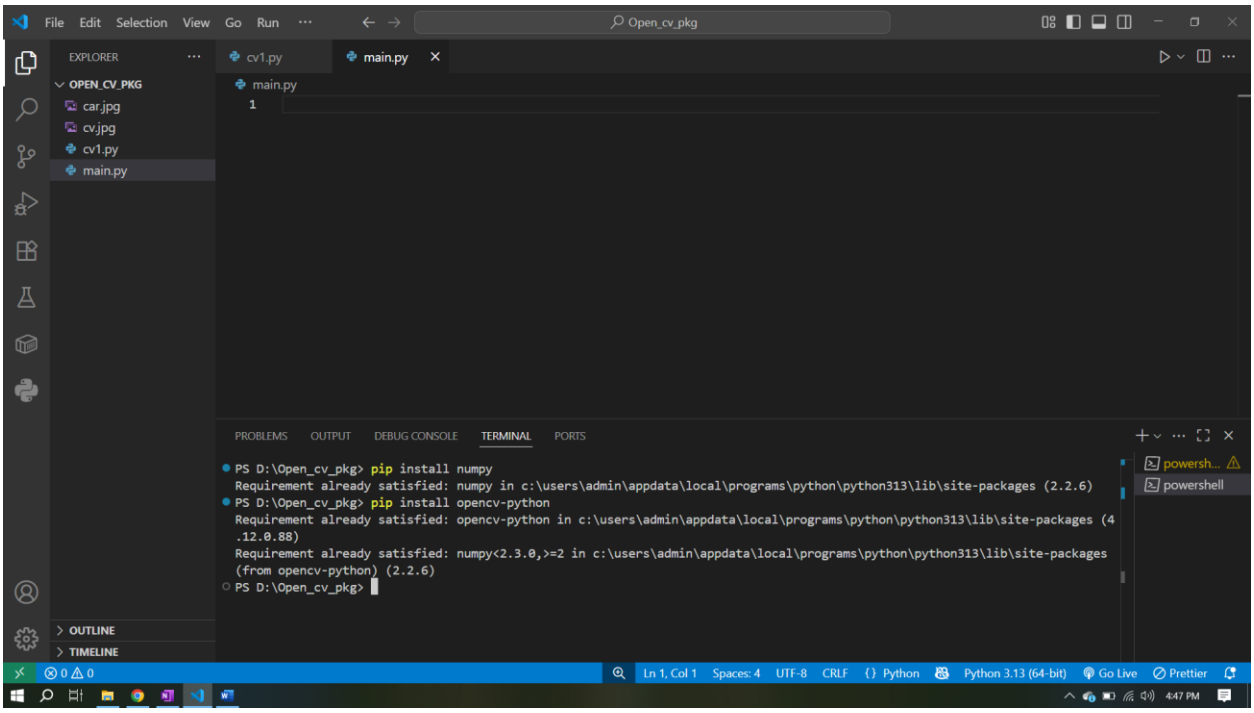
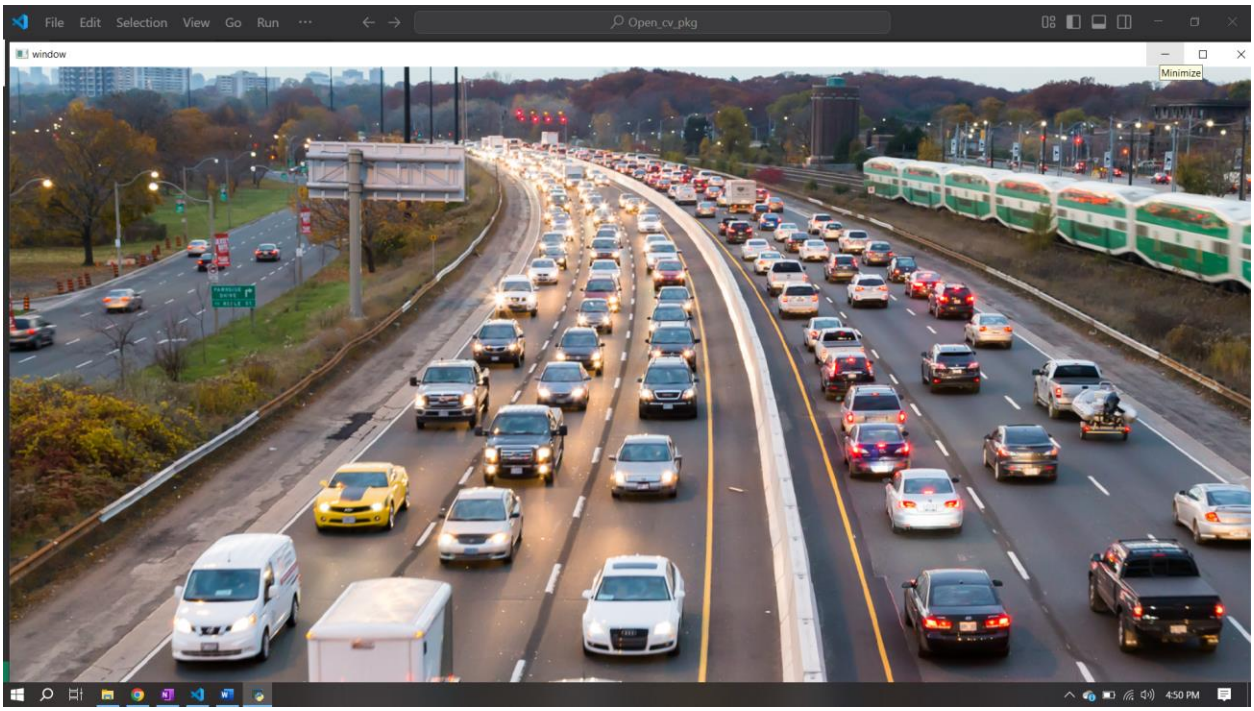
# Assignment-14

- **Installing open-cv:-**

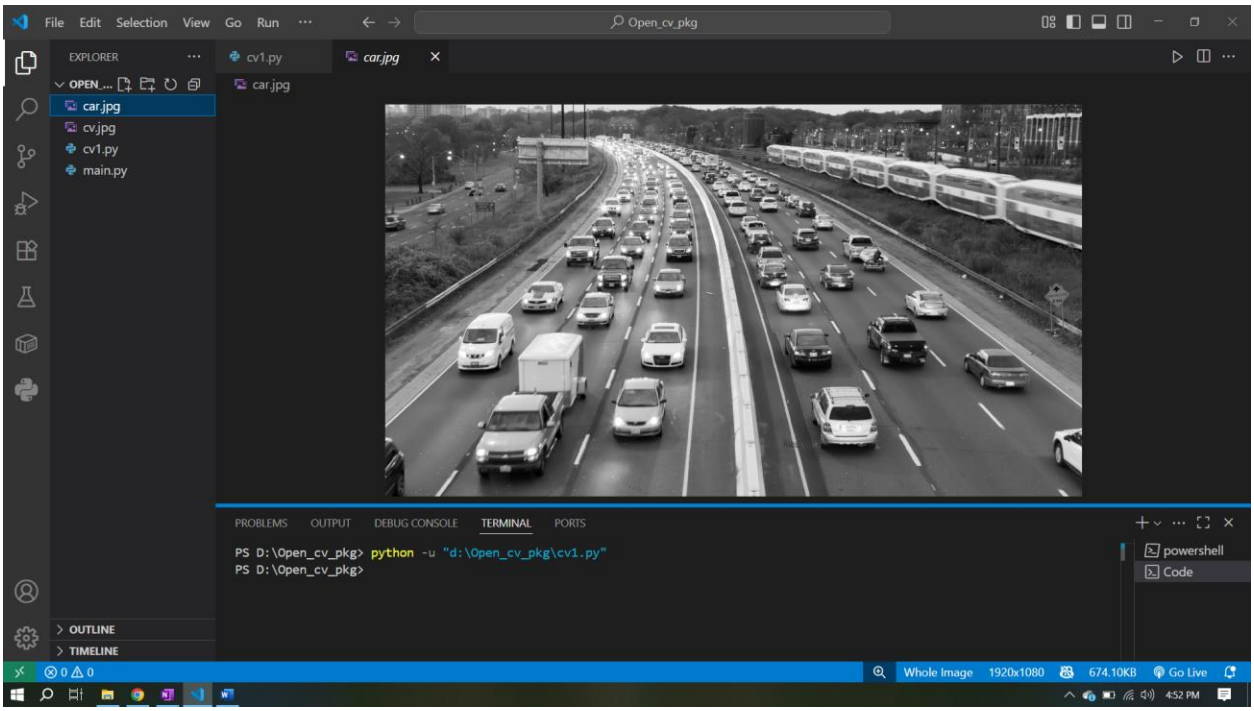



**Reading and writing image:-**

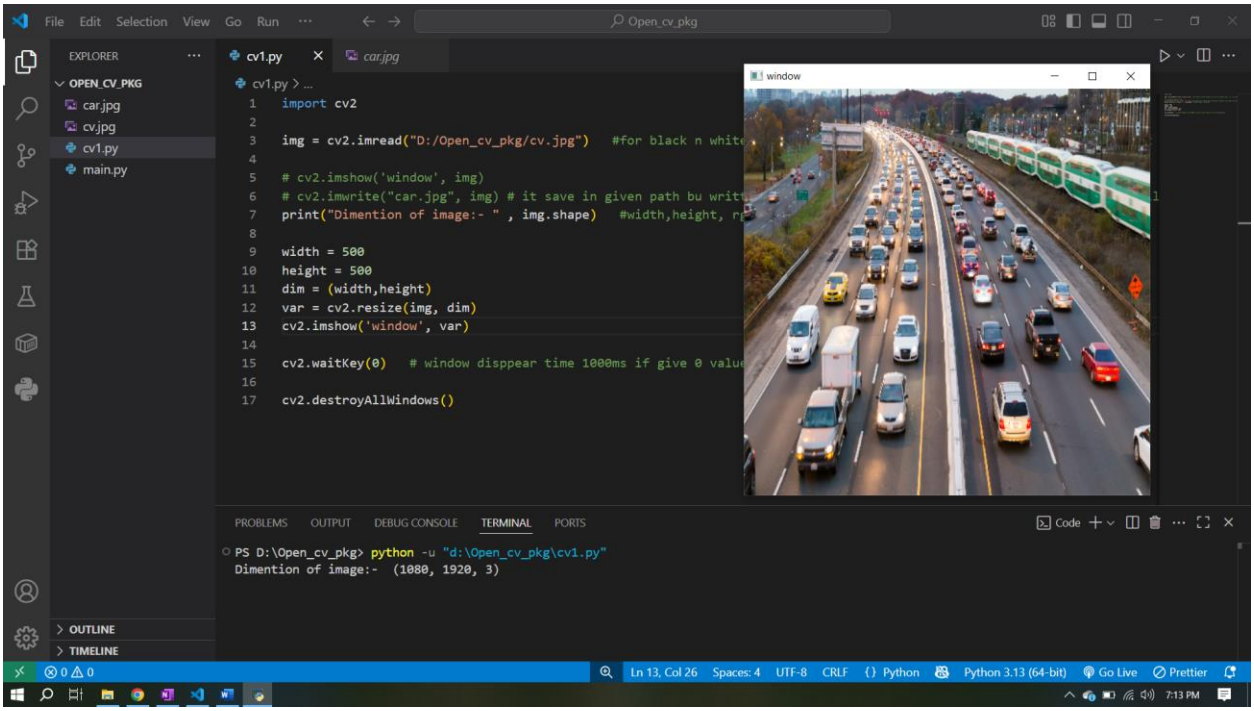
```
import cv2

img = cv2.imread("D:/Open_cv_pkg/cv.jpg",0)   #for black n white img add = 0 zero for original img = 1.-1. its default

cv2.imshow('window', img)
cv2.imwrite("car.jpg", img) # it saved in given path but writting img,can mention filename / path where we saved file

cv2.waitKey(0)   # window disappear time 1000ms if give 0 value window not close automatically

cv2.destroyAllWindows()
```

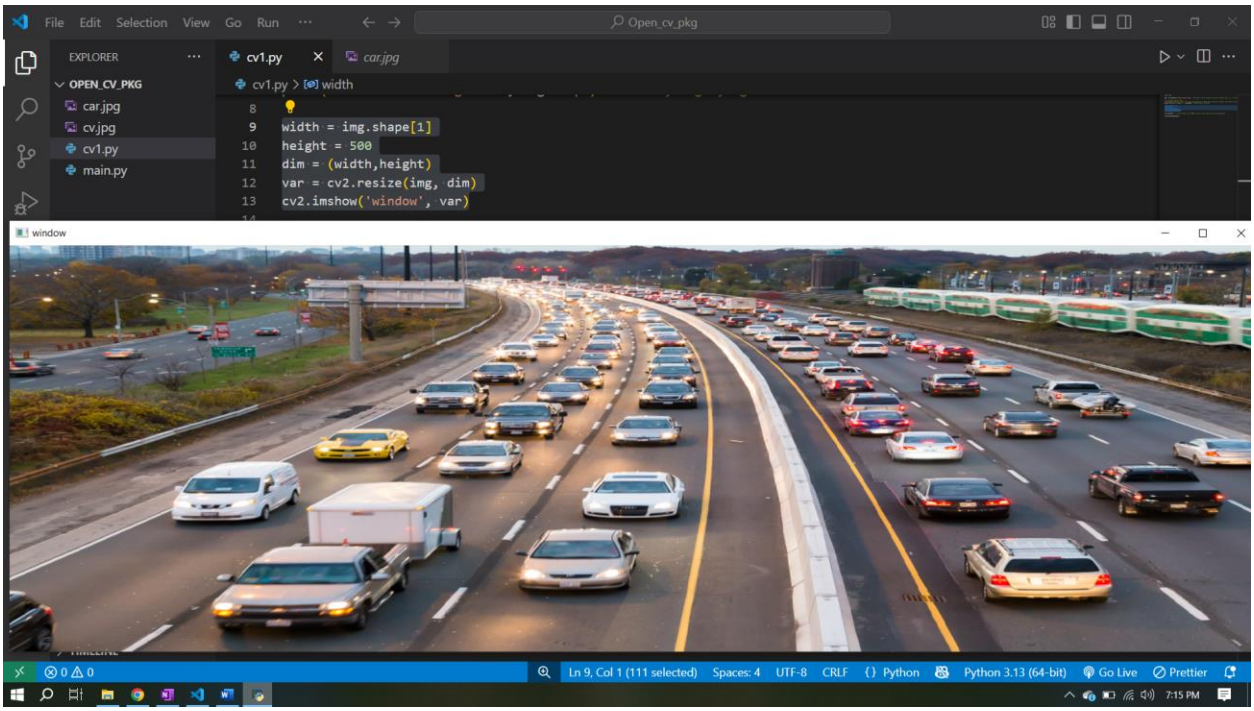

File Edit Selection View Go Run
Open_cv_pkg
window
Minimize
4:50 PM


File Edit Selection View Go Run
Open_cv_pkg
window
4:51 PM




## Resizing an image:

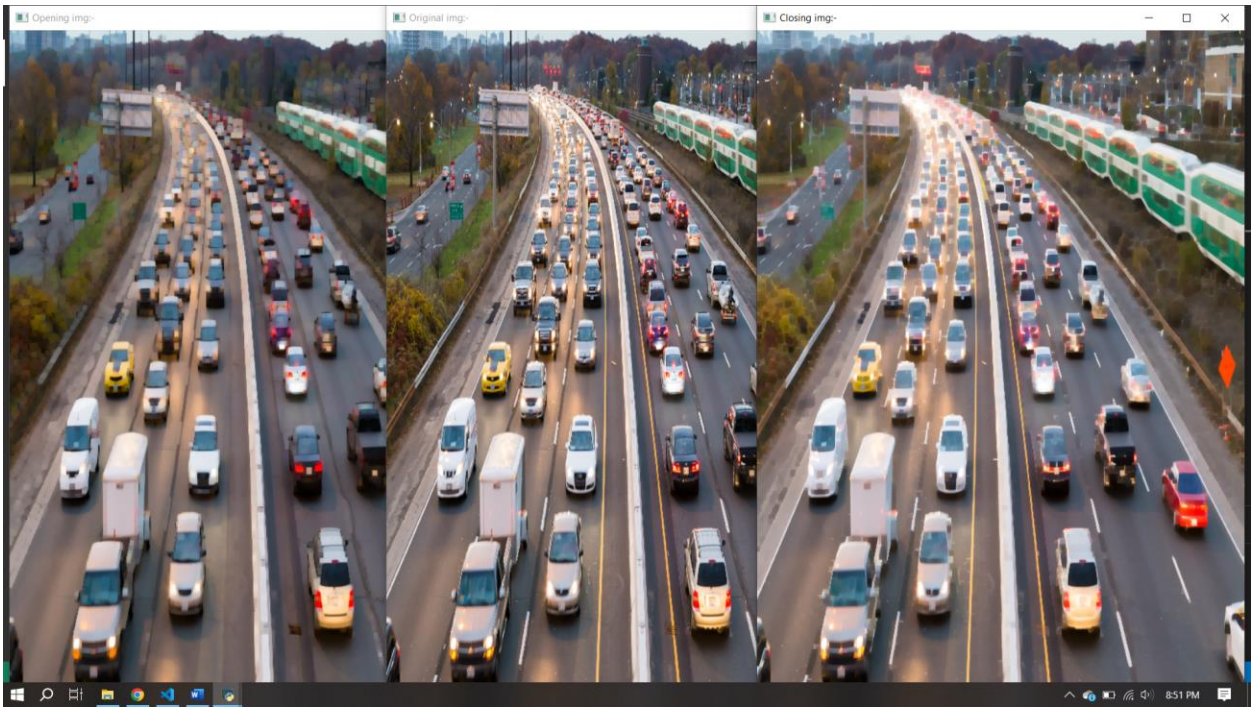




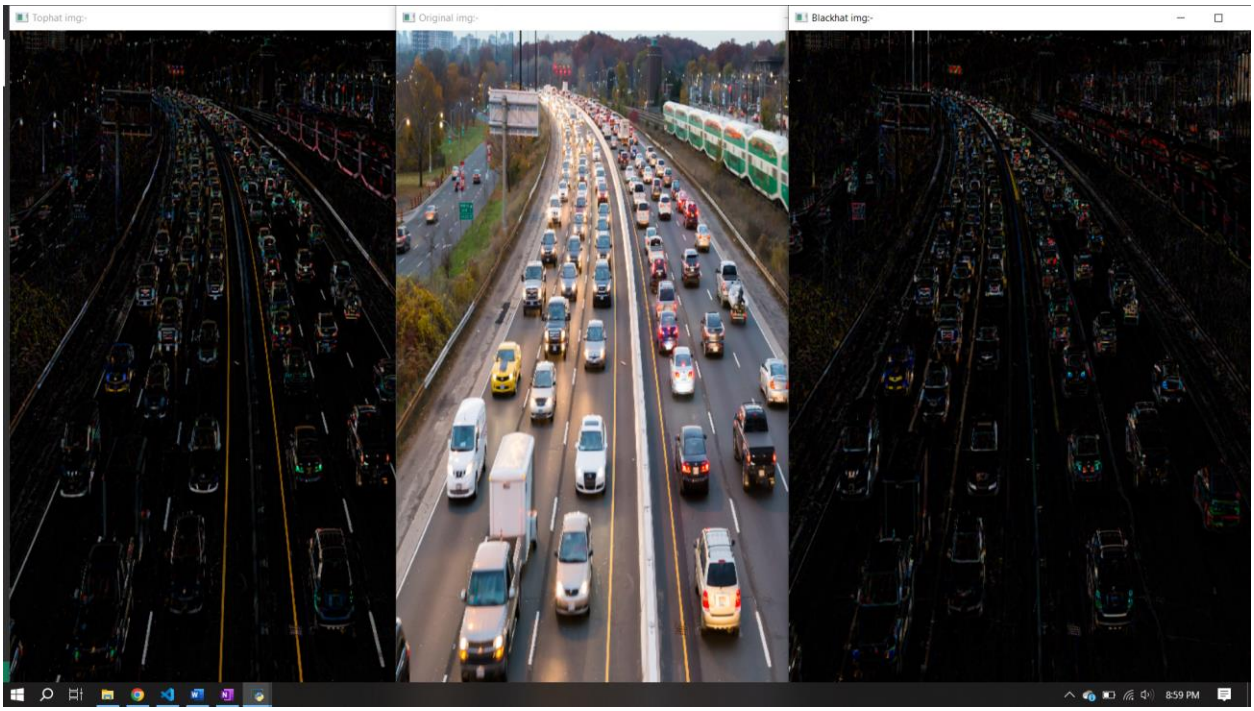

**Morphological Operations:**

```
import cv2
import numpy as np

img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")
width = 600
height = 850
dim = (width,height)
var = cv2.resize(img, dim)

kernel = np.ones((5,5), dtype='uint8')
erosion = cv2.erode(var, kernel, iterations=1)
dilation = cv2.dilate(var, kernel, iterations=1)
opening = cv2.morphologyEx(var, cv2.MORPH_OPEN, kernel)
closing = cv2.morphologyEx(var, cv2.MORPH_CLOSE, kernel)
gradient = cv2.morphologyEx(var, cv2.MORPH_GRADIENT, kernel)
tophat = cv2.morphologyEx(var, cv2.MORPH_TOPHAT, kernel)
blackhat = cv2.morphologyEx(var, cv2.MORPH_BLACKHAT, kernel)

cv2.imshow('Original img:- ', var)
cv2.imshow('Erosion img:- ', erosion)
cv2.imshow('Dilation img:- ', dilation)
cv2.imshow('Opening img:- ', opening)
cv2.imshow('Closing img:- ', closing)
cv2.imshow('Gradient img:- ', gradient)
```

```
cv2.imshow('Tophat img:- ', tophat)
cv2.imshow('Blackhat img:- ', blackhat)

cv2.waitKey(0)

cv2.destroyAllWindows()
```


Opening img:-
Original img:-
Closing img:-
8:51 PM

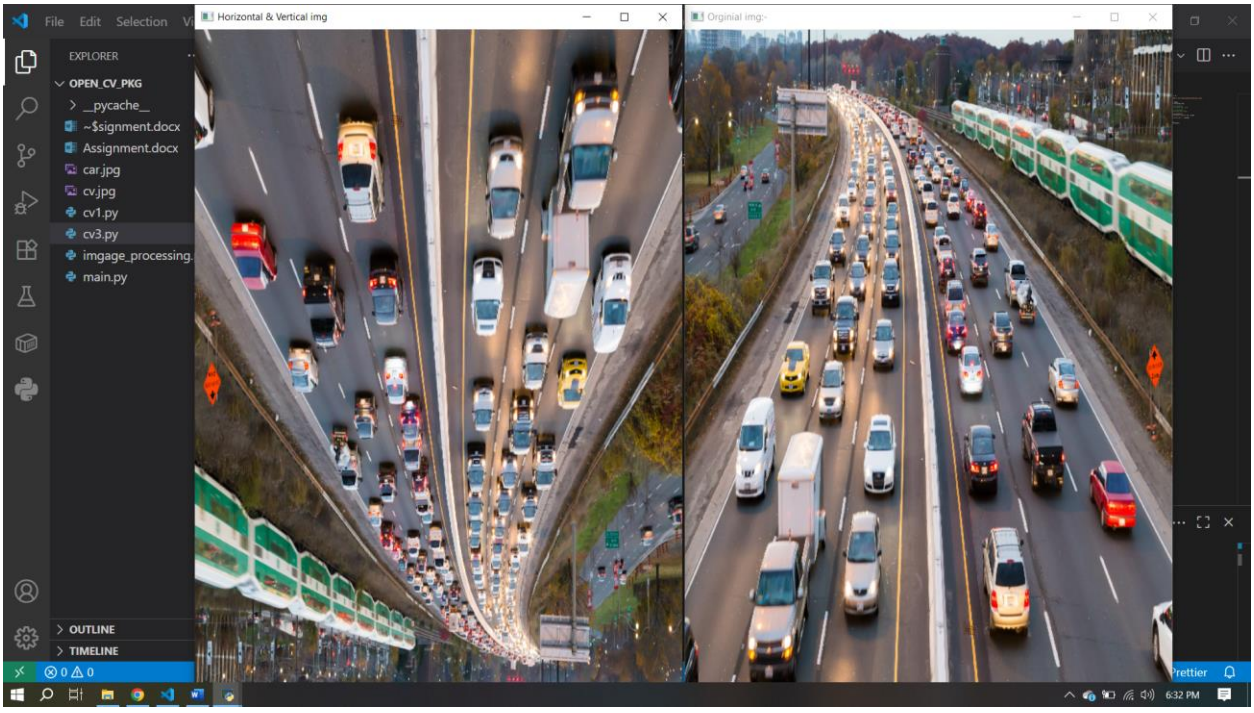

File
Gradient img:-
Original img:-
Code
Go Live
Prettier
8:56 PM

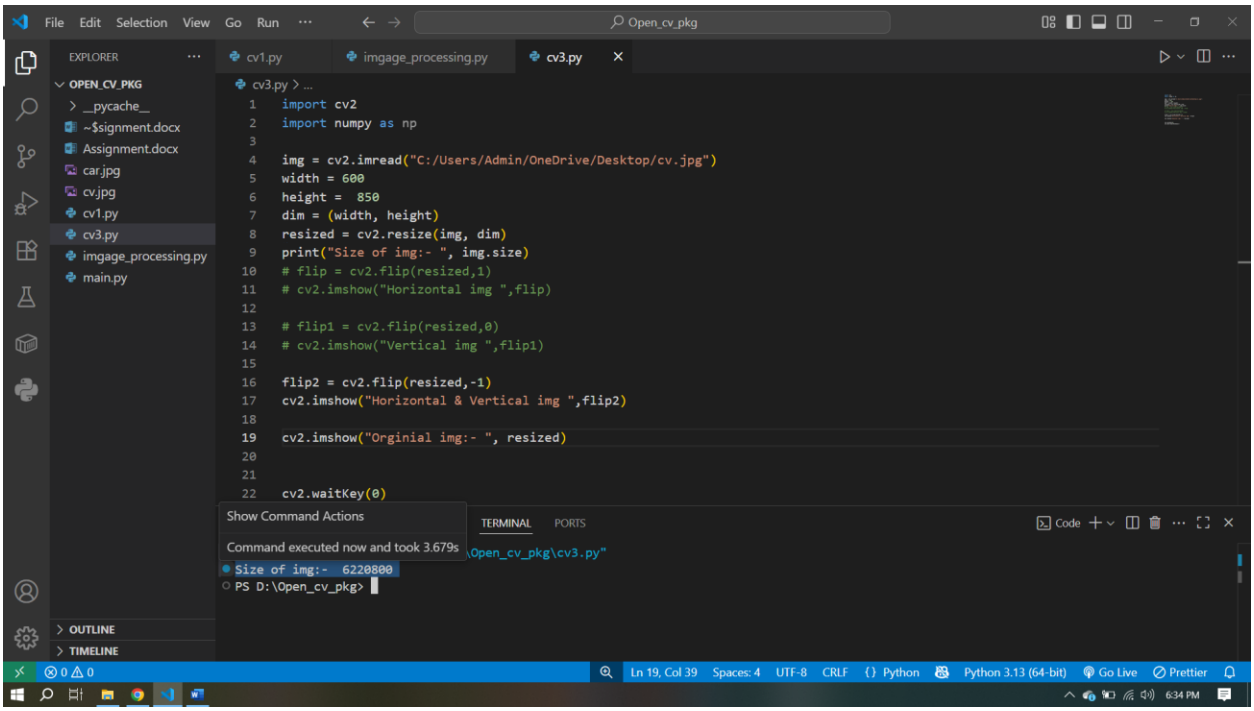



**Flipping an image:-**

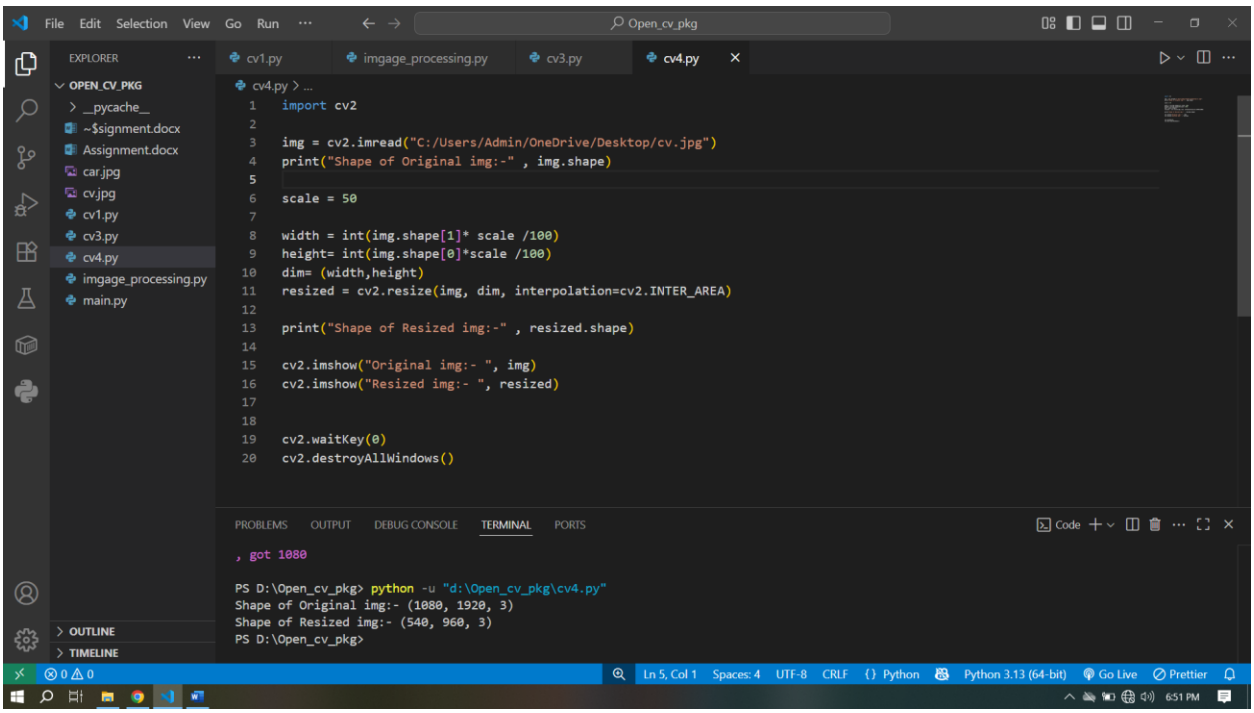
```
import cv2
import numpy as np

img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")
width = 600
height =  850
dim = (width, height)
resized = cv2.resize(img, dim)
print("Size of img:- ", img.size)
# flip = cv2.flip(resized,1)
# cv2.imshow("Horizontal img ",flip)

# flip1 = cv2.flip(resized,0)
# cv2.imshow("Vertical img ",flip1)

flip2 = cv2.flip(resized,-1)
cv2.imshow("Horizontal & Vertical img ",flip2)

cv2.imshow("Orginial img:- ", resized)


cv2.waitKey(0)
cv2.destroyAllWindows()
```

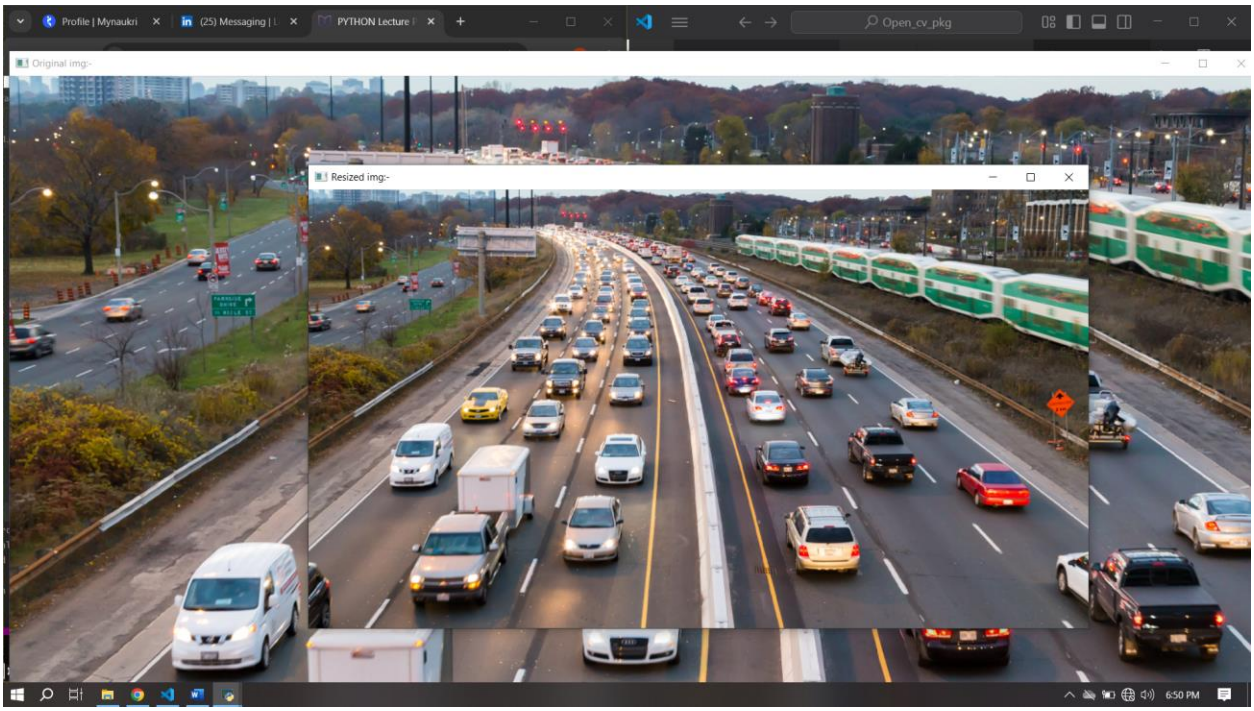

Horizontal img
Orginial img:-
Vertical img
6:30 PM

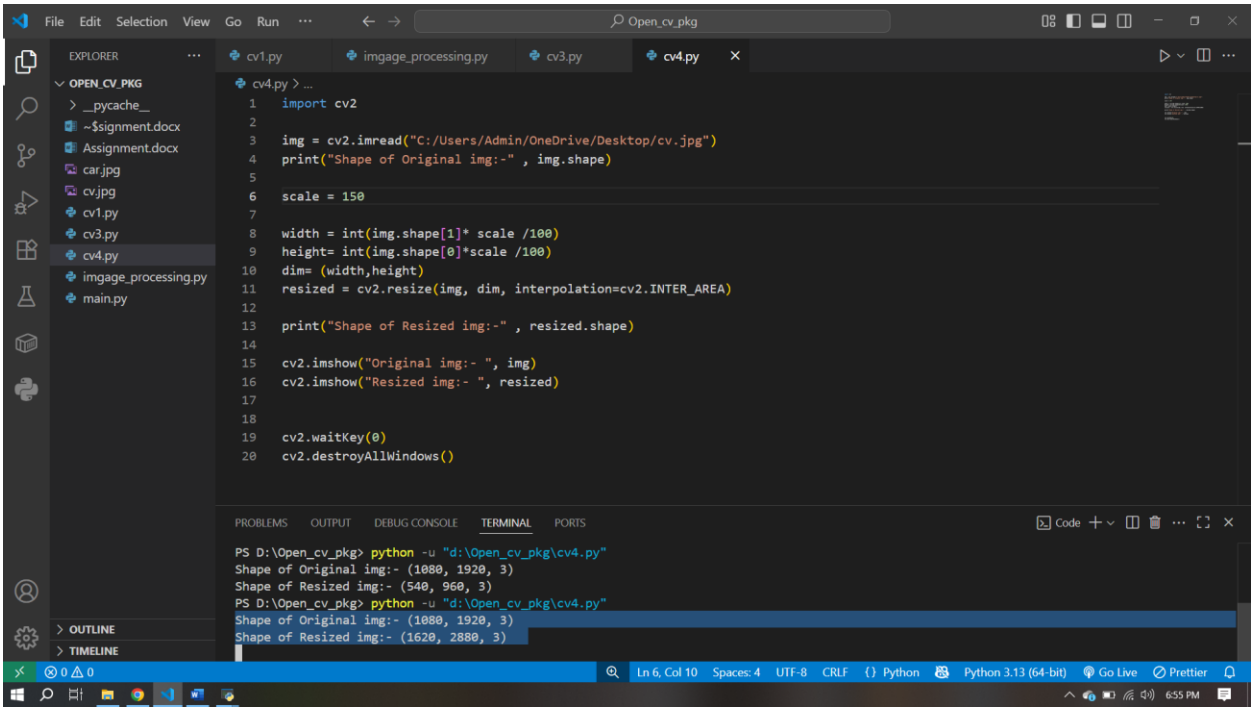

File
Edit
Selection
Horizontal & Vertical img
Orginial img:-
EXPLORER
OPEN_CV_PKG
__pycache__
~$signment.docx
Assignment.docx
car.jpg
cv.jpg
cv1.py
cv3.py
imgage_processing.
main.py
OUTLINE
TIMELINE
Prettier
6:32 PM

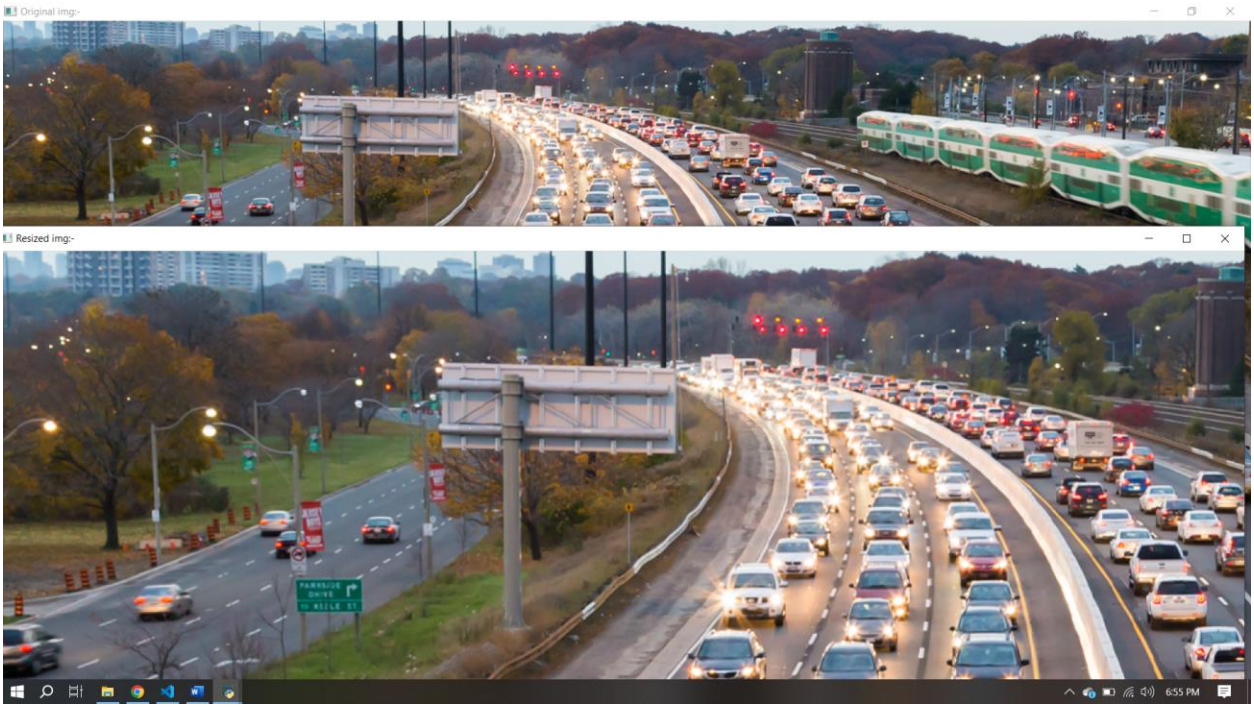



## Resizing and img based on aspect ration:-




Original img:-
Resized img:-
6:50 PM

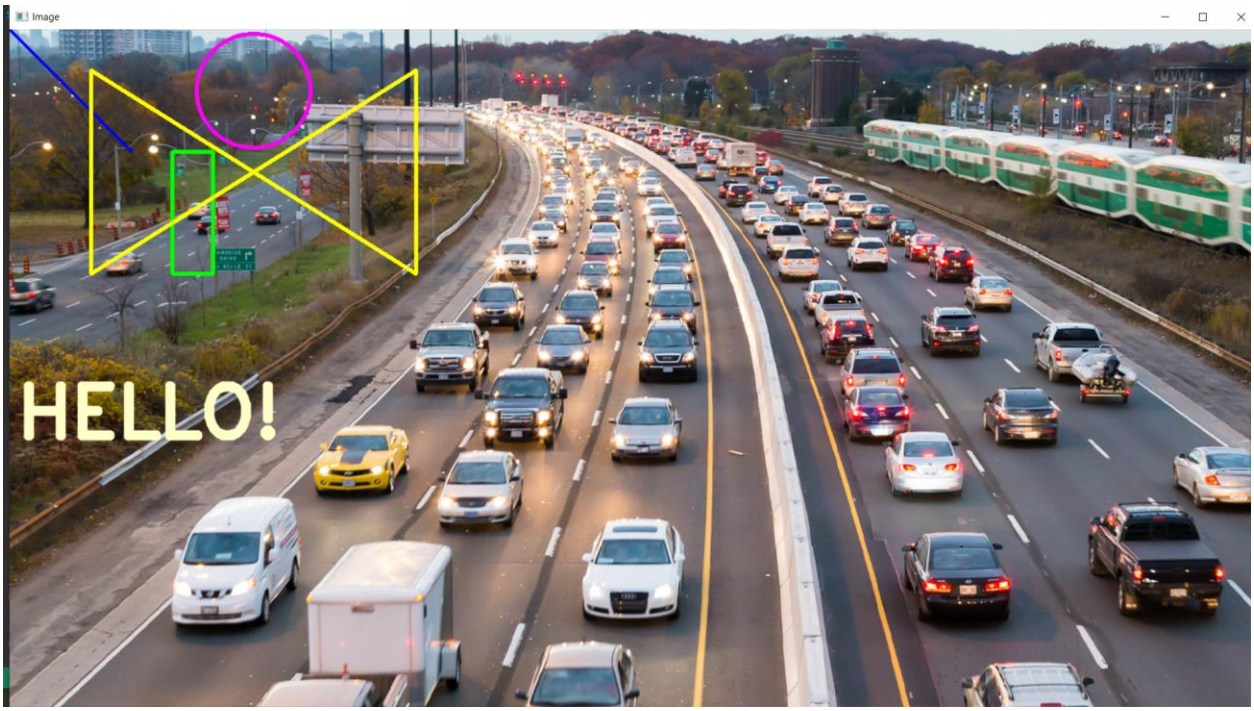

File Edit Selection View Go Run
Open_cv_pkg
EXPLORER
OPEN_CV_PKG
__pycache__
~$signment.docx
Assignment.docx
car.jpg
cv.jpg
cv1.py
cv3.py
cv4.py
imgage_processing.py
main.py
cv1.py
imgage_processing.py
cv3.py
cv4.py
cv4.py > ...
1 import cv2
2
3 img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")
4 print("Shape of Original img:-" , img.shape)
5
6 scale = 150
7
8 width = int(img.shape[1]* scale /100)
9 height= int(img.shape[0]*scale /100)
10 dim= (width,height)
11 resized = cv2.resize(img, dim, interpolation=cv2.INTER_AREA)
12
13 print("Shape of Resized img:-" , resized.shape)
14
15 cv2.imshow("Original img:- ", img)
16 cv2.imshow("Resized img:- ", resized)
17
18
19 cv2.waitKey(0)
20 cv2.destroyAllWindows()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Code
PS D:\Open_cv_pkg> python -u "d:\Open_cv_pkg\cv4.py"
Shape of Original img:- (1080, 1920, 3)
Shape of Resized img:- (540, 960, 3)
PS D:\Open_cv_pkg> python -u "d:\Open_cv_pkg\cv4.py"
Shape of Original img:- (1080, 1920, 3)
Shape of Resized img:- (1620, 2880, 3)
OUTLINE
TIMELINE
Ln 6, Col 10 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 (64-bit) Go Live Prettier
6:55 PM

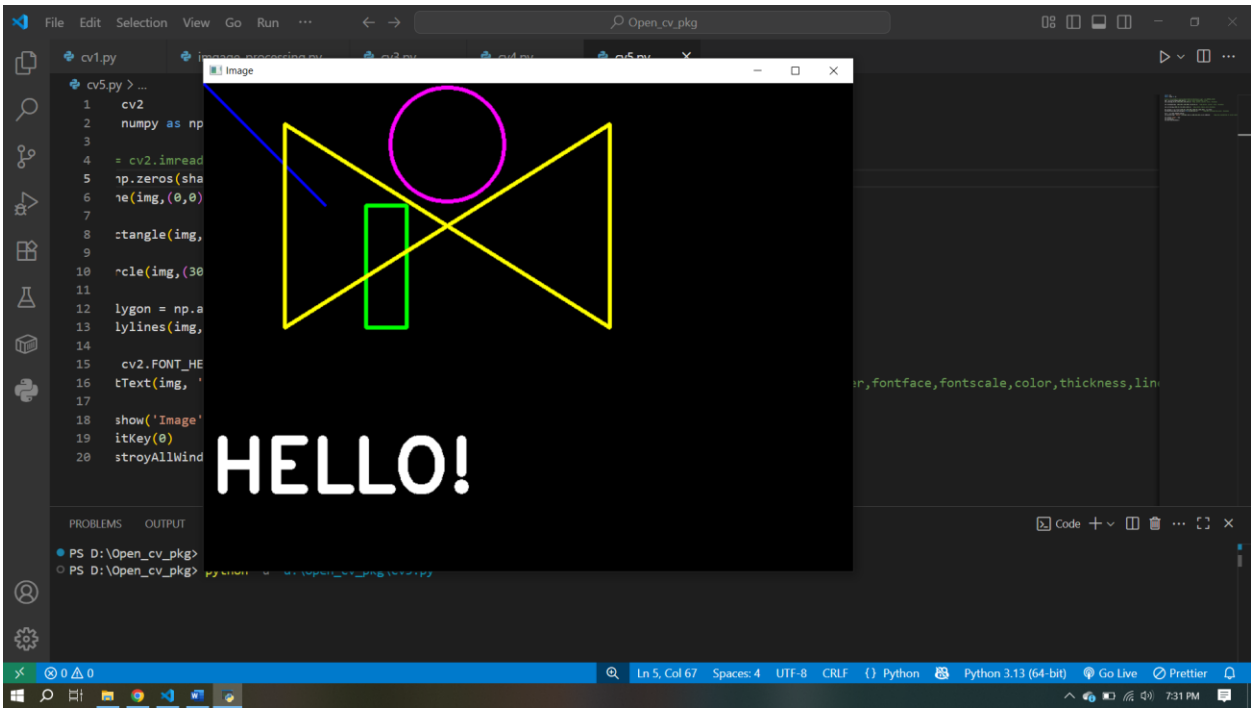


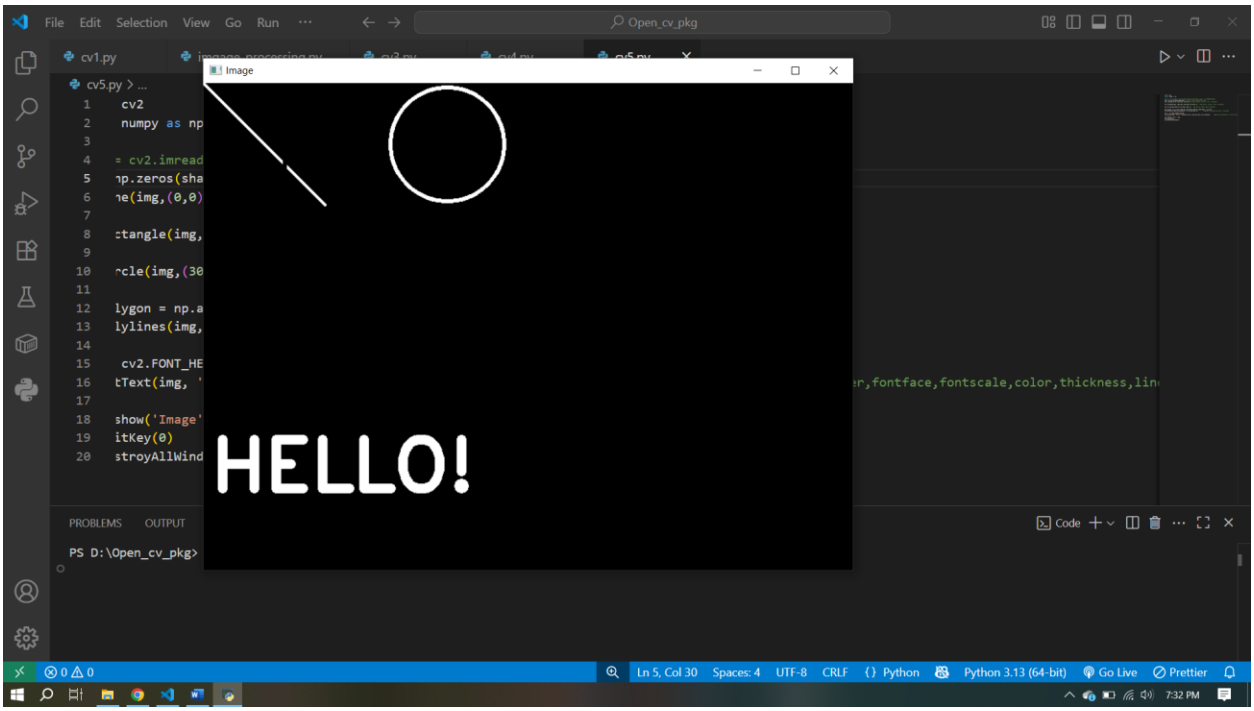

**Adding Shape and Text:-**

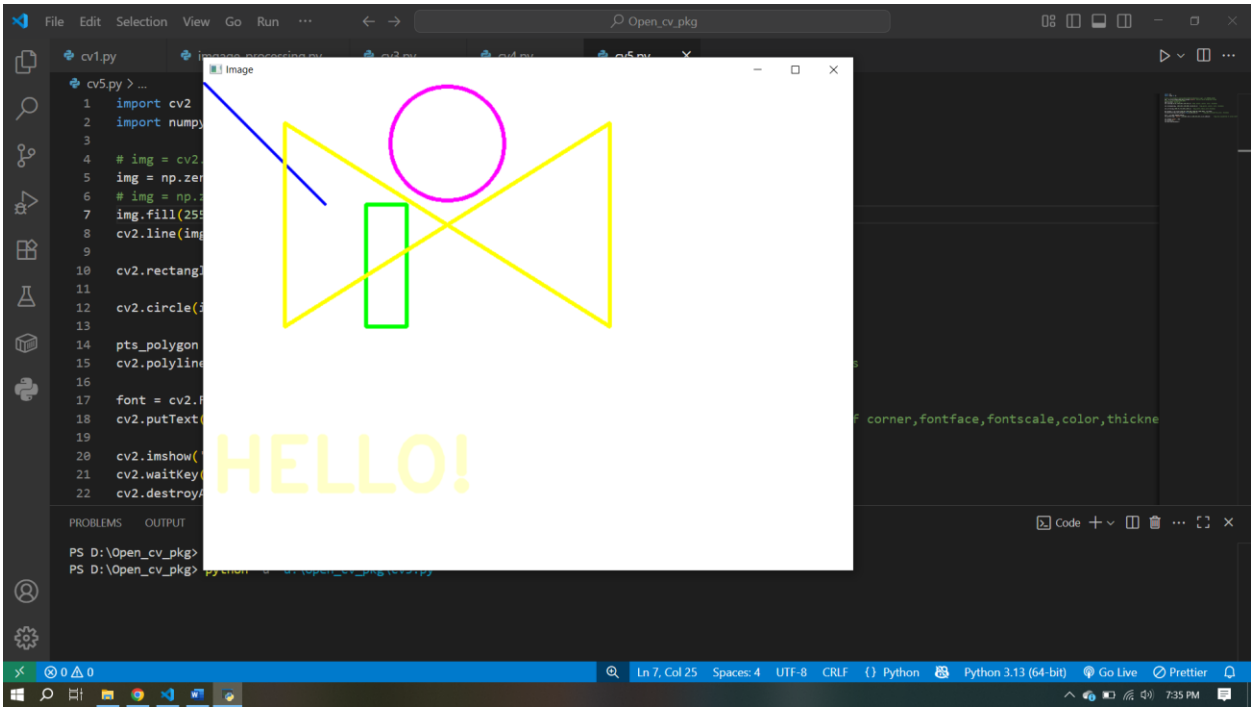
```
import cv2
import numpy as np

# img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg", cv2.IMREAD_COLOR)
img = np.zeros(shape=[600,800,3],dtype='uint8')  #zeros= balck background screen
# img = np.zeros(shape=[600,800])
img.fill(255)  #white bg
cv2.line(img,(0,0),(150,150),(255,0,0),2)  #img, point1, point2, color, thickness

cv2.rectangle(img, (200,150),(250,300),(0,255,0),3)   #img,point1, point2, color,
thickness

cv2.circle(img,(300,75),70,(255,0,255),3)   #img,center,radius,color,thickness

pts_polygon = np.array([[100,50],[100,300],[500,50],[500,300]], np.int32)
cv2.polylines(img,[pts_polygon],True,(0,255,255),3)         #img,pts,isclosed=tru
e,color, thickness

font = cv2.FONT_HERSHEY_DUPLEX
cv2.putText(img,
'HELLO!',(10,500),font,3,(200,255,255),8,cv2.LINE_AA)      #img,text,org=bottom
of corner,fontface,fontscale,color,thickness,line_type

cv2.imshow('Image', img)
```

```
cv2.waitKey(0)
cv2.destroyAllWindows()
```



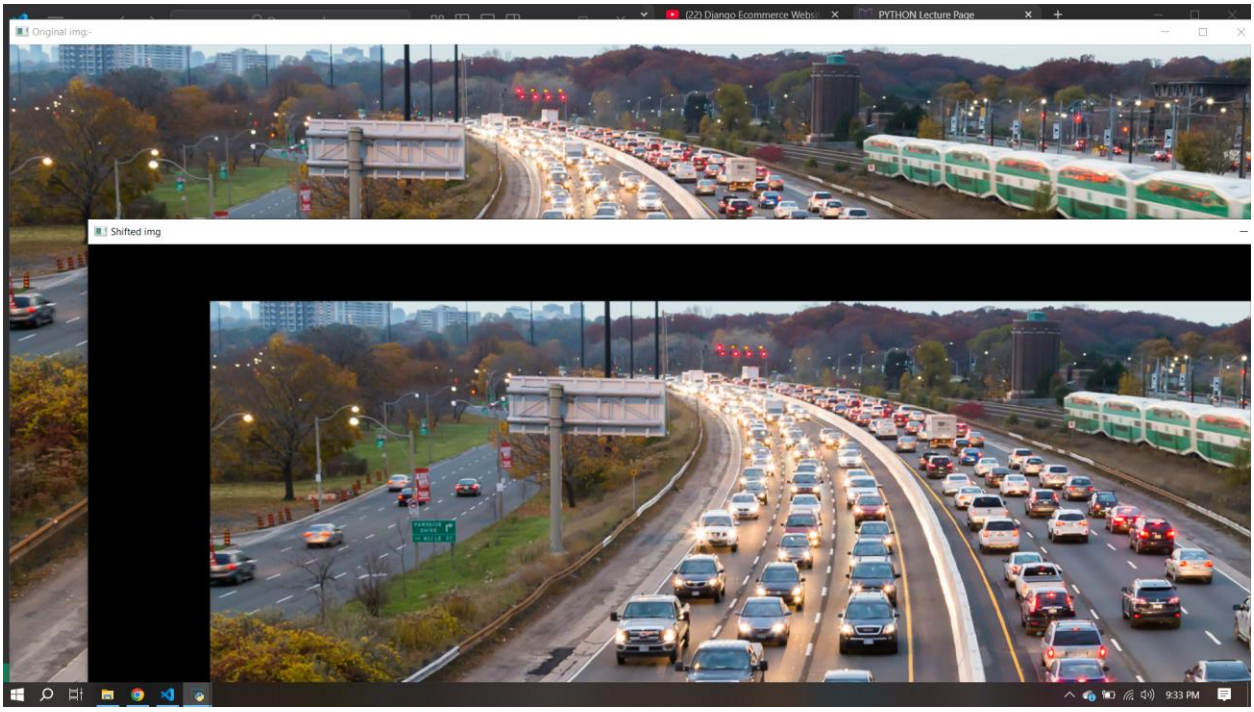



**Shifting an img:-**

```
import cv2
import numpy as np

img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")
```

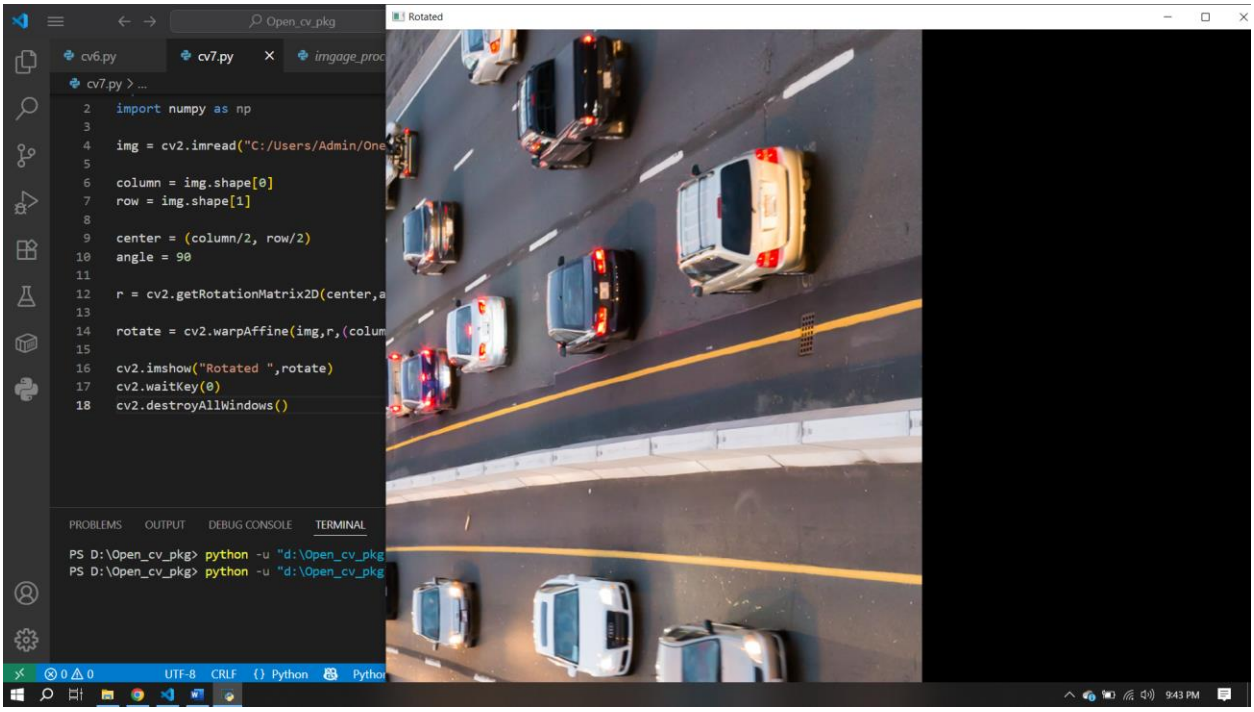
```
column = img.shape[1]
row = img.shape[0]

s = np.float32([(1,0,150),(0,1,70)])    #(right shiffting,height,150 ), (down
shifting)
shifted = cv2.warpAffine(img,s,(column,row))

cv2.imshow("Original img:- ", img)
cv2.imshow("Shifted img", shifted)

cv2.waitKey(0)
cv2.destroyAllWindows()
```




**How to rotate img:-**

```
import cv2
import numpy as np

img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")

column = img.shape[0]
row = img.shape[1]

center = (column/2, row/2)
# angle = 90
```

```
angle = 270
r = cv2.getRotationMatrix2D(center,angle,1)

rotate = cv2.warpAffine(img,r,(column,row))

cv2.imshow("Rotated ",rotate)
cv2.waitKey(0)
cv2.destroyAllWindows()
```



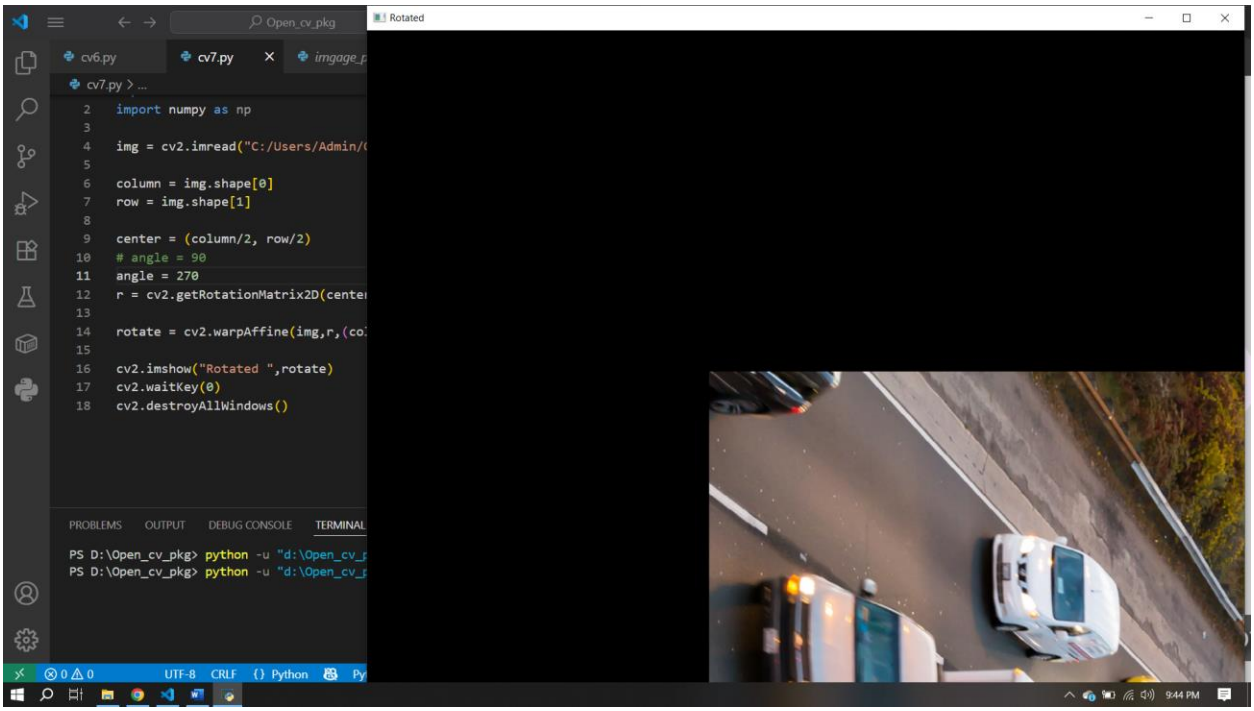
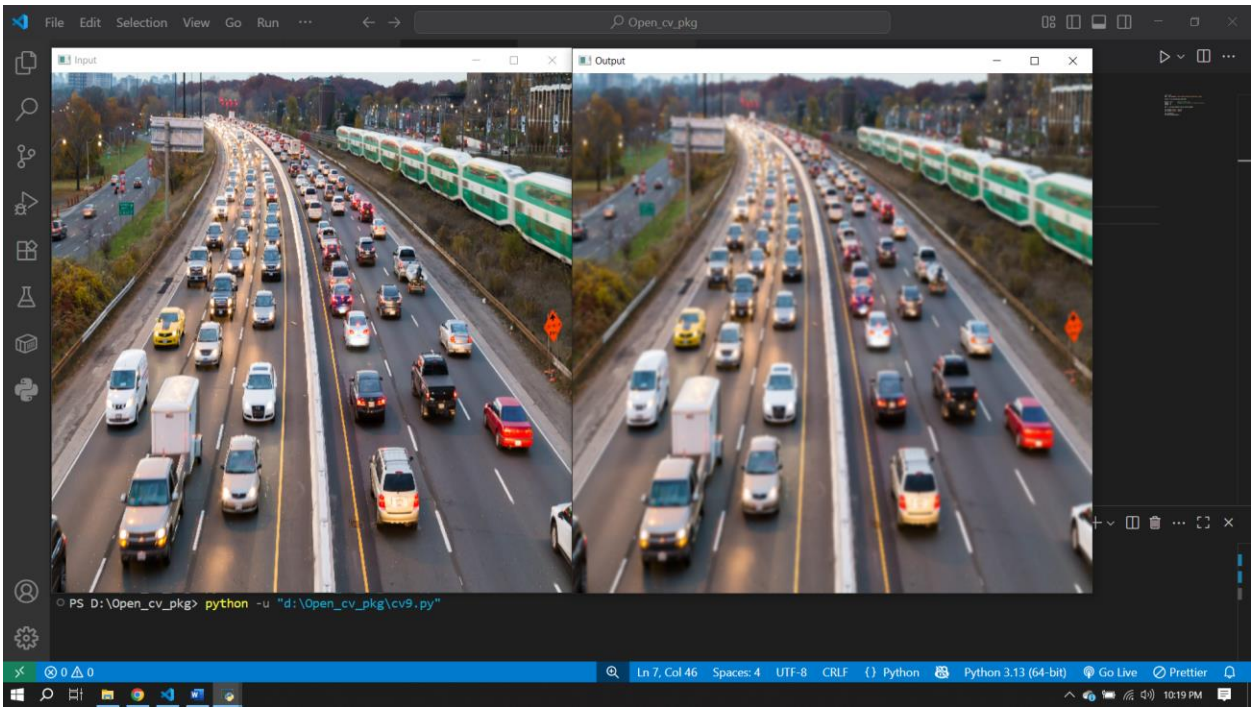



**Image thresholding: -**

```
# CHANNEL -->0 TO 255

import cv2
import numpy as np

img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")

threshold_value= 200

_, binary_threshold = cv2.threshold(img,threshold_value,255,cv2.THRESH_BINARY)

cv2.imshow("Original ", img)
cv2.imshow("Binary threshold", binary_threshold)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

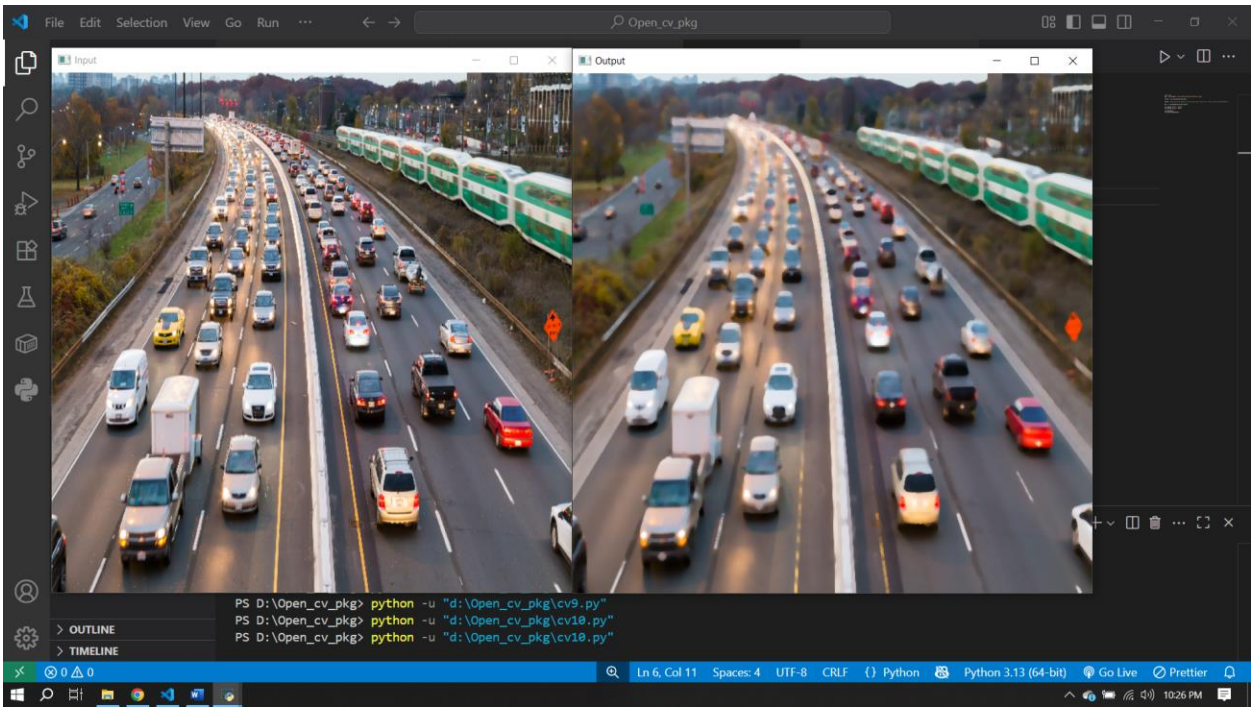

**Gaussian blur:-**

```
import cv2
img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")

resize = cv2.resize(img,(640,640))

ksize = (7,7)       #ksize= kernel size
sigmaX = 0          #sigmax,y its 0 then its taken from ksize
sigmaY = 0

blur = cv2.GaussianBlur(resize,ksize,sigmaX)

cv2.imshow("Input", resize)
cv2.imshow("Output", blur)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



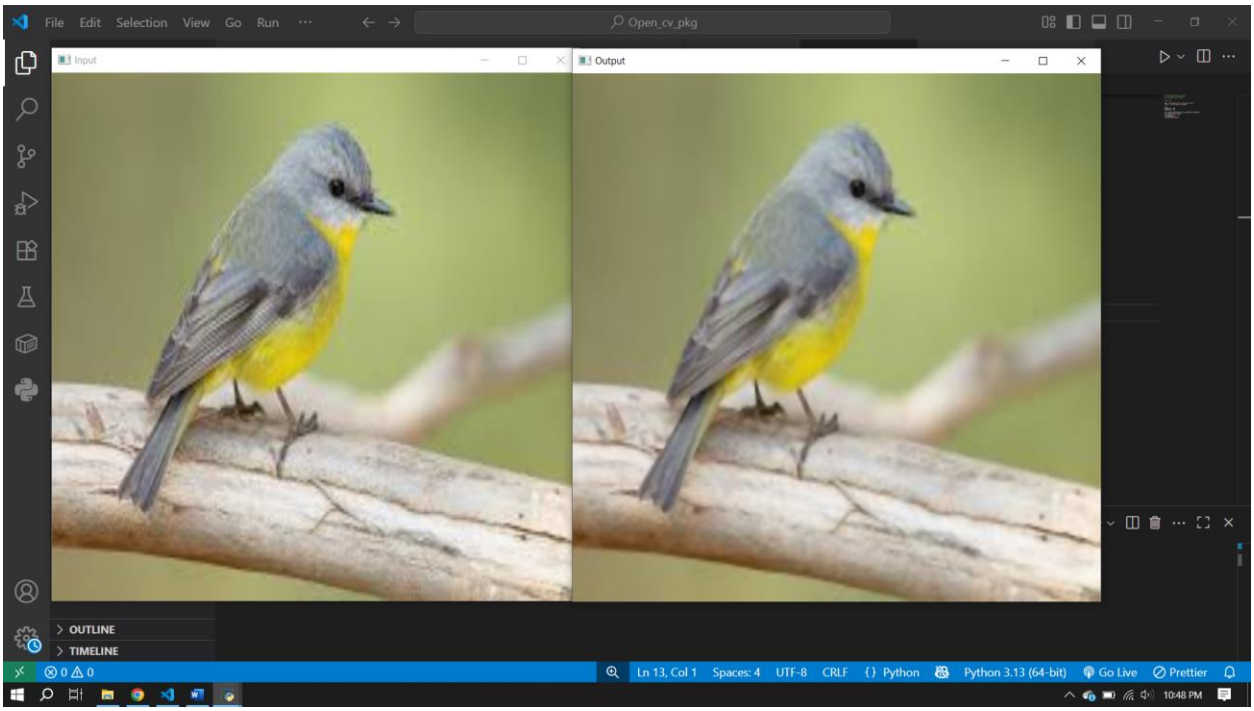

**Median blur:-**

```
import cv2
img = cv2.imread("C:/Users/Admin/OneDrive/Desktop/cv.jpg")

resize = cv2.resize(img,(640,640))

kernel = 7 #gaussian and median blur give same result only diff no need to give
standard deviation

blur = cv2.medianBlur(resize,kernel)

cv2.imshow("Input", resize)
cv2.imshow("Output", blur)

cv2.waitKey(0)
cv2.destroyAllWindows()
```



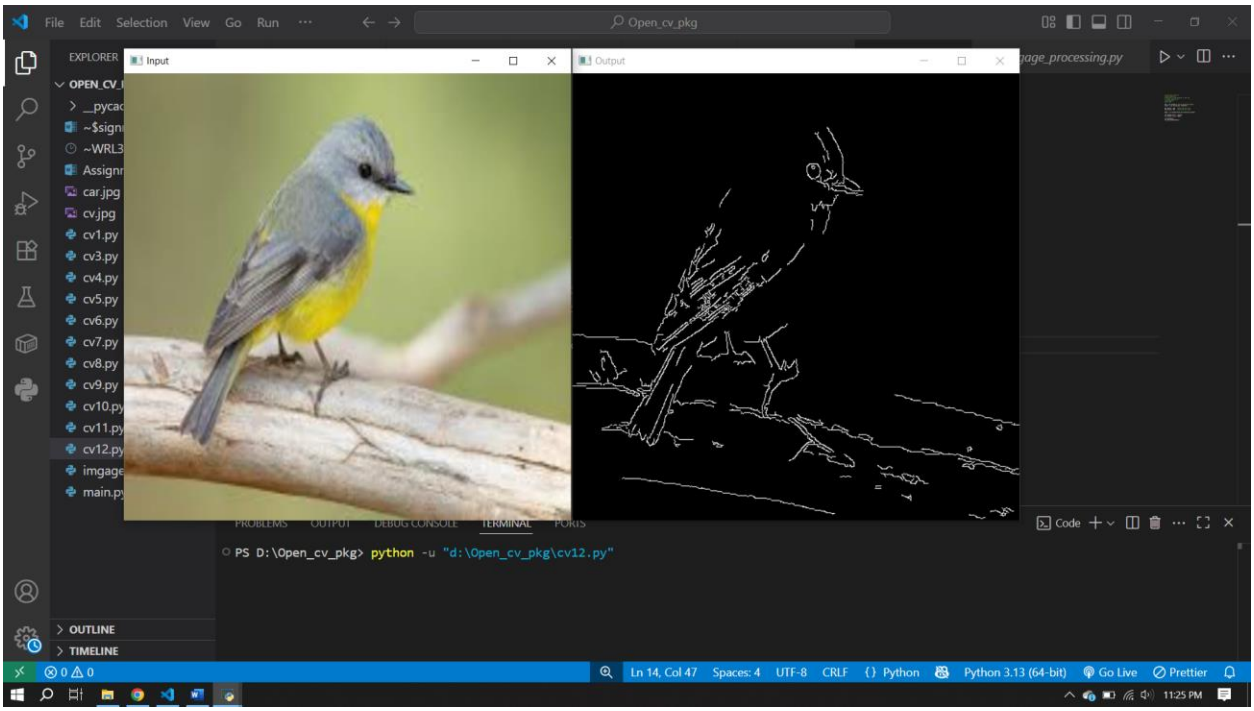

**Bilateral Filter-**

```
# In bilateral filter:- 1.src=ip
# 2.d= diameter of neighborhood
# 3.sigma color and 4. space

import cv2

img = cv2.imread("D:/web_scraping/bird.jpeg")
resize = cv2.resize(img, (650,650))

d= 7
sigmaColor = 100
sigmaSpace = 100

b = cv2.bilateralFilter(resize,d,sigmaColor,sigmaSpace)
cv2.imshow("Input",resize)
cv2.imshow("Output", b)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

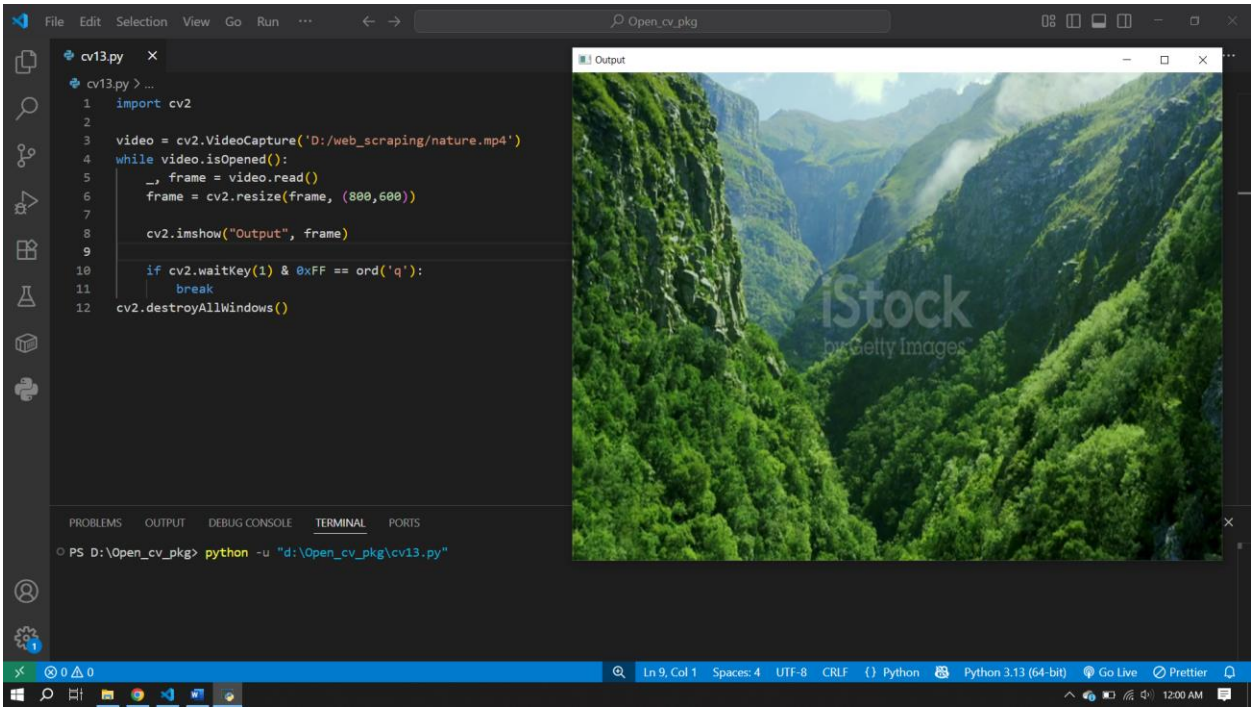



**Edge detection:-**

```
# Canny edge detection
# Noise Reduction
# Intensity of the gradient of the img
# non-max suppression
# thresholding
import cv2

img = cv2.imread("D:/web_scraping/bird.jpeg")
resize = cv2.resize(img, (550,550))

min_thresh = 100    #below 100 its black
max_thresh = 200    #above 200 its white

edges = cv2.Canny(resize,min_thresh,max_thresh)

cv2.imshow("Input", resize)
cv2.imshow("Output",edges)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

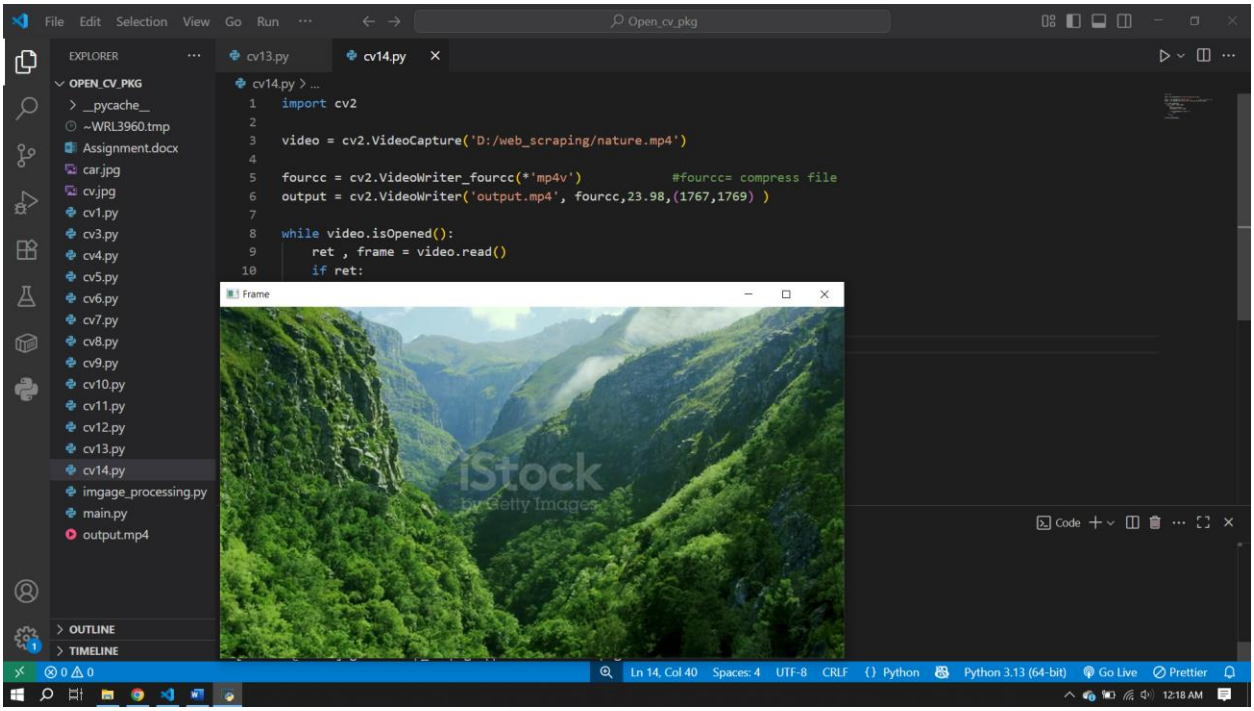



**Reading and writing a video:-**

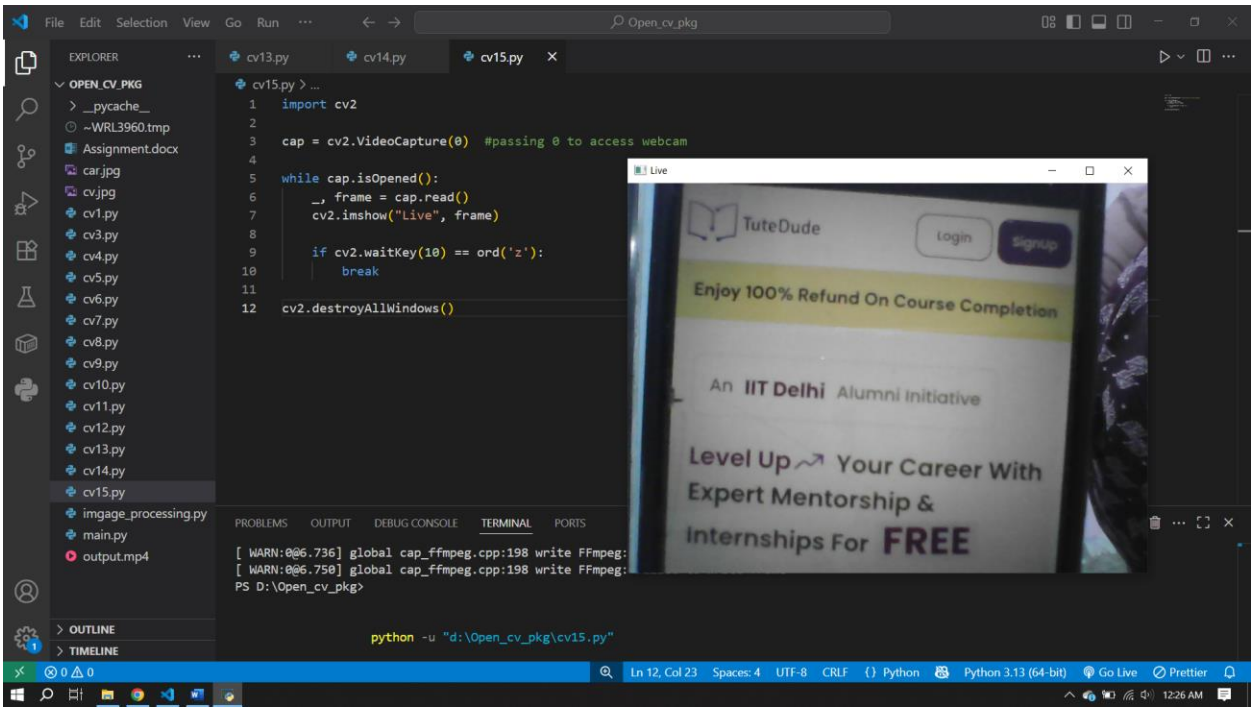



```
import cv2

video = cv2.VideoCapture('D:/web_scraping/nature.mp4')
```

```
fourcc = cv2.VideoWriter_fourcc(*'mp4v')            #fourcc= compress file
output = cv2.VideoWriter('output.mp4', fourcc,23.98,(1767,1769) )   #frame,
parameter in properties.details

while video.isOpened():
    ret , frame = video.read()
    if ret:
        output.write(frame)
        cv2.imshow('Frame',frame)

        if cv2.waitKey(10) == ord('s'):
            break
    else:
        break

cv2.destroyAllWindows()
```




**Accessing the webcam:-**


File Edit Selection View Go Run
Open_cv_pkg
EXPLORER
OPEN_CV_PKG
__pycache__
~WRL3960.tmp
Assignment.docx
car.jpg
cv.jpg
cv1.py
cv3.py
cv4.py
cv5.py
cv6.py
cv7.py
cv8.py
cv9.py
cv10.py
cv11.py
cv12.py
cv13.py
cv14.py
cv15.py
imgage_processing.py
main.py
output.mp4
OUTLINE
TIMELINE
cv13.py
cv14.py
cv15.py
cv15.py > ...
1 import cv2
2
3 cap = cv2.VideoCapture(0) #passing 0 to access webcam
4
5 while cap.isOpened():
6 _, frame = cap.read()
7 cv2.imshow("Live", frame)
8
9 if cv2.waitKey(10) == ord('z'):
10 break
11
12 cv2.destroyAllWindows()
Live
TuteDude
Login
Signup
Enjoy 100% Refund On Course Completion
An IIT Delhi Alumni Initiative
Level Up Your Career With
Expert Mentorship &
Internships For FREE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
[ WARN:0@6.736] global cap_ffmpeg.cpp:198 write FFmpeg:
[ WARN:0@6.750] global cap_ffmpeg.cpp:198 write FFmpeg:
PS D:\Open_cv_pkg>
python -u "d:\Open_cv_pkg\cv15.py"
Ln 12, Col 23 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 (64-bit) Go Live Prettier
12:26 AM