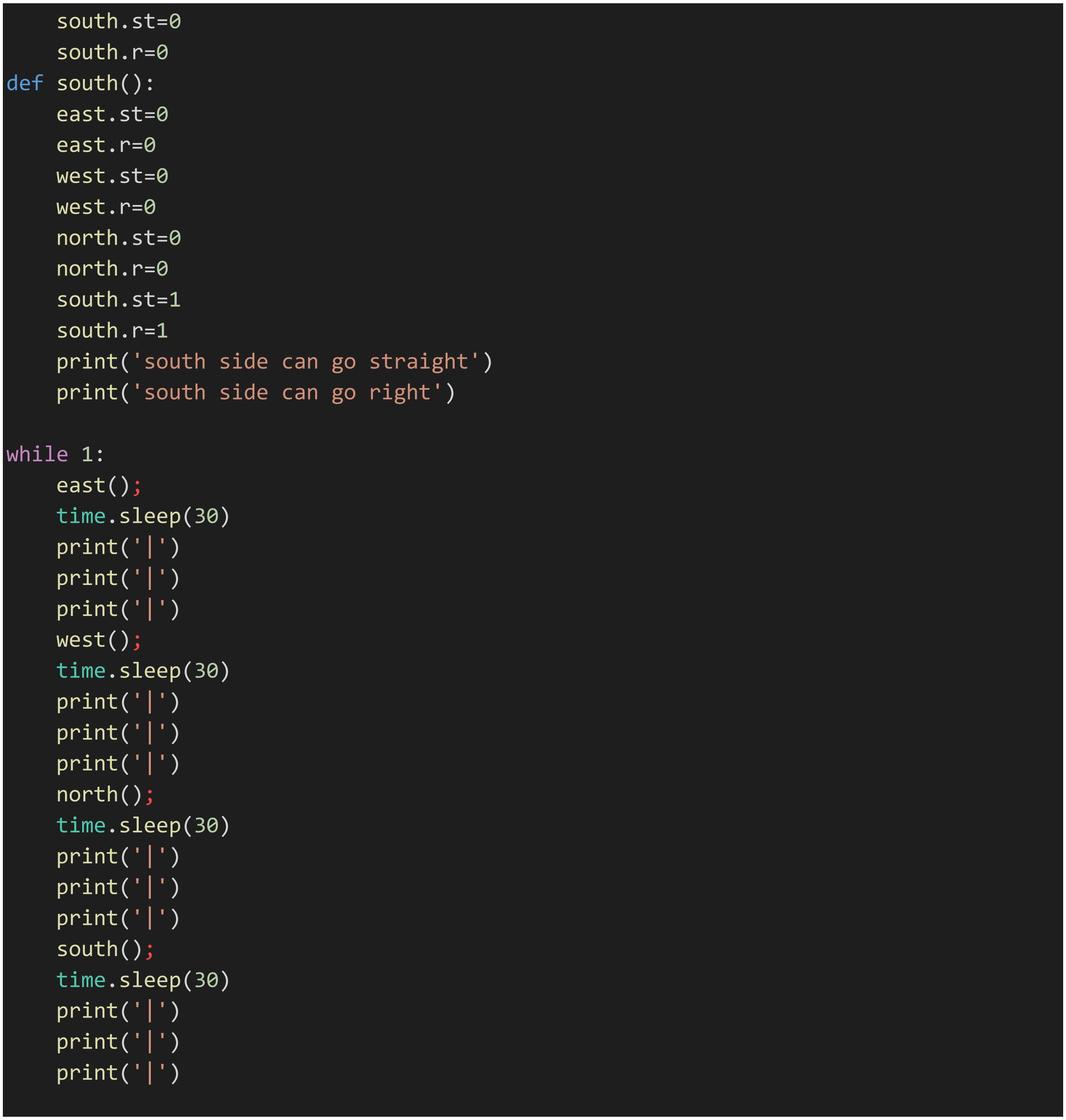
TRAFFIC CONTROL:

In this section, we will show how the pattern used for traffic signals. This is a simple task.We do not need any module for execution. The output is shown in the command prompt due to lack of hardware resources. We have estimated a 4-way junction as East,West,North,South.

This is the code used for traffic signals.



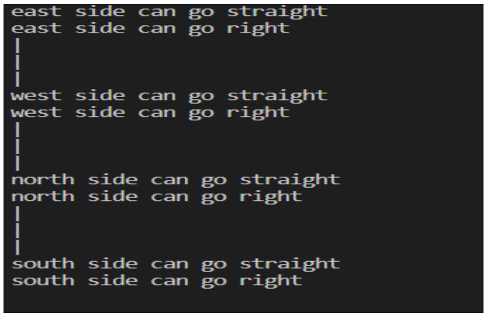
```
import time
import random
def east():
    east.st=1
    east.r=1
    print('east side can go straight')
    print('east side can go right')
    west.st=0
    west.r=0
    north.st=0
    north.r=0
    south.st=0
    south.r=0
def west():
    east.st=0
    east.r=0
    west.st=1
    west.r=1
    print('west side can go straight')
    print('west side can go right')
    north.st=0
    north.r=0
    south.st=0
    south.r=0
def north():
    east.st=0
    east.r=0
    west.st=0
    west.r=0
    north.st=1
    north.r=1
    print('north side can go straight')
    print('north side can go right')
```



```
    south.st=0
    south.r=0
def south():
    east.st=0
    east.r=0
    west.st=0
    west.r=0
    north.st=0
    north.r=0
    south.st=1
    south.r=1
    print('south side can go straight')
    print('south side can go right')

while 1:
    east();
    time.sleep(30)
    print('|')
    print('|')
    print('|')
    west();
    time.sleep(30)
    print('|')
    print('|')
    print('|')
    north();
    time.sleep(30)
    print('|')
    print('|')
    print('|')
    south();
    time.sleep(30)
    print('|')
    print('|')
    print('|')
```

Output:

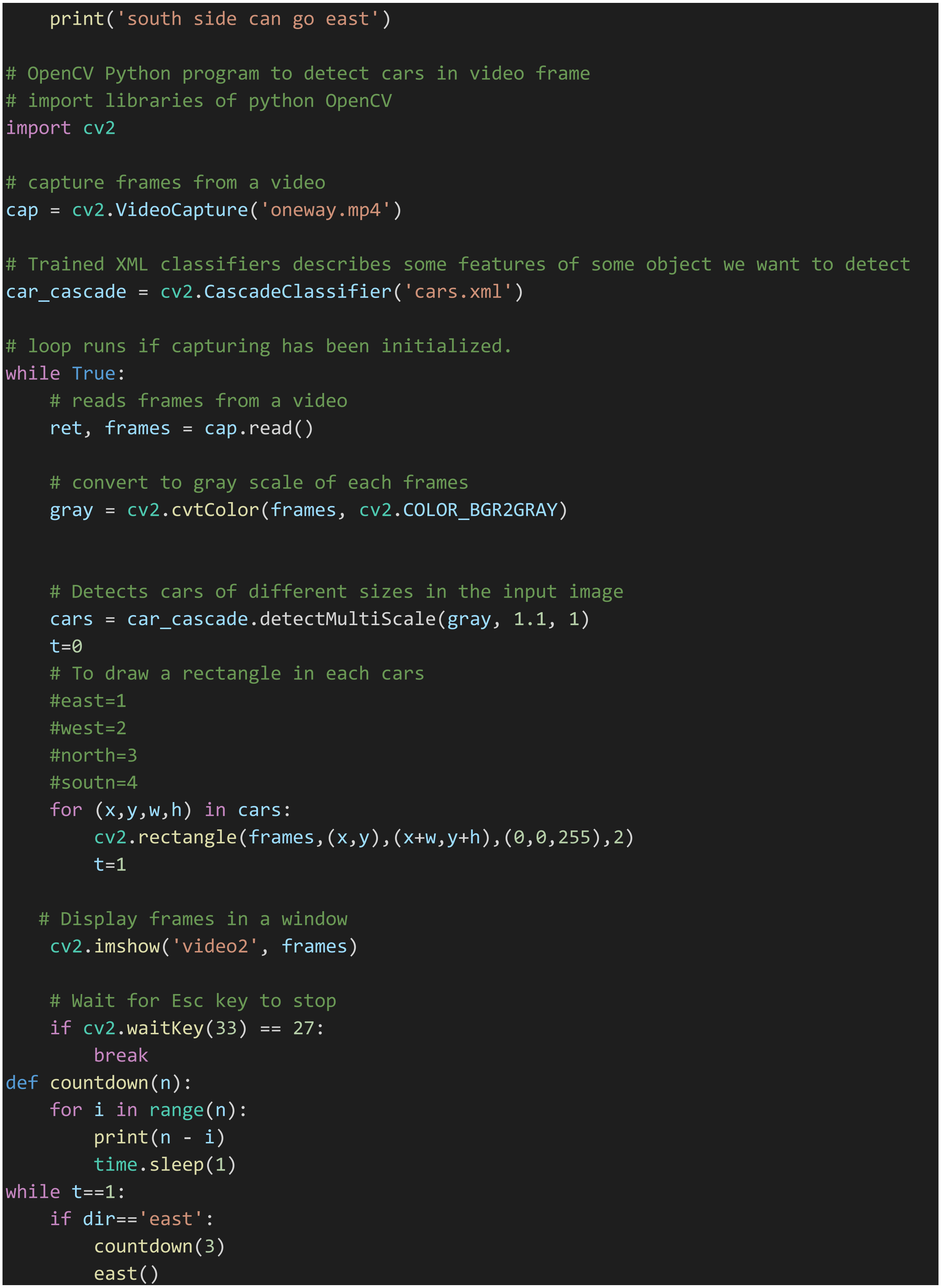
The loop is repeated indefinitely until stopped.



```
east side can go straight
east side can go right
|
|
|
west side can go straight
west side can go right
|
|
|
north side can go straight
north side can go right
|
|
|
south side can go straight
south side can go right
```

COMPLETE CODE:

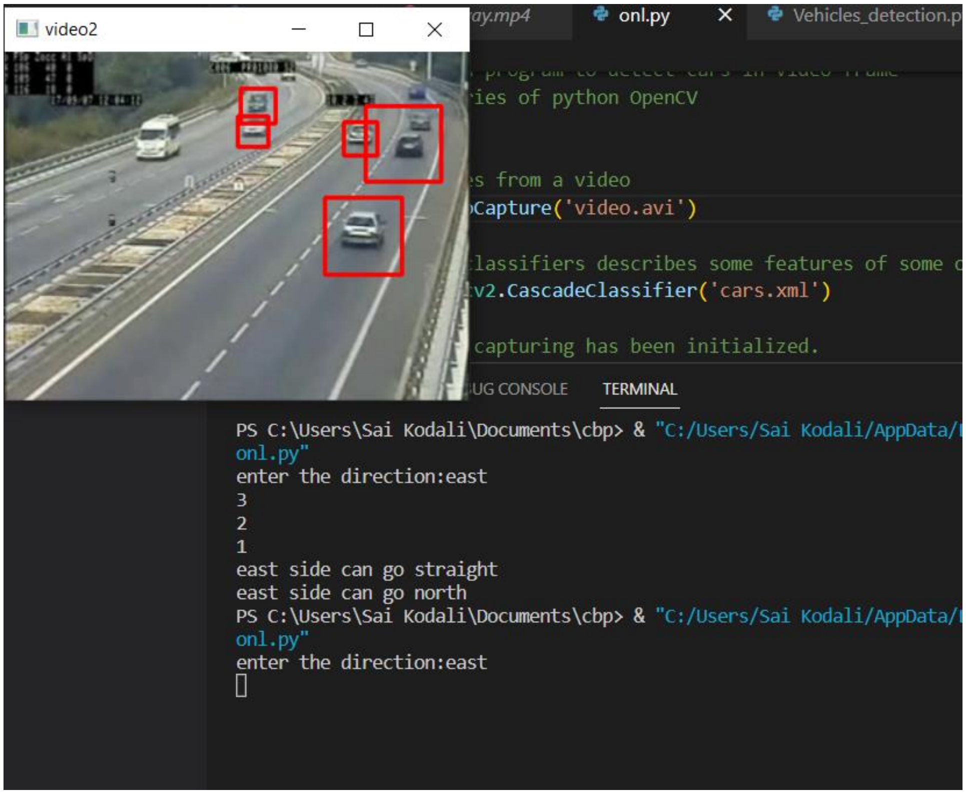
```
import time
dir=input("enter the direction:")
def east():
    east.st=1
    east.r=1
    print('east side can go straight')
    print('east side can go north')
    west.st=0
    west.r=0
    north.st=0
    north.r=0
    south.st=0
    south.r=0
def west():
    east.st=0
    east.r=0
    west.st=1
    west.r=1
    print('west side can go straight')
    print('west side can go south')
    north.st=0
    north.r=0
    south.st=0
    south.r=0
def north():
    east.st=0
    east.r=0
    west.st=0
    west.r=0
    north.st=1
    north.r=1
    print('north side can go straight')
    print('north side can go west')
    south.st=0
    south.r=0
def south():
    east.st=0
    east.r=0
    west.st=0
    west.r=0
    north.st=0
    north.r=0
    south.st=1
    south.r=1
    print('south side can go straight')
```



```
    print('south side can go east')

# OpenCV Python program to detect cars in video frame
# import libraries of python OpenCV
import cv2

# capture frames from a video
cap = cv2.VideoCapture('oneway.mp4')

# Trained XML classifiers describes some features of some object we want to detect
car_cascade = cv2.CascadeClassifier('cars.xml')

# loop runs if capturing has been initialized.
while True:
    # reads frames from a video
    ret, frames = cap.read()

    # convert to gray scale of each frames
    gray = cv2.cvtColor(frames, cv2.COLOR_BGR2GRAY)


    # Detects cars of different sizes in the input image
    cars = car_cascade.detectMultiScale(gray, 1.1, 1)
    t=0
    # To draw a rectangle in each cars
    #east=1
    #west=2
    #north=3
    #soutn=4
    for (x,y,w,h) in cars:
        cv2.rectangle(frames,(x,y),(x+w,y+h),(0,0,255),2)
        t=1

   # Display frames in a window
    cv2.imshow('video2', frames)

    # Wait for Esc key to stop
    if cv2.waitKey(33) == 27:
        break
def countdown(n):
    for i in range(n):
        print(n - i)
        time.sleep(1)
while t==1:
    if dir=='east':
        countdown(3)
        east()
        time.sleep(3)
        break;
    if dir=='west':
        countdown(3)
        west()
        time.sleep(3)
        break;
    if dir=='north':
        countdown(3)
        north()
        time.sleep(3)
        break;
    if dir=='south':
        countdown(3)
        south()
        time.sleep(3)
        break;
# De-allocate any associated memory usage
cv2.destroyAllWindows()
```

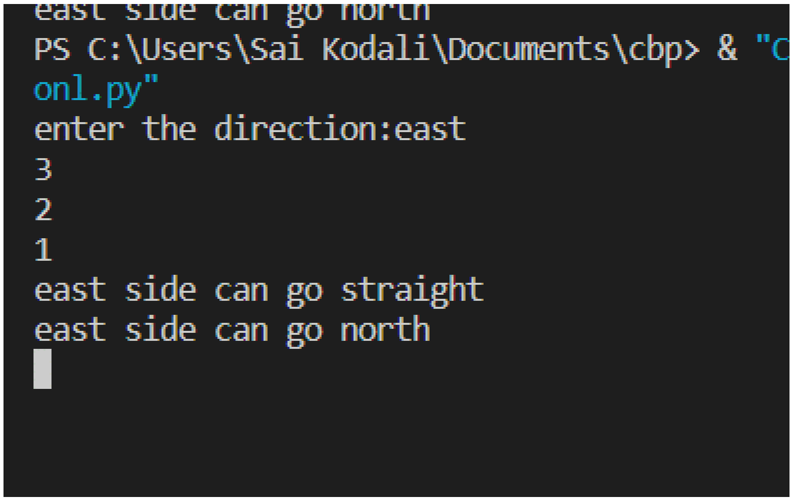
OUTPUT:

1. When Vehicles are detected.



```
ies of python OpenCV

s from a video
Capture('video.avi')

lassifiers describes some features of some c
v2.CascadeClassifier('cars.xml')

capturing has been initialized.
```

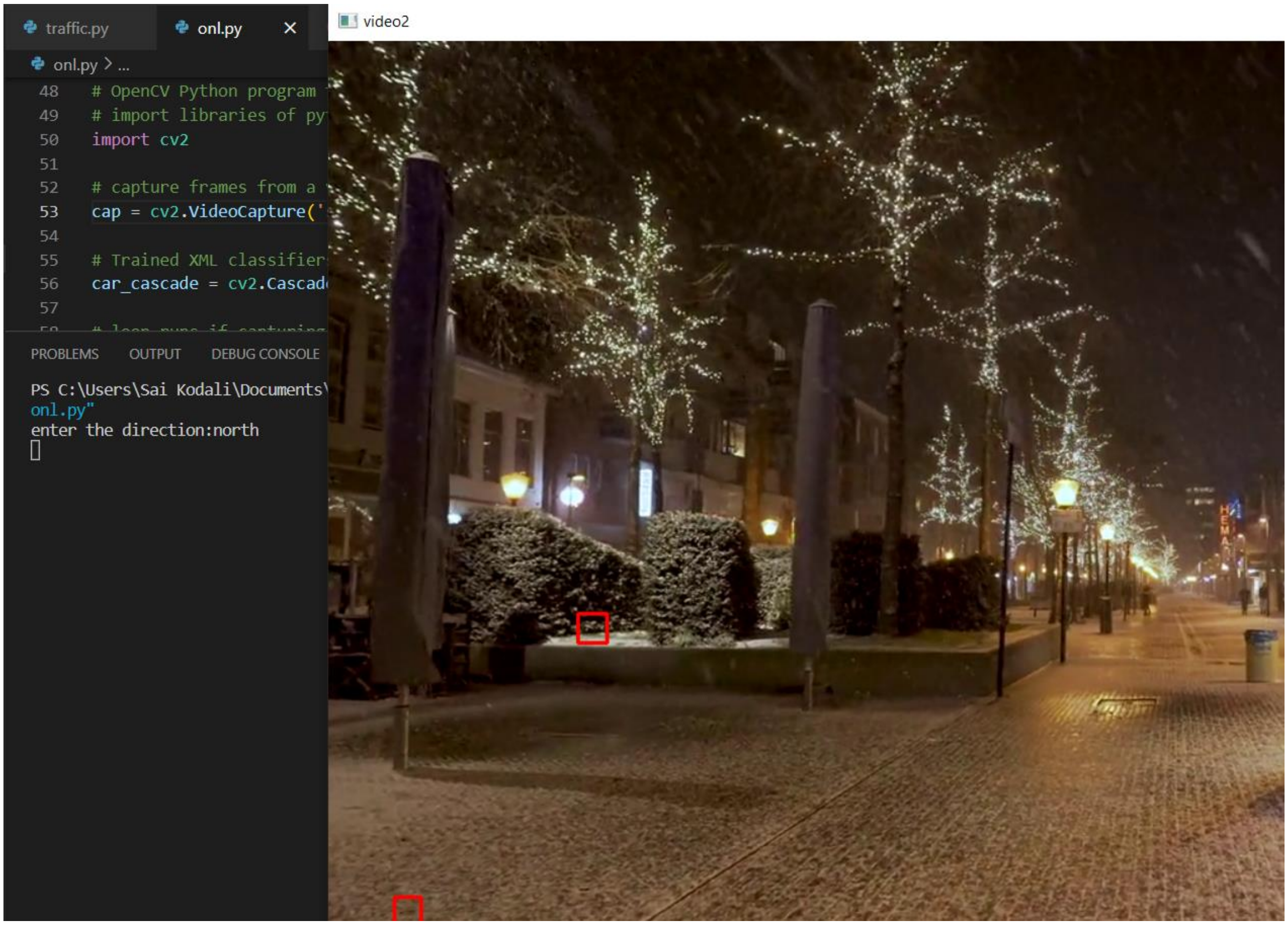
```
PS C:\Users\Sai Kodali\Documents\cbp> & "C:/Users/Sai Kodali/AppData/
onl.py"
enter the direction:east
3
2
1
east side can go straight
east side can go north
PS C:\Users\Sai Kodali\Documents\cbp> & "C:/Users/Sai Kodali/AppData/
onl.py"
enter the direction:east
▯
```



```
east side can go north
PS C:\Users\Sai Kodali\Documents\cbp> & "C
onl.py"
enter the direction:east
3
2
1
east side can go straight
east side can go north
```

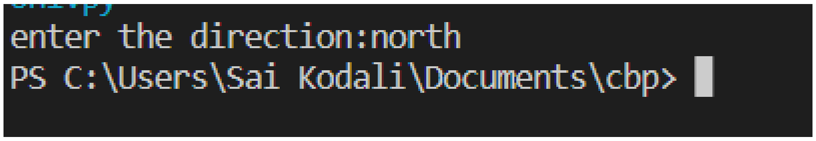
Note: The output only shows up if the program detects vehicles

When vehicles are not detected.



```
# OpenCV Python program
# import libraries of py
import cv2

# capture frames from a
cap = cv2.VideoCapture('

# Trained XML classifier
car_cascade = cv2.Cascad

# loop runs if capturing
```

```
PS C:\Users\Sai Kodali\Documents\
onl.py"
enter the direction:north
```



```
onl.py"
enter the direction:north
PS C:\Users\Sai Kodali\Documents\cbp>
```

As we can observe, we did not get any output, which means that the signal continues to be in the same state.

CONCLUSION:

Python Countdown Timer can be utilized to wait for a certain duration of time in the idle state before reiterating the same piece of code in the loop again as required. Python's time library contains a predefined sleep() function.

Not to mention, Python's opencv module can be used for many things, this module has a lot of potential and many pattern recognition capabilities. This module can detect any design or any other pattern like a human face for example, this is used in face recognition software. With in development in AI, these modules will become self-sufficient.