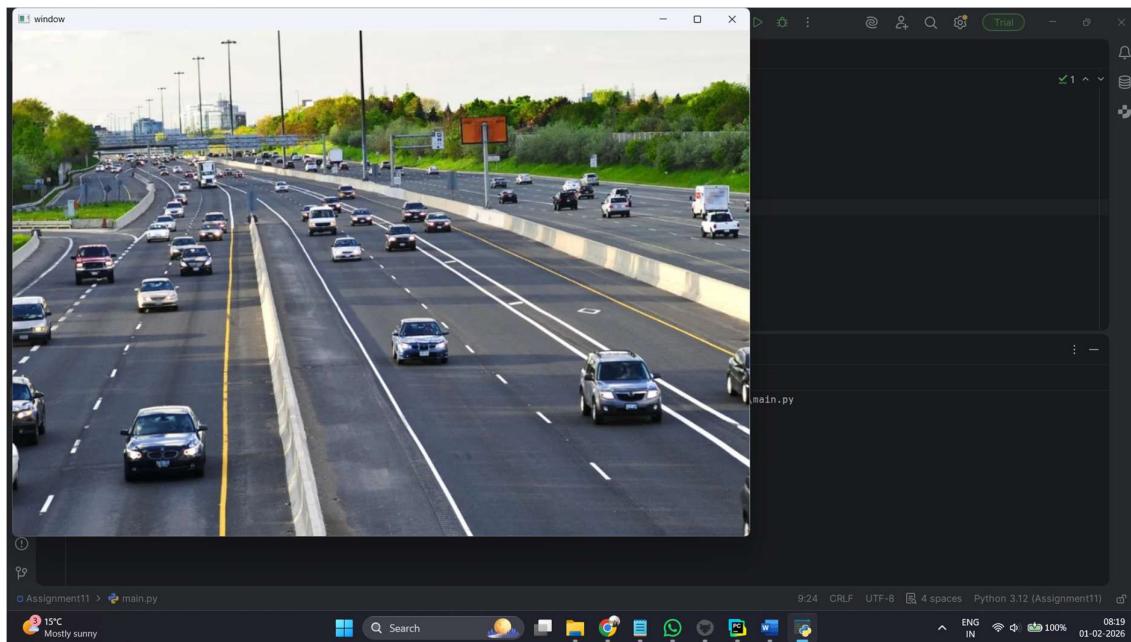
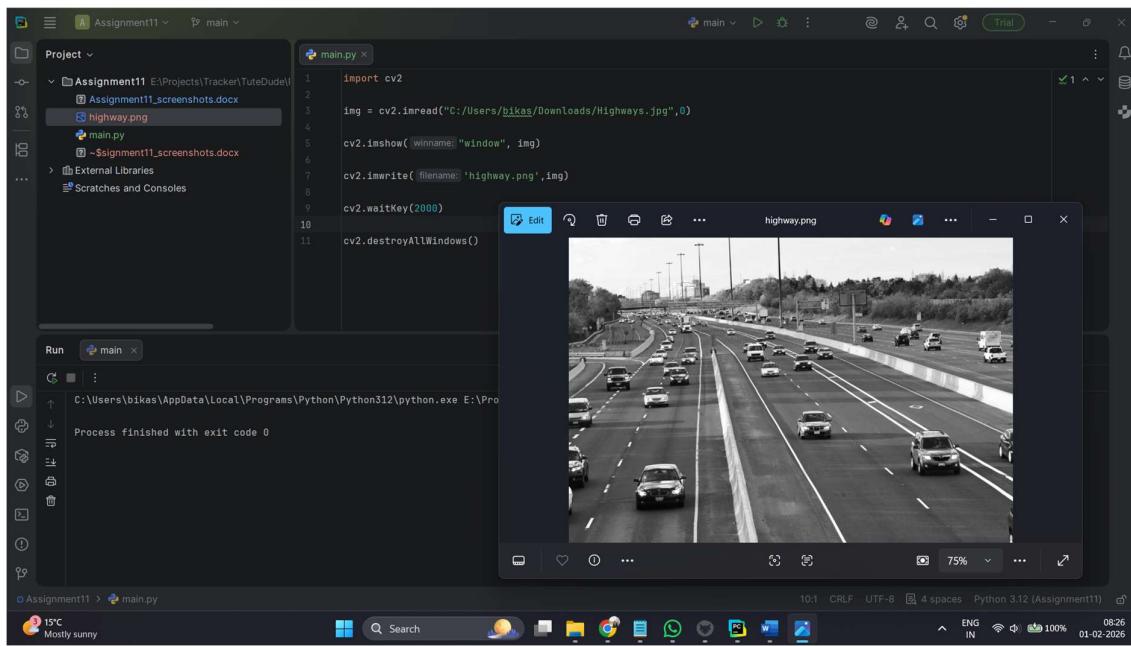


Task 1: Displaying an image



Task2: Image loaded as B/W and saved into file



Task3: Shape of Image

The screenshot shows the PyCharm IDE interface. The project tree on the left contains a folder named 'Assignment11' which includes files like 'highway.png', 'main.py', and 'Assignment11_screenshots.docx'. The main editor window displays the following Python code:

```
import cv2
img = cv2.imread("C:/Users/bikas/Downloads/Highways.jpg",0)
cv2.imshow("Window", img)
print("Shape of image is : ",img.shape)
cv2.waitKey(2000)
cv2.destroyAllWindows()
```

The 'Run' tool window at the bottom shows the output of running the script:

```
C:\Users\bikas\AppData\Local\Programs\Python\Python312\python.exe E:\Projects\Tracker\TuteDude\Python\Assignment11\main.py
Shape of image is : (685, 1000)
Process finished with exit code 0
```

The status bar at the bottom right indicates the date and time as 01-02-2026, and the Python version as 3.12 (Assignment11).

Task 4: Image resizing

The screenshot shows the PyCharm IDE interface. The project tree on the left contains a folder named 'Assignment11' which includes files like 'highway.png', 'main.py', and 'Assignment11_screenshots.docx'. The main editor window displays the following Python code:

```
import cv2
img = cv2.imread("C:/Users/bikas/Downloads/Highways.jpg",0)
height, width = img.shape[1:2]
width = 685 # new value assigned
dim = (width, height)
resized = cv2.resize(img, dim)
cv2.imshow(winname="window",resized)
cv2.waitKey(2000)
```

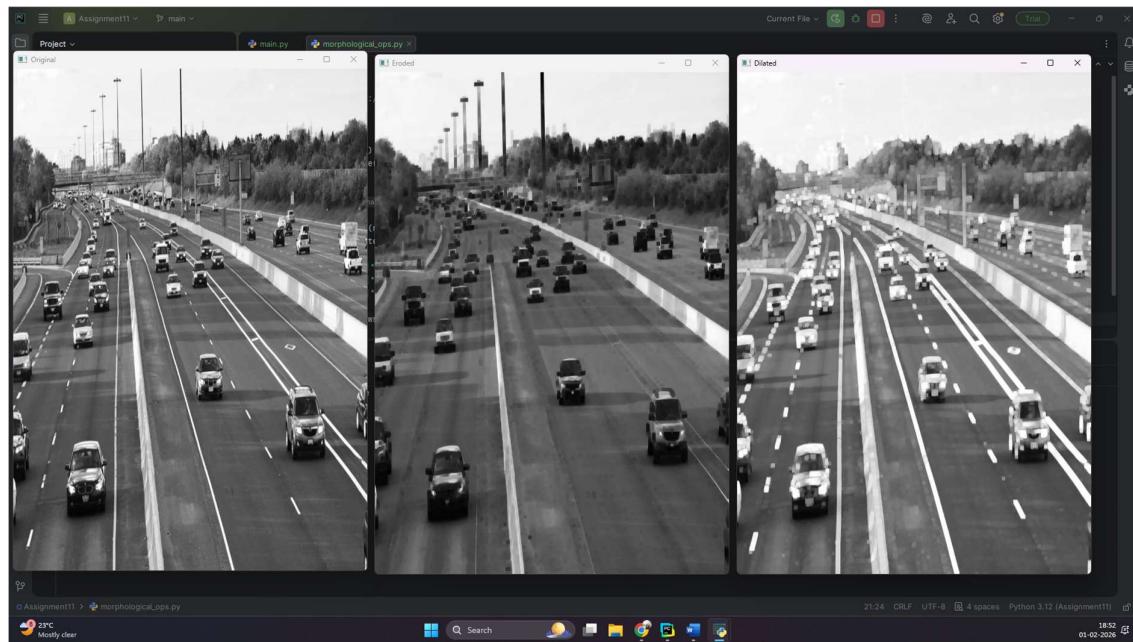
The 'Run' tool window at the bottom shows the output of running the script:

```
C:\Users\bikas\AppData\Local\Programs\Python\Python312\python.exe E:\Projects\Tracker\TuteDude\Python\Assignment11\main.py
Shape of image is : (685, 1000)
```

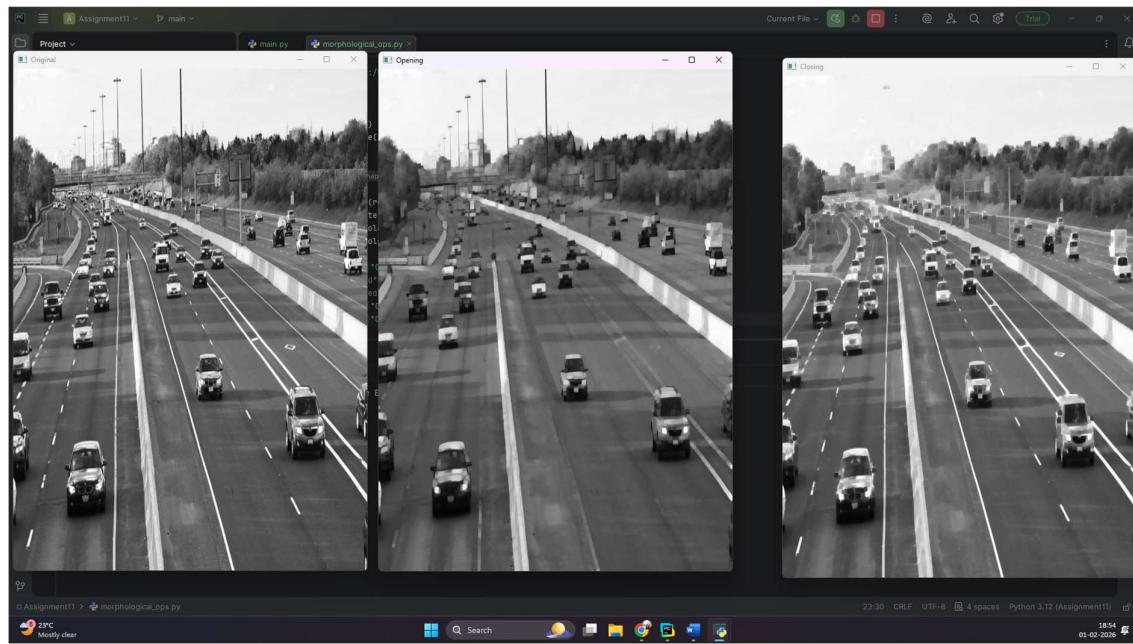
A separate window titled 'window' is open, displaying a grayscale image of a highway with multiple lanes and cars. The status bar at the bottom right indicates the date and time as 01-02-2026, and the Python version as 3.12 (Assignment11).

Task 5: Morphological operation

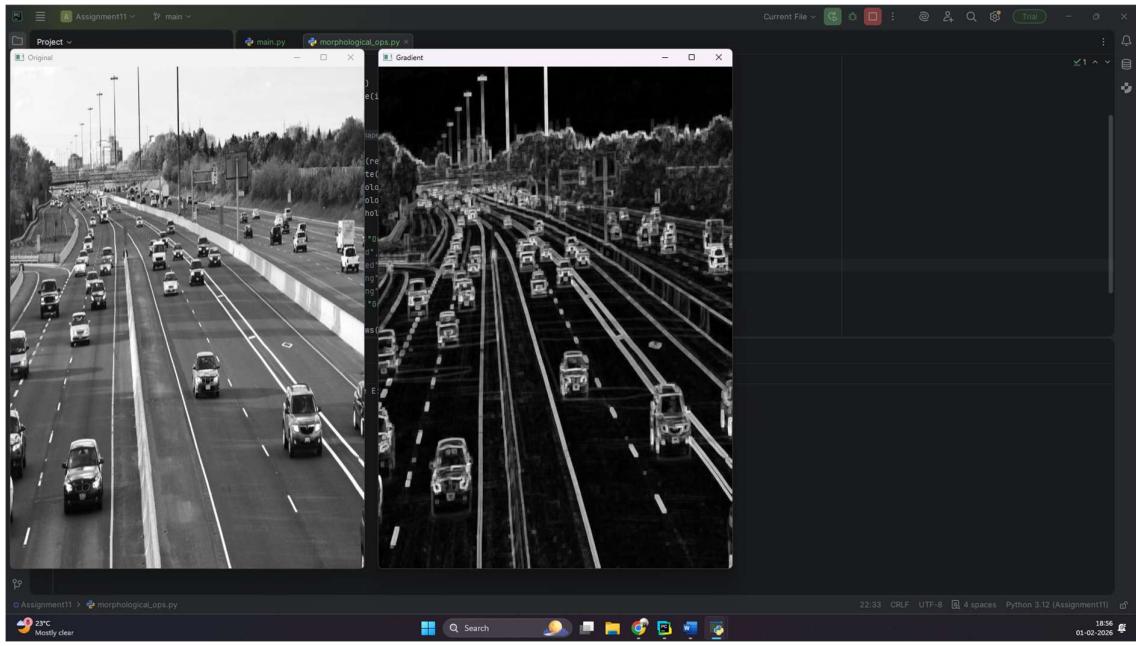
Erosion & dilation:



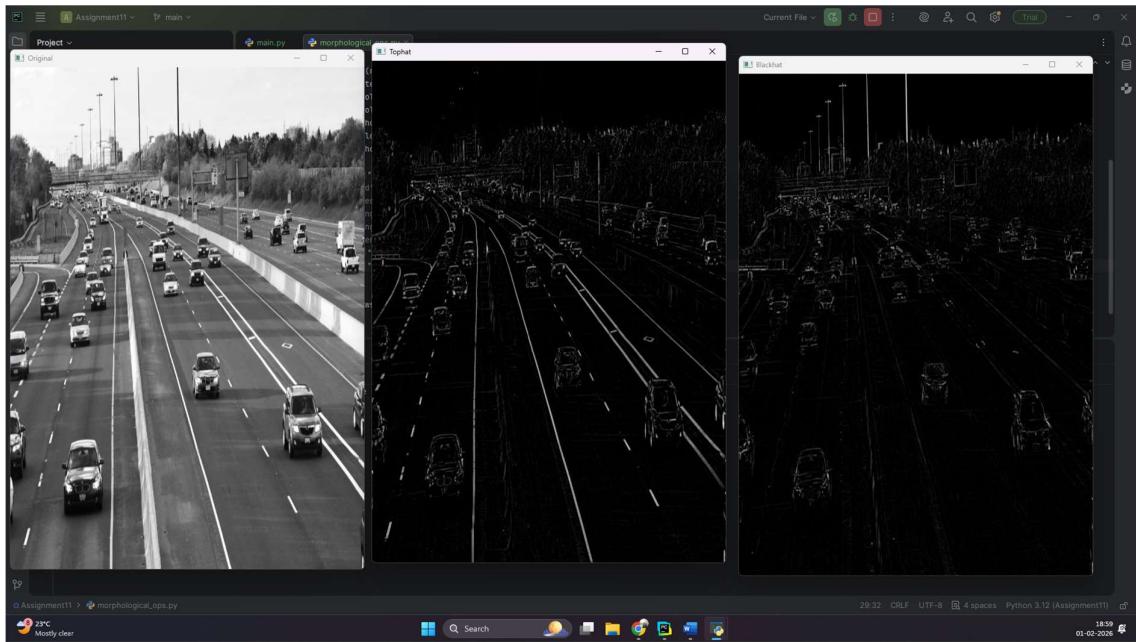
Opening and closing:



Gradient:

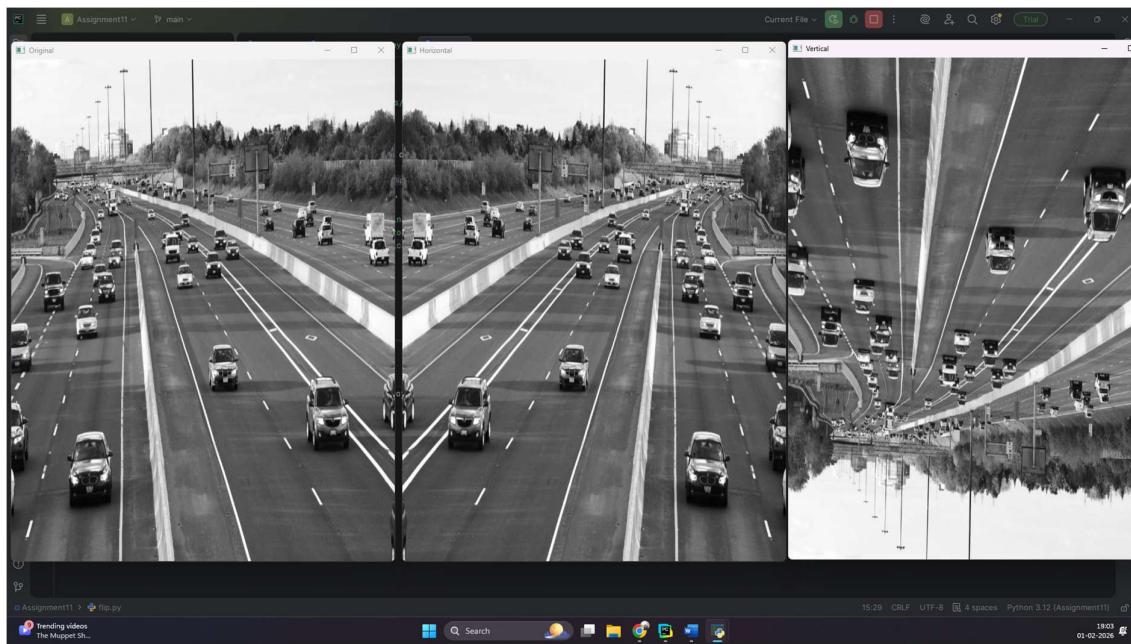


Tophat and Blackhat:

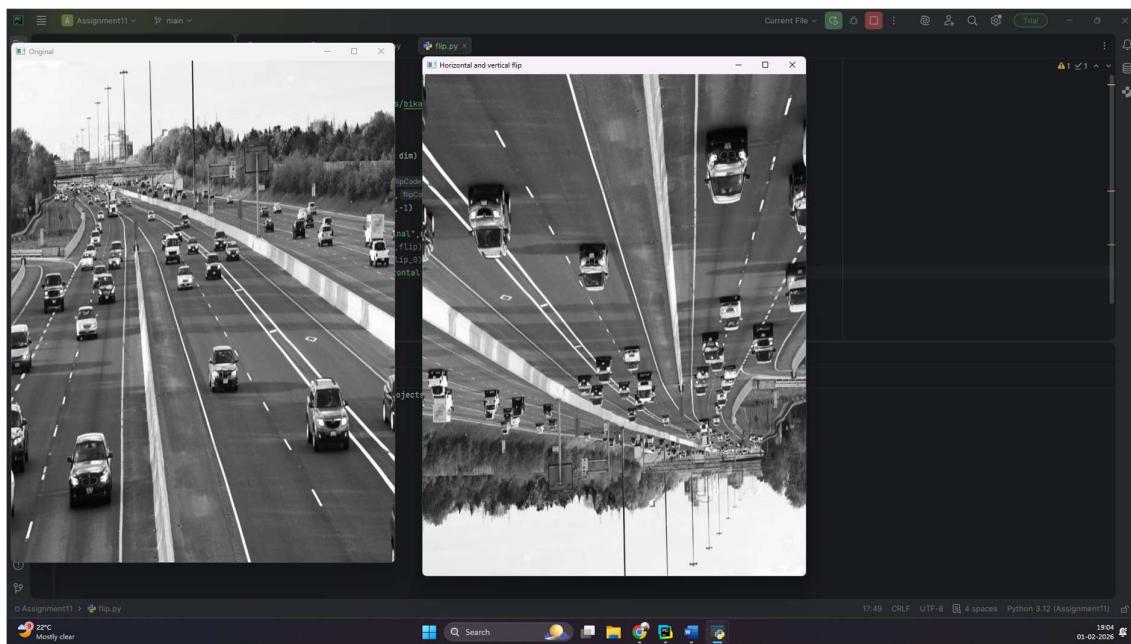


Task 6: Flipping

Horizontal vs vertical:

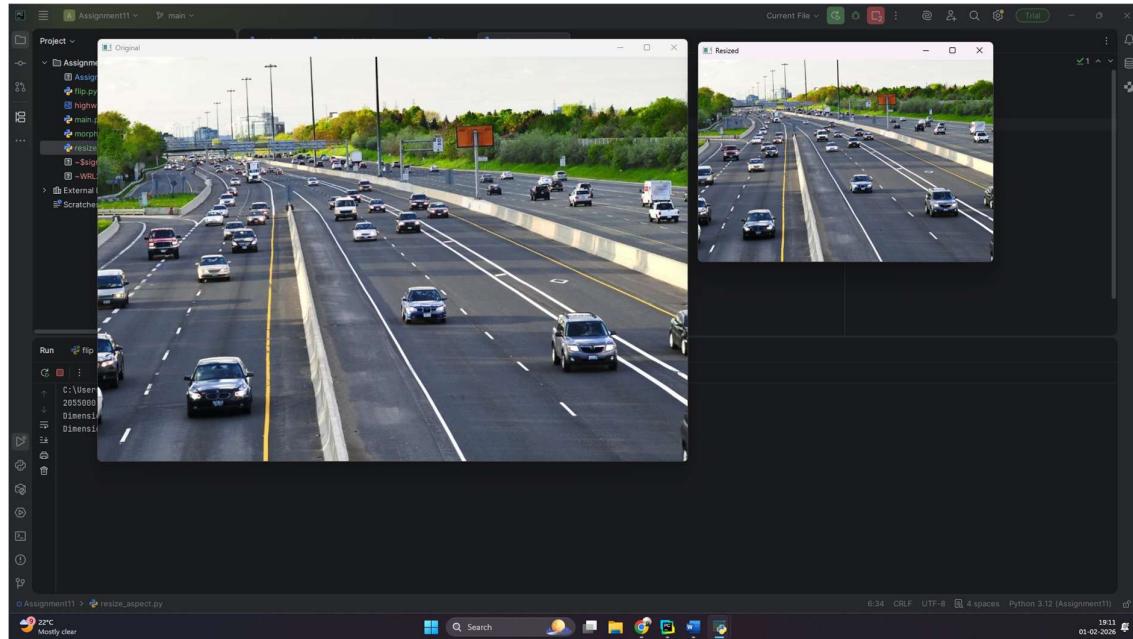


Horizontal and vertical:

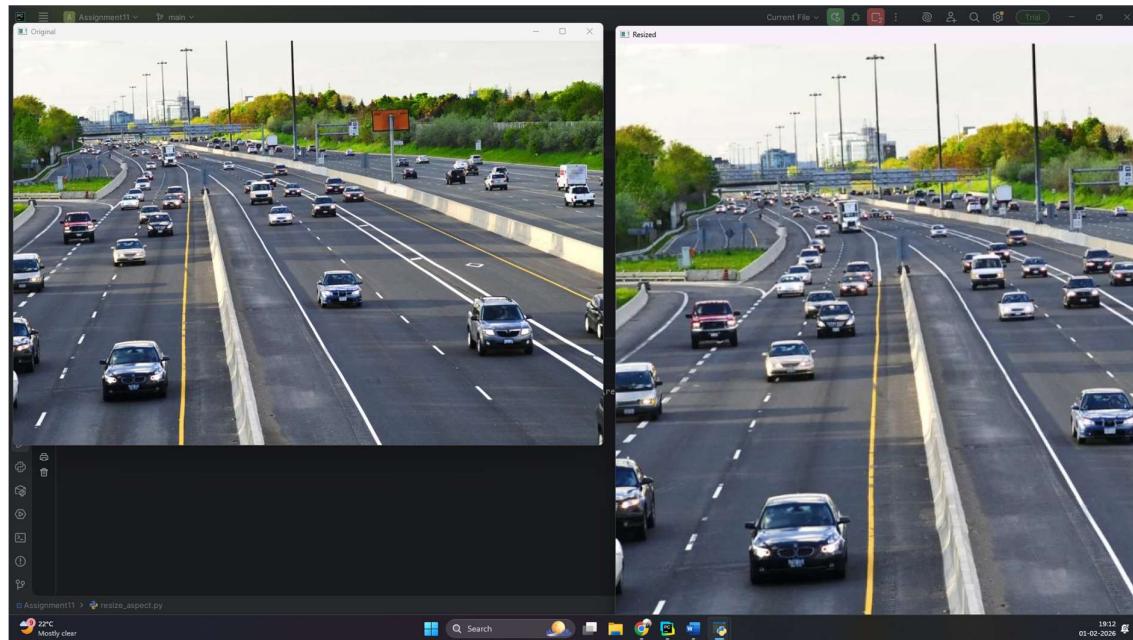


Task 7:

Scale down(50%):



Scale up(150%):



Task 8: shape and text

A screenshot of a Windows desktop with a Python IDE open. The IDE shows a project structure for 'Assignment11' with files like 'main.py', 'morphological_ops.py', 'flip.py', 'resize_aspect.py', and 'shape&text.py'. The 'shape&text.py' file contains Python code for drawing shapes and text on an image. The code imports cv2 and numpy, reads a highway image, and uses cv2 functions to draw a polygon, two rectangles, a circle, and the text 'Hello there'. The resulting image is displayed in a window, showing a highway scene with red outlines of the drawn shapes and the text 'Hello there' in red.

```
import cv2
import numpy as np
img = cv2.imread("C:/Users/bikas/Downloads/Highways.jpg")

cv2.line(img, pt1=(0,0), pt2=(200,200), color=(0,0,255), thickness=2)
cv2.rectangle(img, (800,0), (600,400), (0,0,255), 2)
cv2.circle(img, center=(500,500), radius=100, color=(0,0,255), thickness=2)

pts_polygon = np.array([(100,300),(150,280),(100,250),(50,300)])
cv2.polylines(img, pts=[pts_polygon], isClosed=True, color=(0,0,255), thickness=2)

font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, text="Hello there", org=(400,300), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=1, color=(0,0,255), thickness=2)

cv2.imshow("newname", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

A screenshot of a Windows desktop with a Python IDE open. The IDE shows a project structure for 'Assignment11' with files like 'main.py', 'morphological_ops.py', 'flip.py', 'resize_aspect.py', and 'shape&text.py'. The 'shape&text.py' file contains Python code for drawing shapes and text on a black background. The code imports cv2 and numpy, reads a black image, and uses cv2 functions to draw a polygon, two rectangles, a circle, and the text 'Hello there'. The resulting image is displayed in a window, showing a black background with red outlines of the drawn shapes and the text 'Hello there' in red.

```
import cv2
import numpy as np
img = cv2.imread("C:/Users/bikas/Downloads/Highways.jpg")

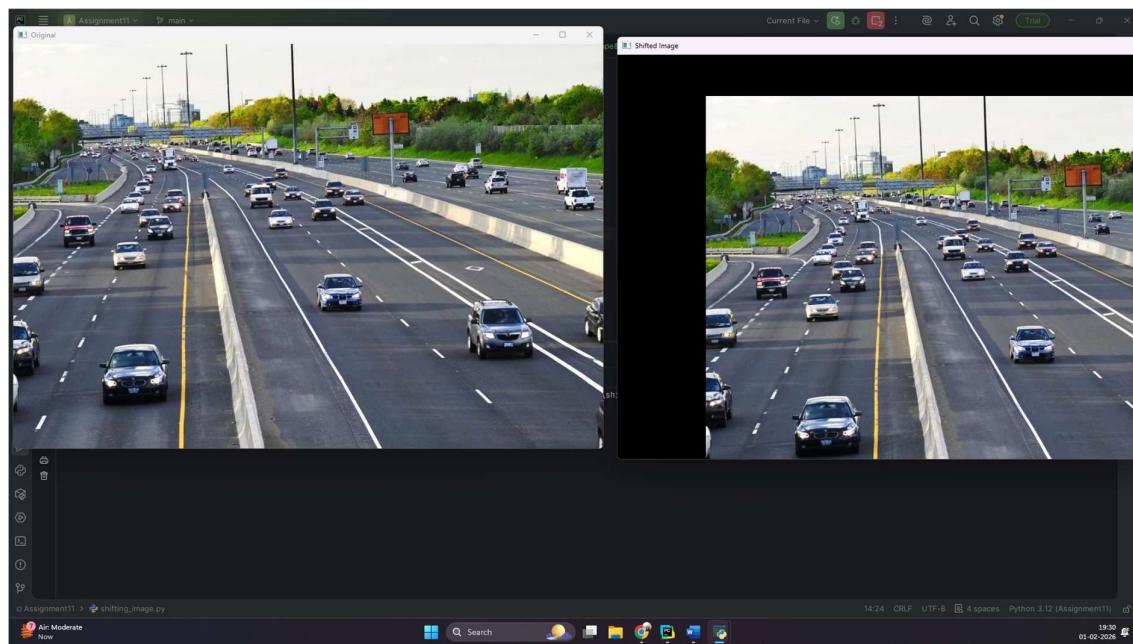
cv2.line(img, pt1=(0,0), pt2=(200,200), color=(0,0,255), thickness=2)
cv2.rectangle(img, (800,0), (600,400), (0,0,255), 2)
cv2.circle(img, center=(500,500), radius=100, color=(0,0,255), thickness=2)

pts_polygon = np.array([(100,300),(150,280),(100,250),(50,300)])
cv2.polylines(img, pts=[pts_polygon], isClosed=True, color=(0,0,255), thickness=2)

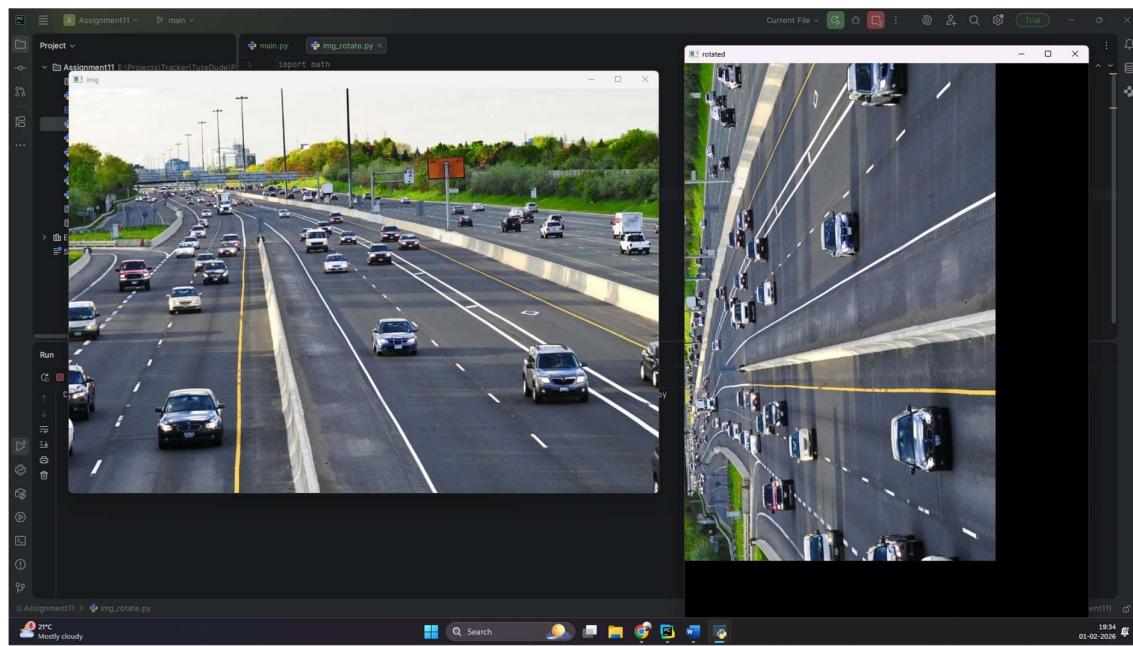
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, text="Hello there", org=(400,300), fontFace=cv2.FONT_HERSHEY_SIMPLEX, fontScale=1, color=(0,0,255), thickness=2)

cv2.imshow("newname", img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

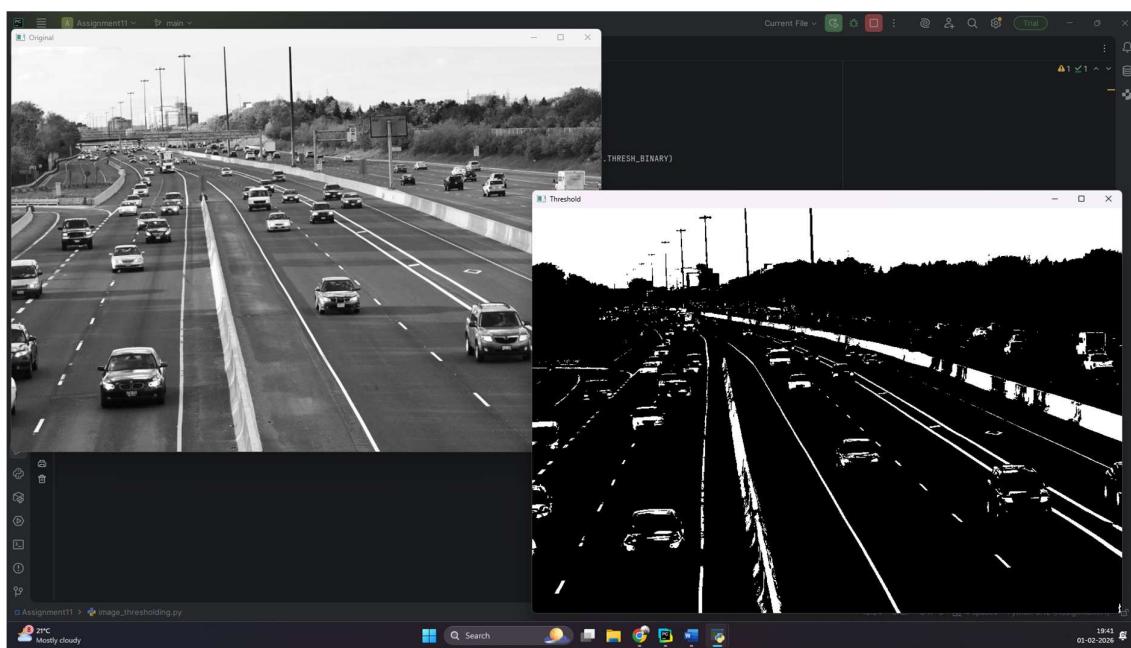
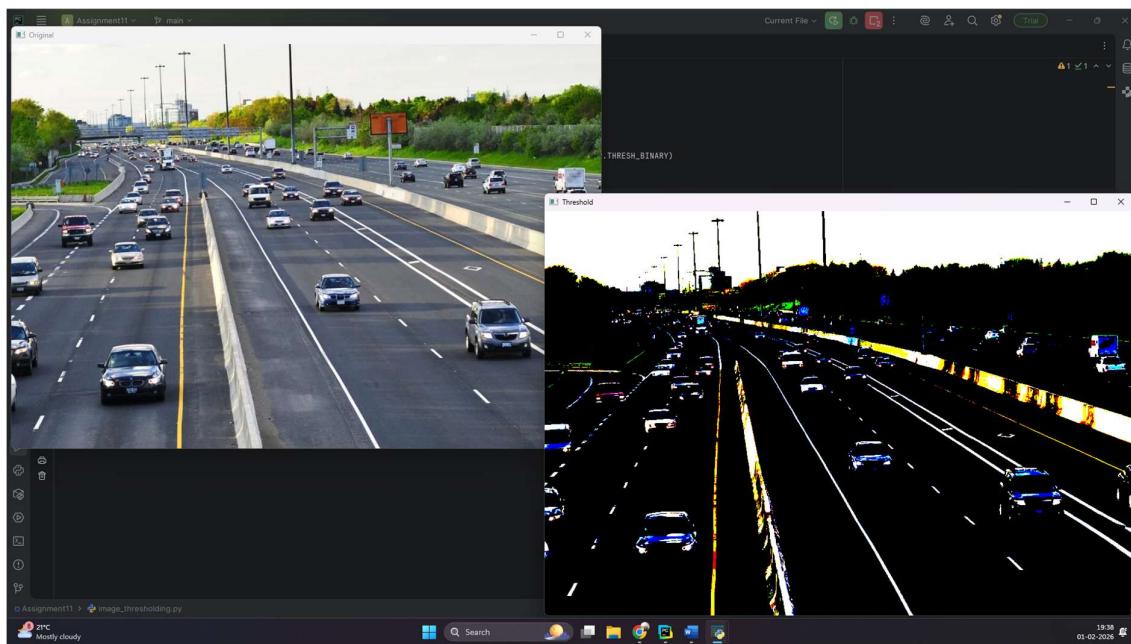
Task 9: Shifting image



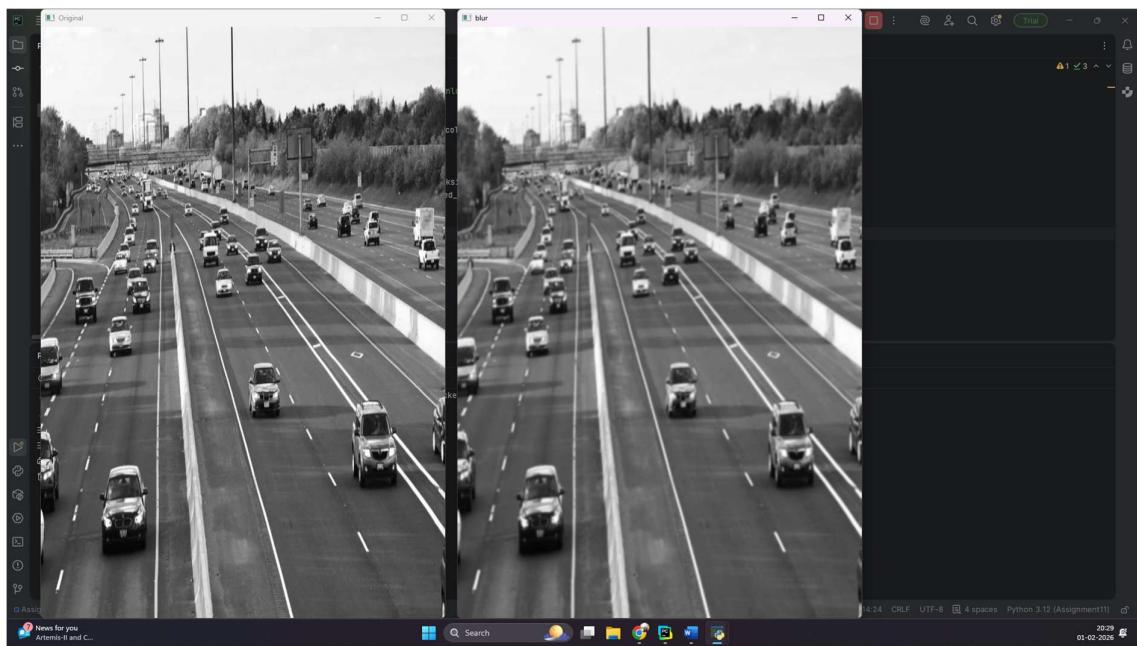
Task 10: Image rotation



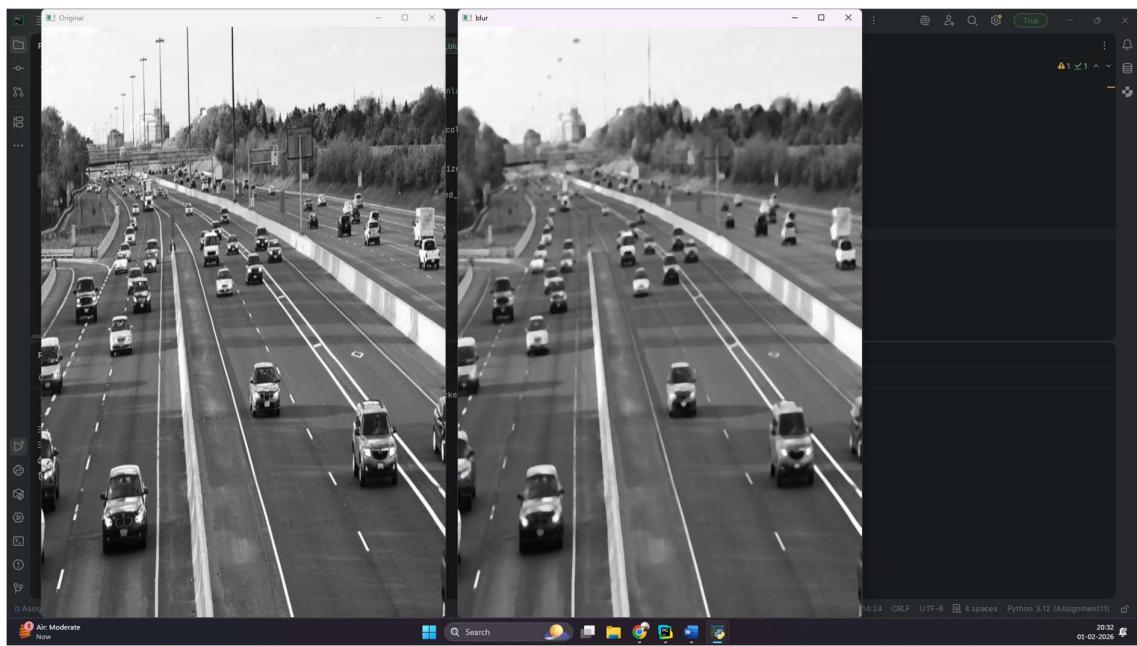
Task 11: Image Thresholding



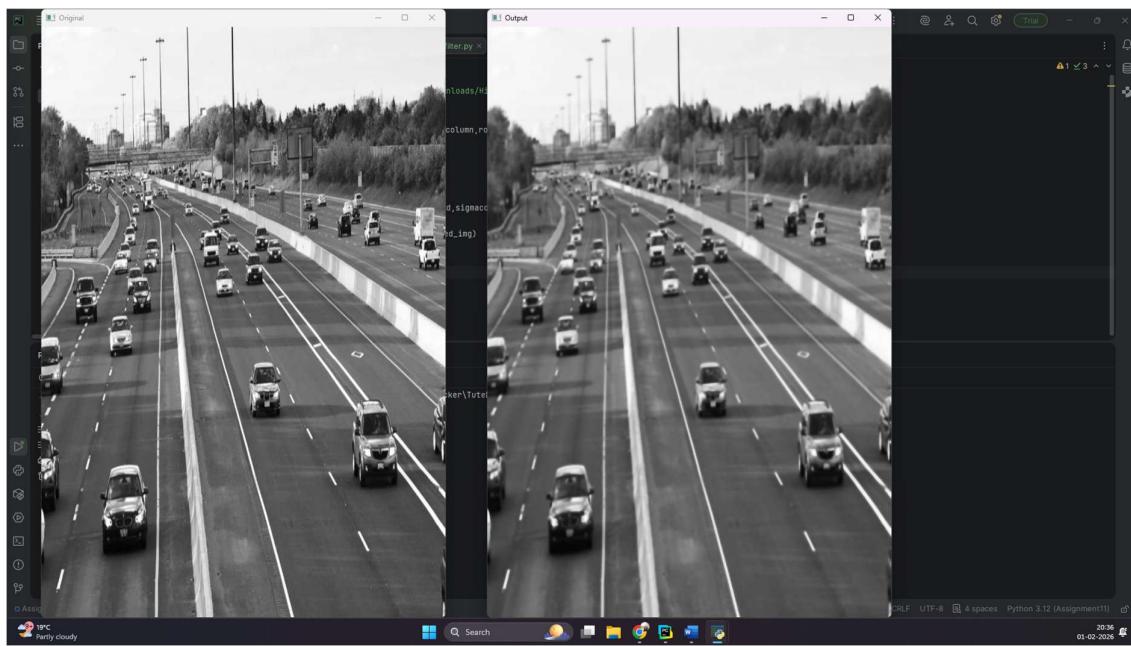
Task 12: Gaussian Blur



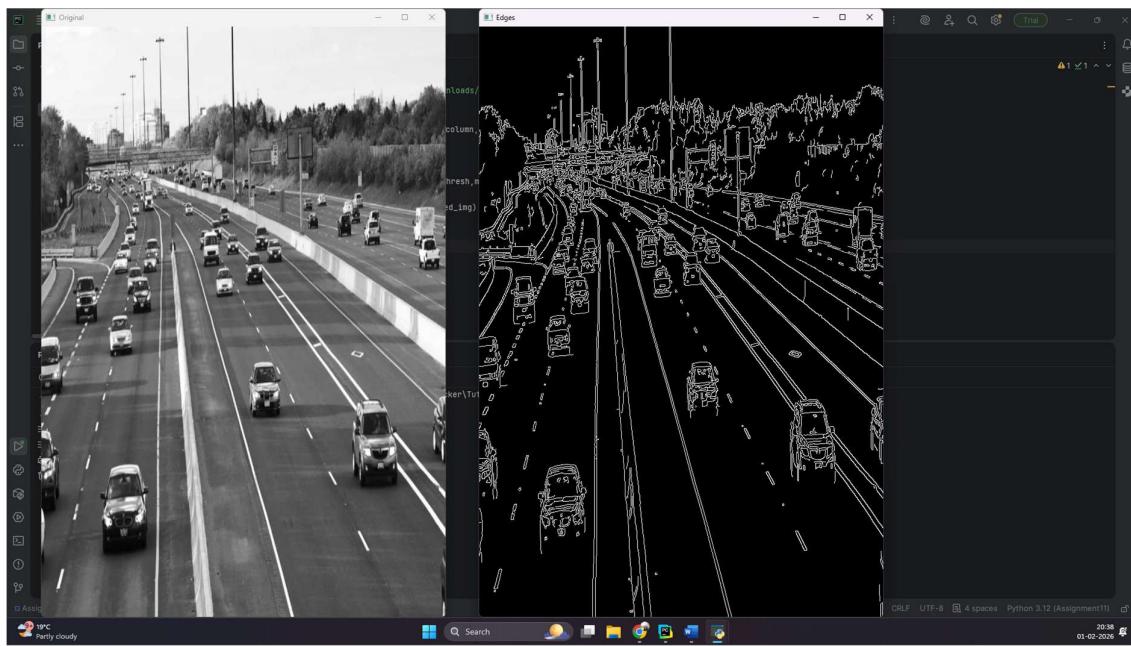
Task 13: Median blur



Task 14: Bilateral filter



Task 15: Edge Detection

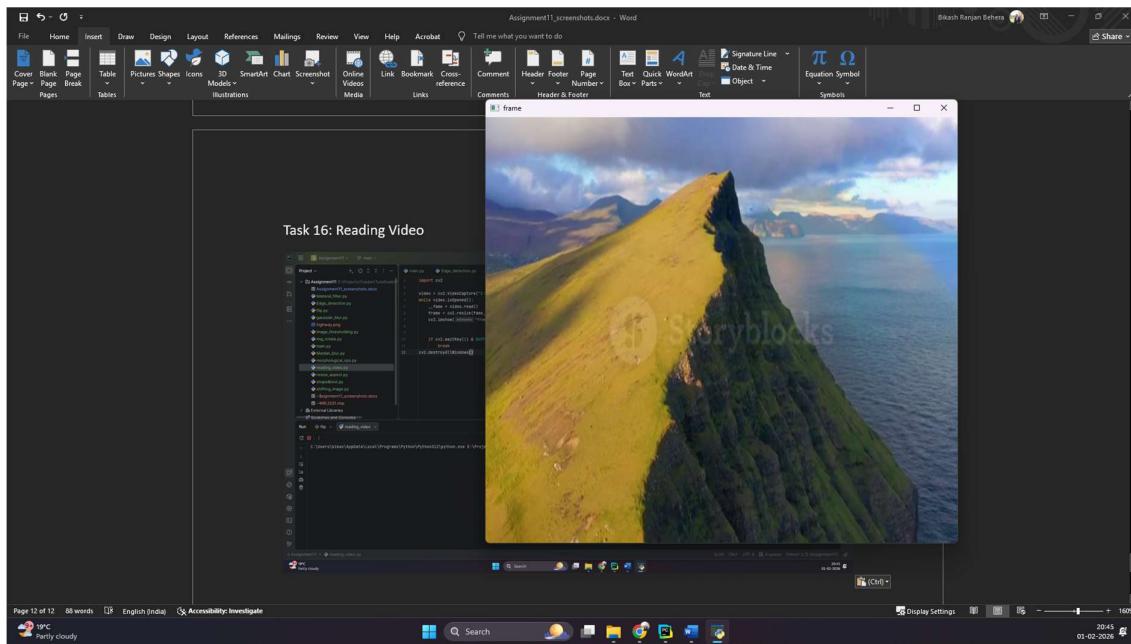


Task 16: Reading Video



```
import cv2
video = cv2.VideoCapture("C:/users/bikas/downloads/video.mp4")
while video.isOpened():
    frame = cv2.resize(frame, (800,720))
    cv2.imshow('frame',frame)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
cv2.destroyAllWindows()
```



Task 17: Writing video

The screenshot shows the PyCharm IDE interface. On the left is the Project tree with files like main.py, Edge_detection.py, reading_video.py, and writing_video.py. The writing_video.py file is open in the editor. It contains Python code for writing video frames. A properties dialog for 'output.mp4' is open on the right, showing details such as Length: 00:01:09, Frame width: 640, Frame height: 360, Data rate: 1364kps, Total bitrate: 1364kps, and Frame rate: 25.00 frames/second.

```
4     fourcc = cv2.VideoWriter_fourcc(*'mp4v')
5
6     # try:
7     #     frame_width = int(video.get(cv2.CAP_PROP_FRAME_WIDTH))
8     #     frame_height = int(video.get(cv2.CAP_PROP_FRAME_HEIGHT))
9     #     print("Frame width:", frame_width, "Frame height:", frame_height)
10    # except:
11    #     pass
12    output = cv2.VideoWriter('output.mp4', fourcc, 25.0, (640,360))
13
14    while video.isOpened():
15        ret, frame = video.read()
16        if ret == True:
17            output.write(frame)
18            cv2.imshow('frame',frame)
19
20            if cv2.waitKey(1) & 0xFF == ord('q'):
21                break
22            else:
23                break
24
25    cv2.destroyAllWindows()
26
```

Process finished with exit code 0

Task 18: Accessing the webcam

The screenshot shows the PyCharm IDE interface. On the left is the Project tree with files like main.py, Edge_selection.py, flip.py, gaussian_blur.py, highway.png, image_thresholding.py, img_rotate.py, main.py, Median_blur.py, morphological_ops.py, output.mp4, reading_video.py, resize_aspect.py, shape_text.py, shifting_image.py, writing_video.py, and a screenshot file. The webcam.py file is open in the editor. It contains Python code for capturing frames from a webcam. A live video feed window titled 'Live' is displayed, showing a hand holding a white mug with text on it. The status bar at the bottom shows the date and time as 01-02-2026.

```
1 import cv2
2
3 cap = cv2.VideoCapture(0)
4
5 while cap.isOpened():
6     ret, frame = cap.read()
7     if ret == True:
8         cv2.imshow('frame',frame)
9         if cv2.waitKey(1) & 0xFF == ord('q'):
10            break
11
12 cv2.destroyAllWindows()
```

C:\Users\bikas\AppData\Local\Programs\Python\Python312\python.exe E:\Projects\Tracker\TuteDude\Python\Assignment11\webcam.py