

Practical-2

Student Name: Gauri Prabhakar

UID: 18BCS6201

Branch: 18AITAIML-2

Section/Group: B

Semester: 7

Date of Performance: 3rd September, 2021

Subject Name: Computer Vision Lab

Subject Code: CSF - 432

1. Aim/Overview of the practical:

To demonstrate the use of different image processing functions using python and OpenCv.

2. Task to be done:

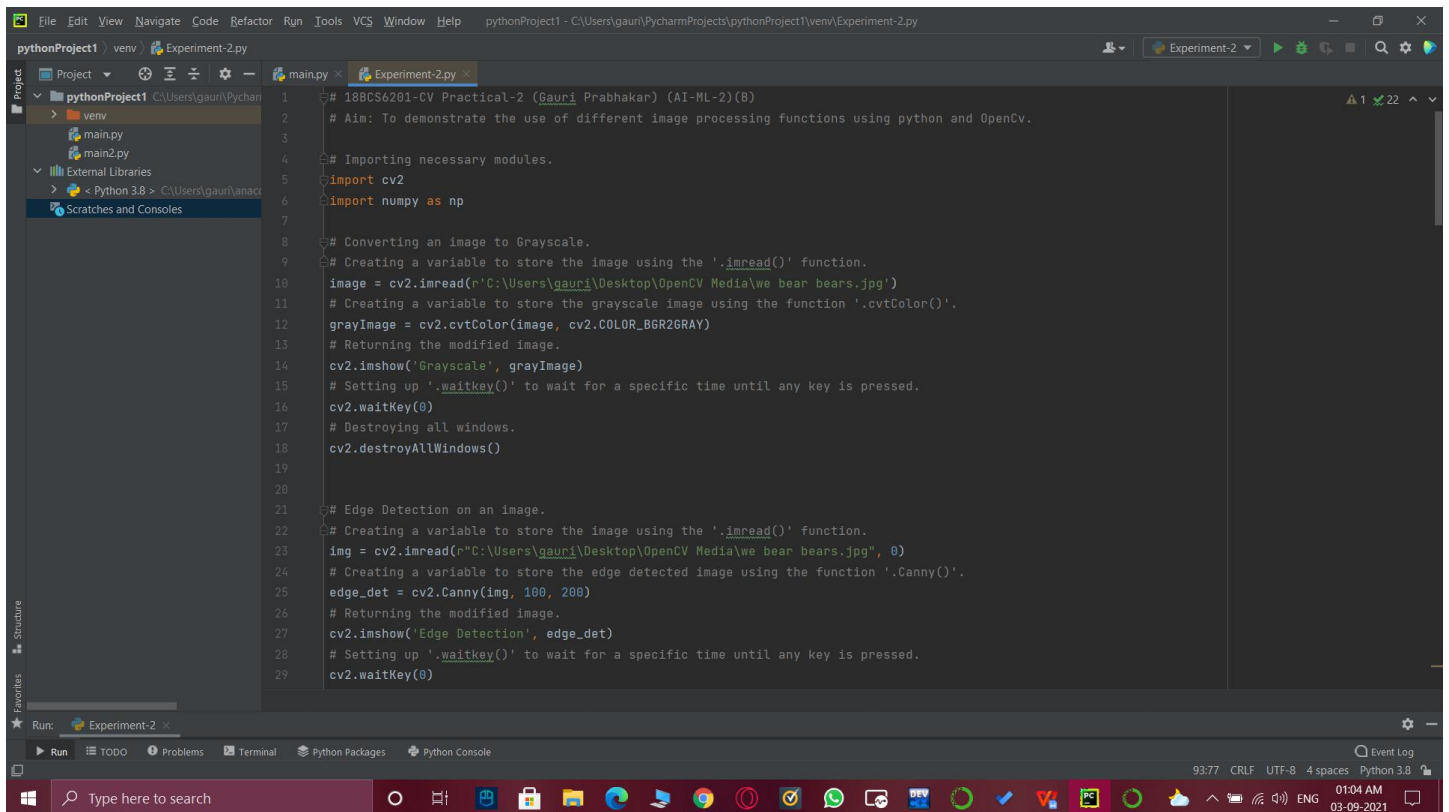
To demonstrate the use of different image processing functions using python and OpenCv and the explanation.

3. Steps to be followed:

1. Importing necessary modules.
2. Converting an image to Grayscale.
3. Creating a variable to store the image using the '.imread()' function.
4. Creating a variable to store the grayscale image using the function '.cvtColor()'.
5. Returning the modified image.
6. Setting up '.waitkey()' to wait for a specific time until any key is pressed.
7. Destroying all windows.
8. Creating a variable to store the edge detected image using the function '.Canny()'.
9. Returning the modified image.
10. Creating a variable to store the average blurred image using the function '.blur()'.
11. Creating a variable to store the median blurred image using the function '.medianBlur()'.
12. Creating a variable to store the gaussian blurred image using the function '.GaussianBlur()'.
13. Returning the image.

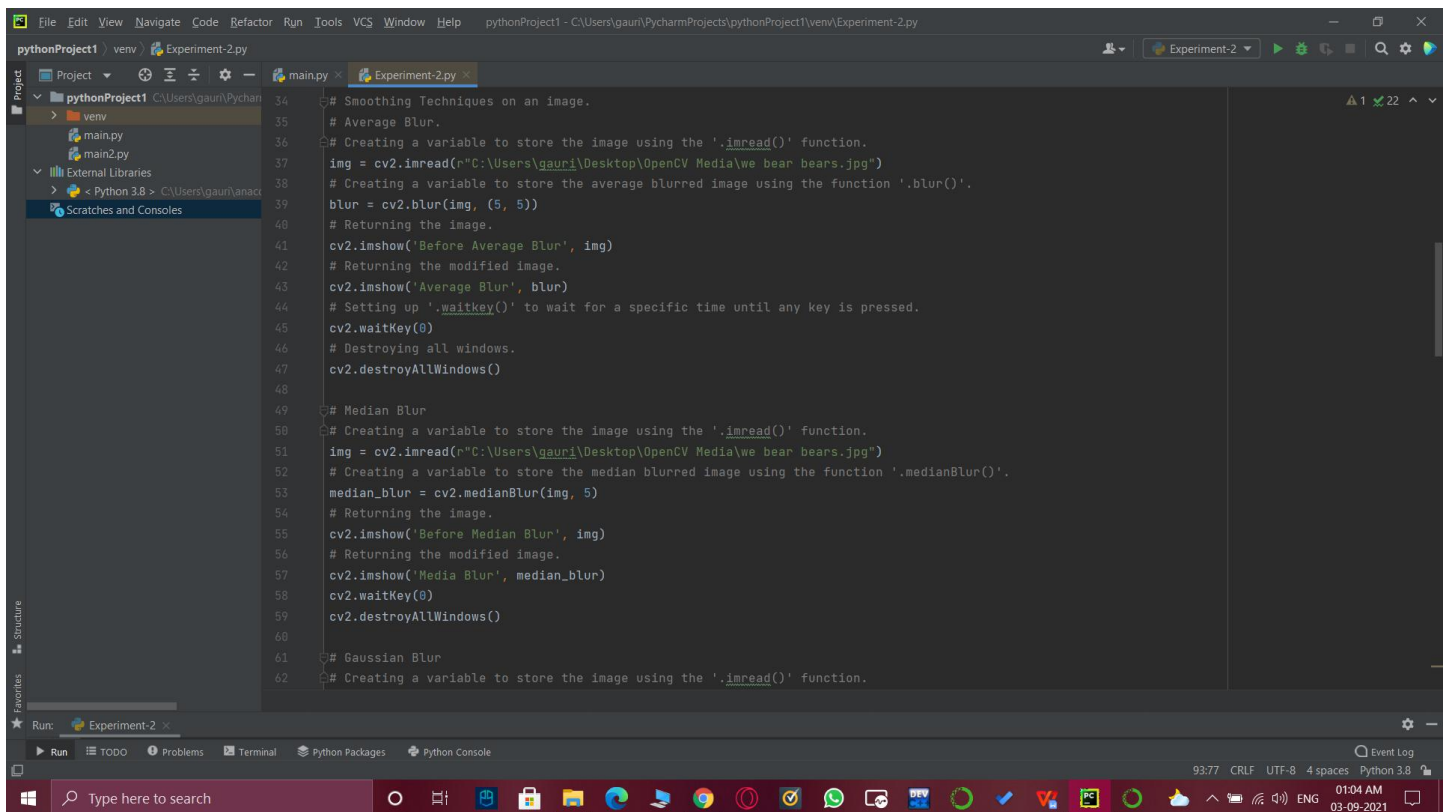
14. Returning the modified image.
15. 'np.ones()' returns an array of 1's of the given shape, data type.
16. Creating a variable to store the eroded image using the function '.erode()'.
17. Creating a variable to store the dilated image using the function '.dilate()'.
18. Returning the image.
19. Returning the modified image.
20. Returning the shape of the image in the form of an array.
21. Resizing the image.
22. Cropping the image.
23. Returning the image.
24. Returning the modified image.
25. Drawing a line on the image, it takes parameters as image, start point, end point, color and finally thickness.
26. Drawing a rectangle on the image, it takes parameters as image, start point, end point, color and finally thickness.
27. Drawing a circle on the image, it takes parameters as image, center co-ordinates, radius, color and finally thickness.
28. Writing a text on the image, it takes parameters as image, text, co-ordinates, font, font scale, color, thickness.
29. Returning the modified image.

4. Result/Output/Writing Summary:



The screenshot shows the PyCharm IDE with a Python script named 'Experiment-2.py'. The script is titled '18BCS6201-CV Practical-2 (Gauri Prabhakar) (AI-ML-2) (B)' and aims to demonstrate image processing functions using Python and OpenCV. The code includes comments and imports for cv2 and numpy. It performs the following steps:

- Importing necessary modules (cv2, numpy as np).
- Converting an image to Grayscale.
- Creating a variable to store the image using the '.imread()' function.
- Creating a variable to store the grayscale image using the function '.cvtColor()'.
- Returning the modified image.
- Displaying the grayscale image using 'cv2.imshow()'.
- Setting up '.waitKey()' to wait for a specific time until any key is pressed.
- Destroying all windows using 'cv2.destroyAllWindows()'.
- Edge Detection on an image.
- Creating a variable to store the image using the '.imread()' function.
- Using 'cv2.imread()' to load the image.
- Creating a variable to store the edge detected image using the function '.Canny()'.
- Returning the modified image.
- Displaying the edge detected image using 'cv2.imshow()'.
- Setting up '.waitKey()' to wait for a specific time until any key is pressed.
- Destroying all windows using 'cv2.destroyAllWindows()'.



The screenshot shows the PyCharm IDE with the continuation of the Python script 'Experiment-2.py'. The script continues with the following steps:

- Smoothing Techniques on an image.
- Average Blur.
- Creating a variable to store the image using the '.imread()' function.
- Using 'cv2.imread()' to load the image.
- Creating a variable to store the average blurred image using the function '.blur()'.
- Returning the image.
- Displaying the image before average blur using 'cv2.imshow()'.
- Returning the modified image.
- Displaying the average blurred image using 'cv2.imshow()'.
- Setting up '.waitKey()' to wait for a specific time until any key is pressed.
- Destroying all windows using 'cv2.destroyAllWindows()'.
- Median Blur.
- Creating a variable to store the image using the '.imread()' function.
- Using 'cv2.imread()' to load the image.
- Creating a variable to store the median blurred image using the function '.medianBlur()'.
- Returning the image.
- Displaying the image before median blur using 'cv2.imshow()'.
- Returning the modified image.
- Displaying the median blurred image using 'cv2.imshow()'.
- Setting up '.waitKey()' to wait for a specific time until any key is pressed.
- Destroying all windows using 'cv2.destroyAllWindows()'.
- Gaussian Blur.
- Creating a variable to store the image using the '.imread()' function.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1 - C:\Users\gaun\PcharmProjects\pythonProject1\venv\Experiment-2.py
pythonProject1 venv Experiment-2.py
Project pythonProject1 C:\Users\gaun\PcharmProjects\pythonProject1\venv
  venv
    main.py
    main2.py
  External Libraries
  > < Python 3.8 > C:\Users\gaun\anaconda3
  Scratches and Consoles
main.py Experiment-2.py
60
61 # Gaussian Blur
62 # Creating a variable to store the image using the '.imread()' function.
63 img = cv2.imread(r"C:\Users\gaun\Desktop\OpenCV Media\we bear bears.jpg")
64 # Creating a variable to store the gaussian blurred image using the function '.GaussianBlur()'.
65 Gaussian_blur = cv2.GaussianBlur(img, (5, 5), 0)
66 # Returning the image.
67 cv2.imshow('Before Gaussian Blur', img)
68 # Returning the modified image.
69 cv2.imshow('Gaussian Blur', median_blur)
70 cv2.waitKey(0)
71 cv2.destroyAllWindows()
72
73
74 # Erosion on an image.
75 # Creating a variable to store the image using the '.imread()' function.
76 img = cv2.imread(r"C:\Users\gaun\Desktop\OpenCV Media\we bear bears.jpg")
77 # 'np.ones()' returns an array of 1's of the given shape, data type.
78 kernel = np.ones((5, 5), np.uint8)
79 # Creating a variable to store the eroded image using the function '.erode()'.
80 img_erosion = cv2.erode(img, kernel, iterations=1)
81 # Returning the image.
82 cv2.imshow('Before Erosion', img)
83 # Returning the modified image.
84 cv2.imshow('Erosion', img_erosion)
85 cv2.waitKey(0)
86 cv2.destroyAllWindows()
87
88 # Dilation on an image.
```


```
File Edit View Navigate Code Refactor Run Tools VCS Window Help pythonProject1 - C:\Users\gaun\PcharmProjects\pythonProject1\venv\Experiment-2.py
pythonProject1 venv Experiment-2.py
Project pythonProject1 C:\Users\gaun\PcharmProjects\pythonProject1\venv
  venv
    main.py
    main2.py
  External Libraries
  > < Python 3.8 > C:\Users\gaun\anaconda3
  Scratches and Consoles
main.py Experiment-2.py
87
88 # Dilation on an image.
89 # Creating a variable to store the image using the '.imread()' function.
90 img = cv2.imread(r"C:\Users\gaun\Desktop\OpenCV Media\we bear bears.jpg")
91 # 'np.ones()' returns an array of 1's of the given shape, data type.
92 kernel = np.ones((5, 5), np.uint8)
93 # Creating a variable to store the dilated image using the function '.dilate()'.
94 img_dilation = cv2.dilate(img, kernel, iterations=1)
95 # Returning the image.
96 cv2.imshow('Before Dilation', img)
97 # Returning the modified image.
98 cv2.imshow('Dilation', img_dilation)
99 cv2.waitKey(0)
100 cv2.destroyAllWindows()
101
102
103 # Cropping and Resizing on an image.
104 # Creating a variable to store the image using the '.imread()' function.
105 img = cv2.imread(r"C:\Users\gaun\Desktop\OpenCV Media\we bear bears.jpg")
106 # Returning the shape of the image in the form of an array.
107 print(img.shape)
108 # Resizing the image.
109 imgResize = cv2.resize(img, (1000, 500))
110 # Returning the shape of the modified image.
111 print(imgResize.shape)
112 # Cropping the image.
113 imgCropped = img[0:200, 200:500]
114 # Returning the image.
115 cv2.imshow("Before Resizing and Cropping", img)
116 # Returning the modified image.
```

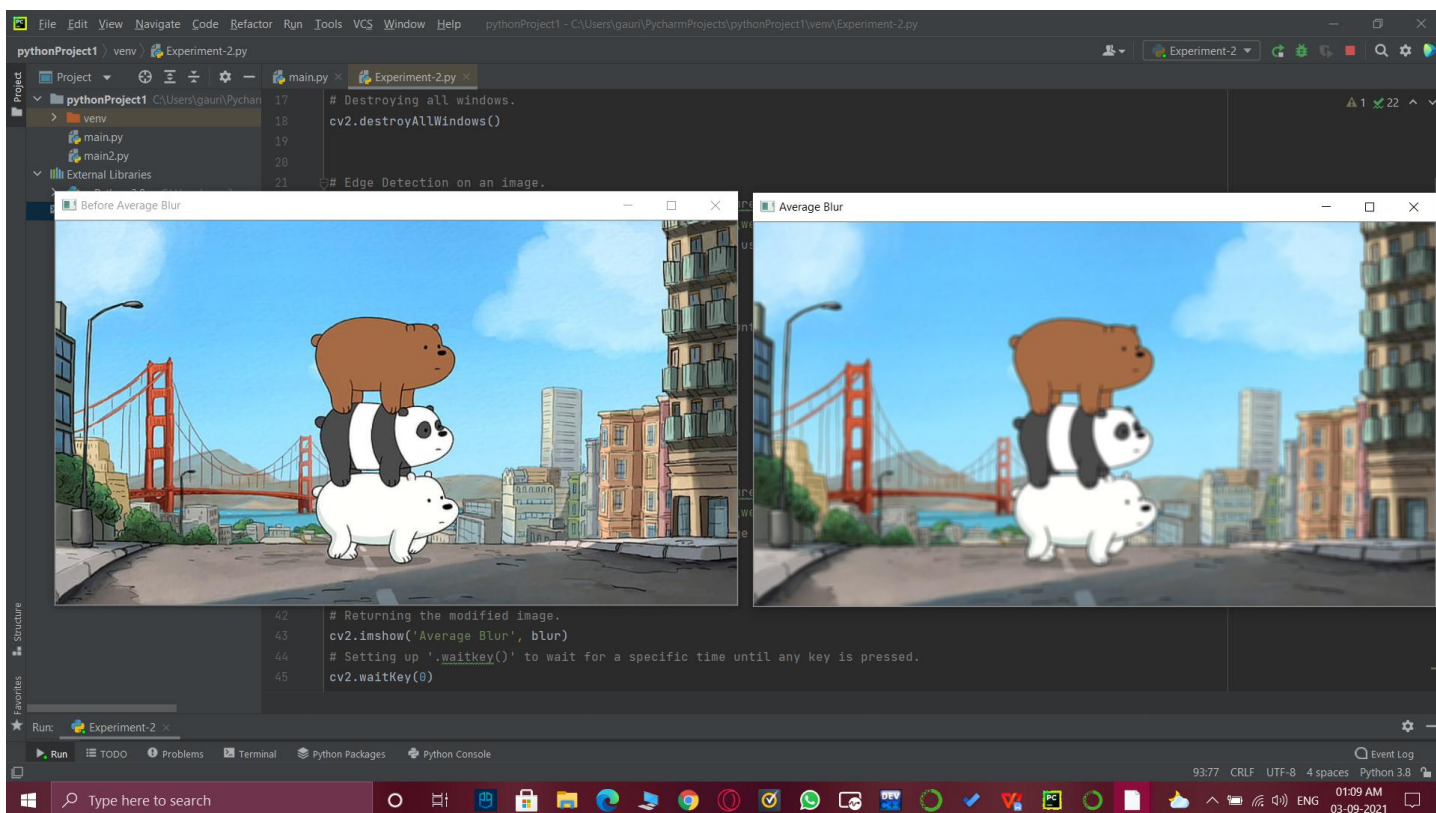
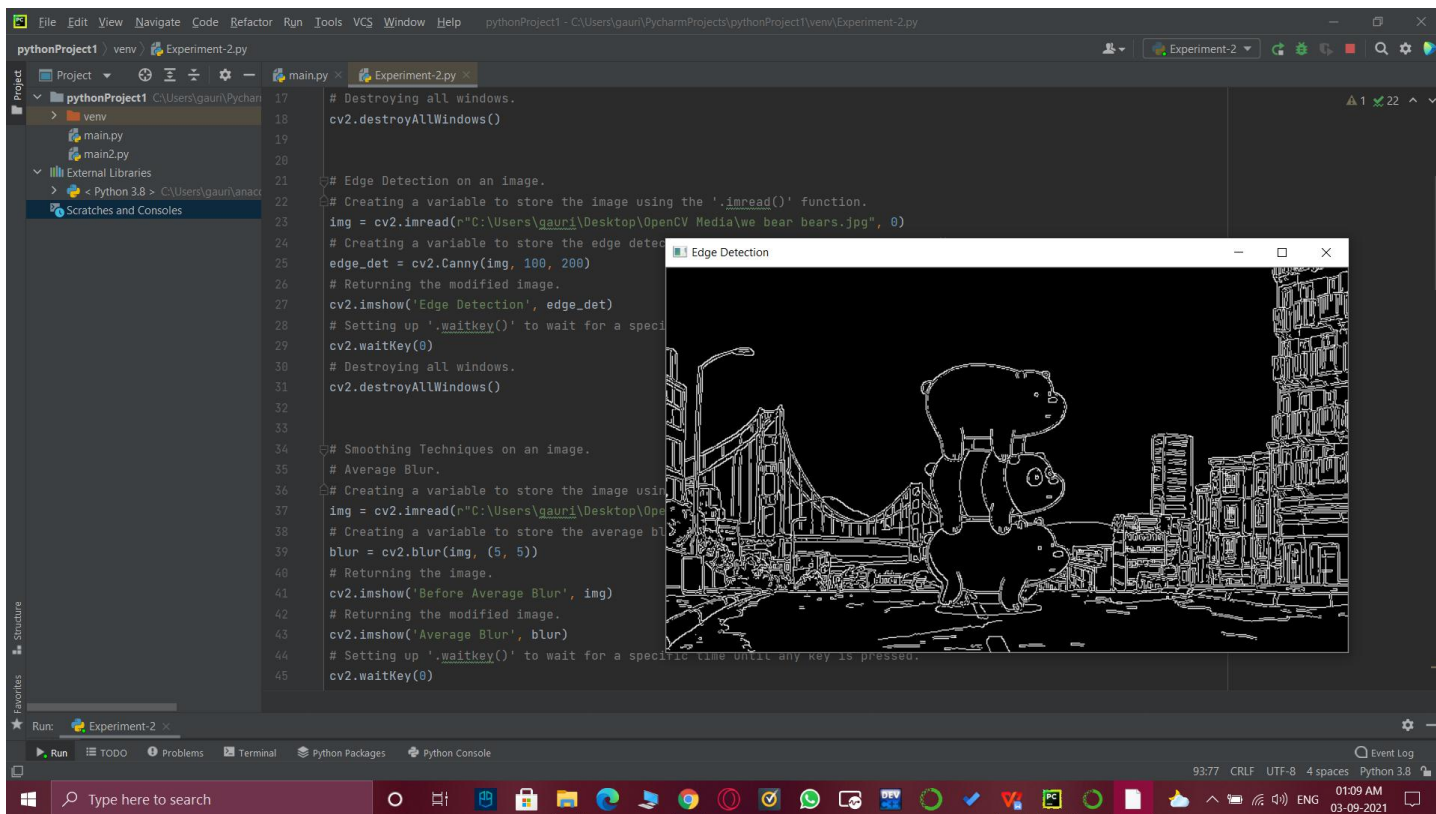
```
pythonProject1 - C:\Users\gaurn\PycharmProjects\pythonProject1\venv\Experiment-2.py
pythonProject1 venv Experiment-2.py
Project
  pythonProject1 C:\Users\gaurn\PycharmProjects\pythonProject1\venv\
    venv
      main.py
      main2.py
    External Libraries
      < Python 3.8 > C:\Users\gaurn\anaconda3
    Scratches and Consoles
Structure
  pythonProject1
    venv
      main.py
      main2.py
    External Libraries
      < Python 3.8 > C:\Users\gaurn\anaconda3
    Scratches and Consoles
Run: Experiment-2
Run TODO Problems Terminal Python Packages Python Console
93:77 CRLF UTF-8 4 spaces Python 3.8
Type here to search
```

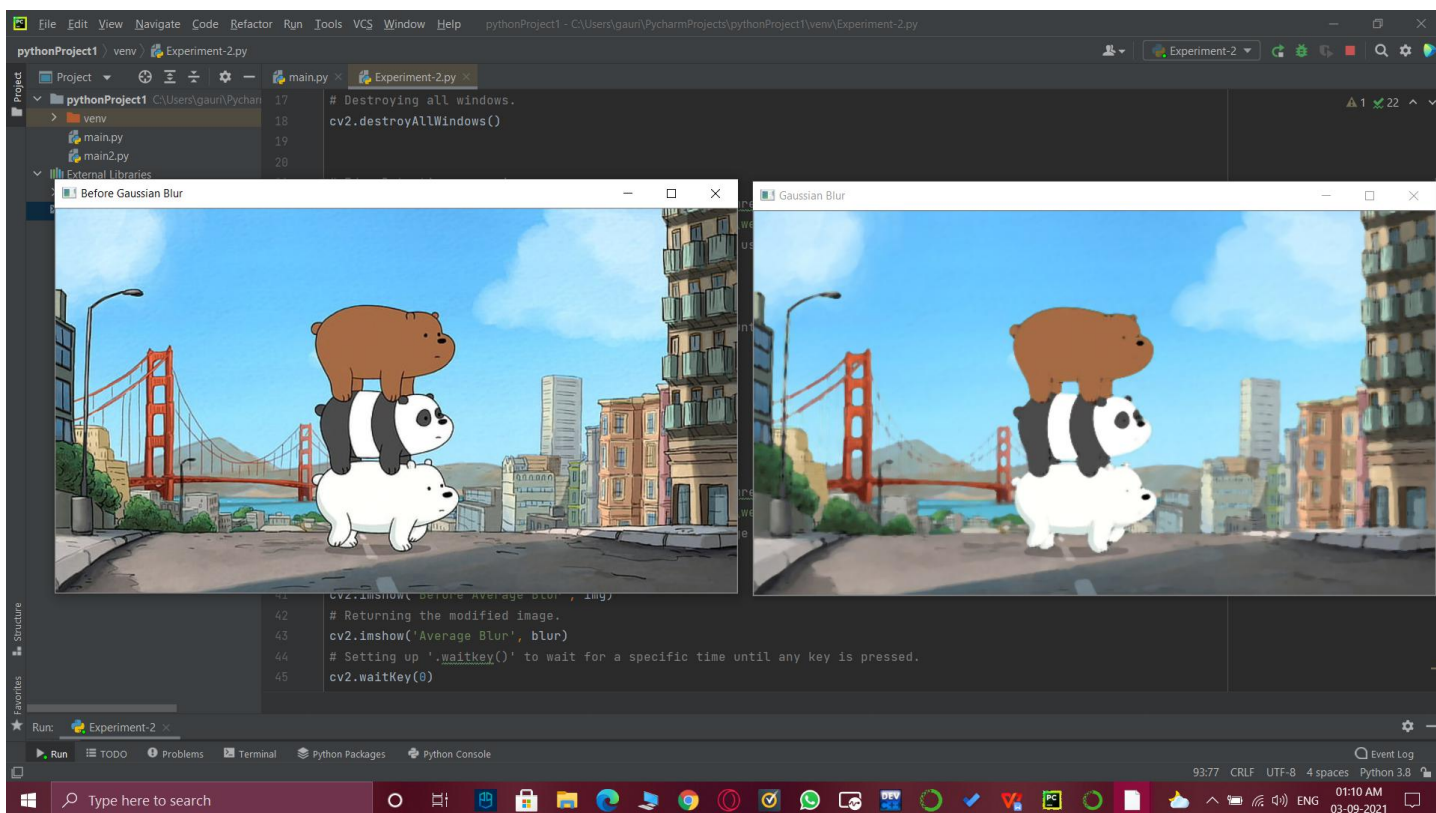
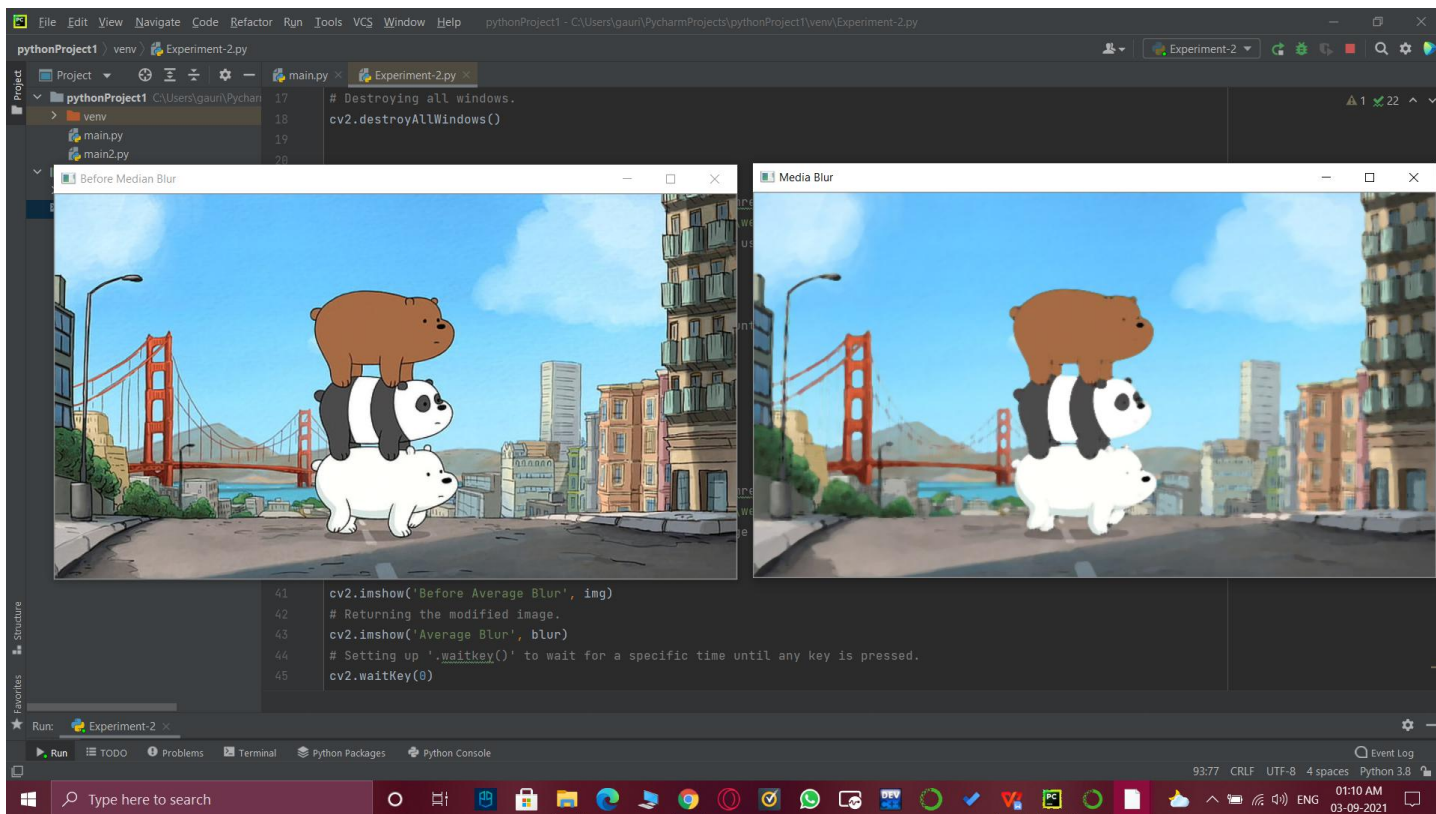
```
113 imgCropped = img[0:200, 200:500]
114 # Returning the image.
115 cv2.imshow("Before Resizing and Cropping", img)
116 # Returning the modified image.
117 cv2.imshow("Image Resize", imgResize)
118 # Returning the modified image.
119 cv2.imshow("Image Cropped", imgCropped)
120 cv2.waitKey(0)
121 cv2.destroyAllWindows()
122
123
124 # Shape and Text on an image.
125 # 'np.ones()' returns an array of 1's of the given shape, data type.
126 img = np.zeros((512, 512, 3), np.uint8)
127 # Returning the image.
128 print('Before Modifications', img)
129 img[:] = 204, 204, 255
130 # Drawing a line on the image, it takes parameters as image, start point, end point, color and finally thickness.
131 cv2.line(img, (0, 0), (img.shape[1], img.shape[0]), (204, 0, 102), 2)
132 # Drawing a rectangle on the image, it takes parameters as image, start point, end point, color and finally thickness.
133 cv2.rectangle(img, (0, 0), (300, 350), (64, 64, 64), 2)
134 # Drawing a circle on the image, it takes parameters as image, center co-ordinates, radius, color and finally thickness.
135 cv2.circle(img, (400, 50), 30, (255, 255, 0), 5)
136 # Writing a text on the image, it takes parameters as image, text, co-ordinates, font, font scale, color, thickness.
137 cv2.putText(img, "Gauri Prabhakar", (200, 150), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255, 153, 51), 2)
138 # Returning the modified image.
139 cv2.imshow("Shape and Text", img)
140 cv2.waitKey(0)
141 cv2.destroyAllWindows()
```

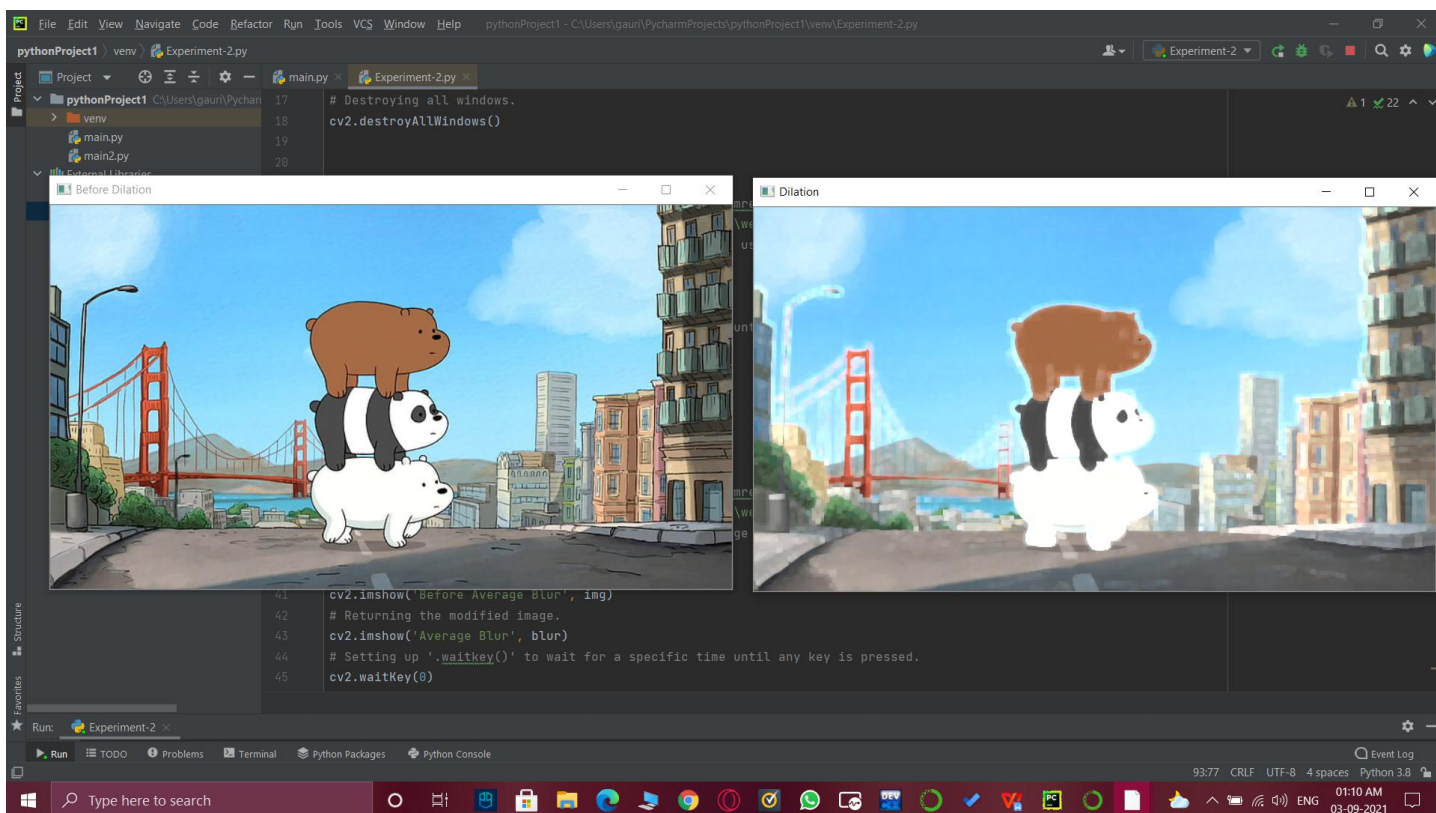
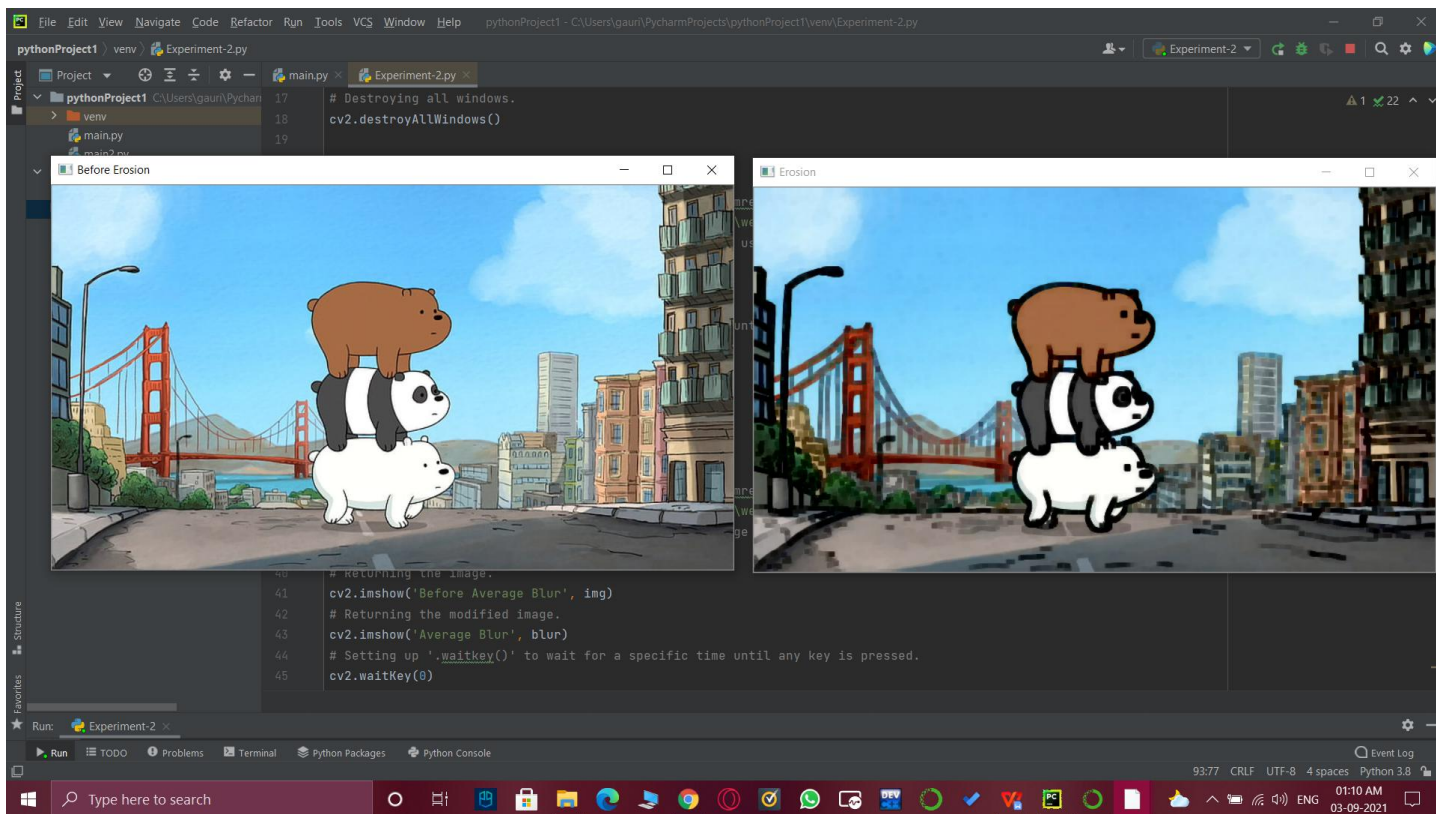
```
pythonProject1 - C:\Users\gaurn\PycharmProjects\pythonProject1\venv\Experiment-2.py
pythonProject1 venv Experiment-2.py
Project
  pythonProject1 C:\Users\gaurn\PycharmProjects\pythonProject1\venv\
    venv
      main.py
      main2.py
    External Libraries
      < Python 3.8 > C:\Users\gaurn\anaconda3
    Scratches and Consoles
Structure
  pythonProject1
    venv
      main.py
      main2.py
    External Libraries
      < Python 3.8 > C:\Users\gaurn\anaconda3
    Scratches and Consoles
Run: Experiment-2
Run TODO Problems Terminal Python Packages Python Console
93:77 CRLF UTF-8 4 spaces Python 3.8
Type here to search
```

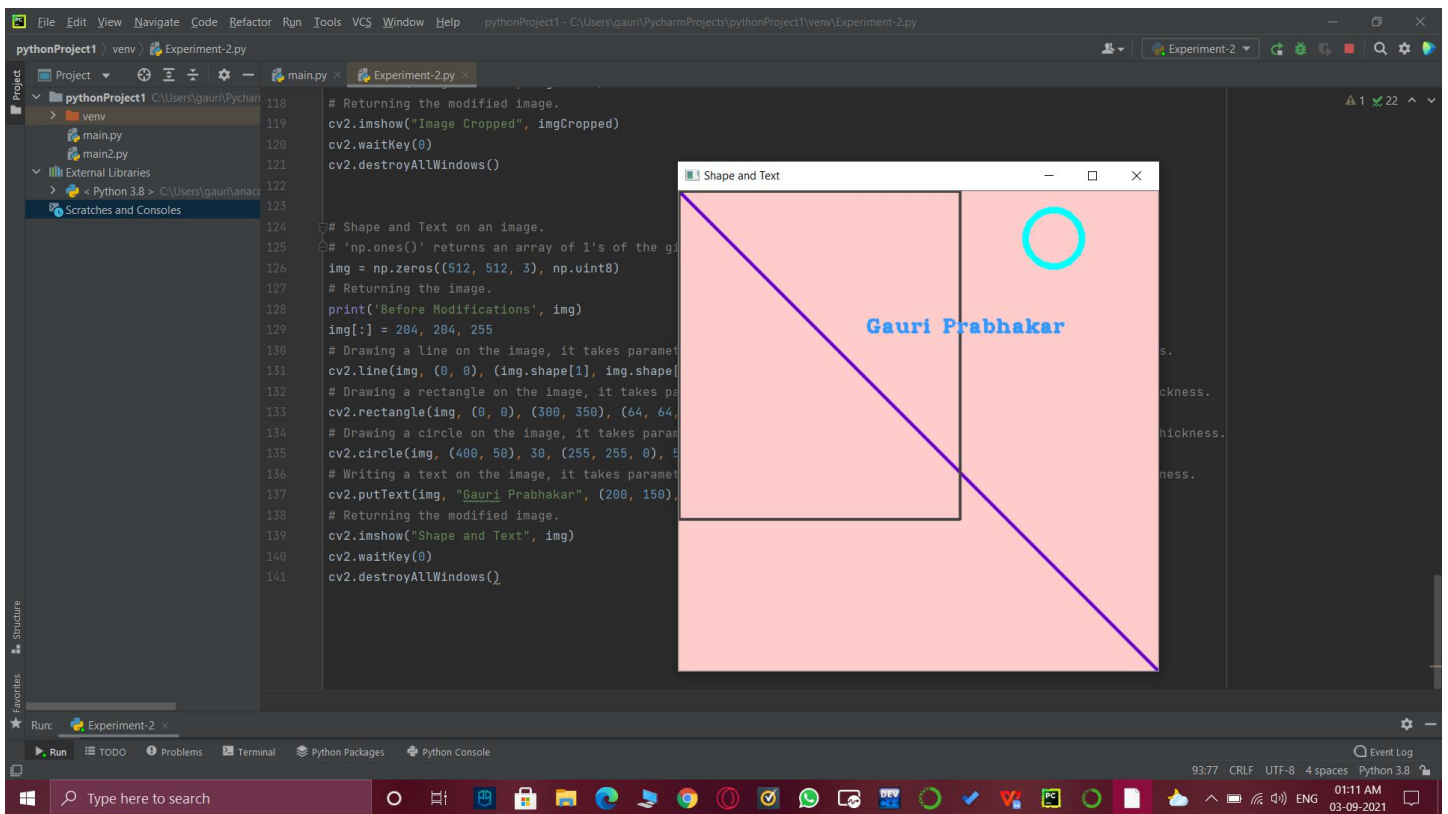
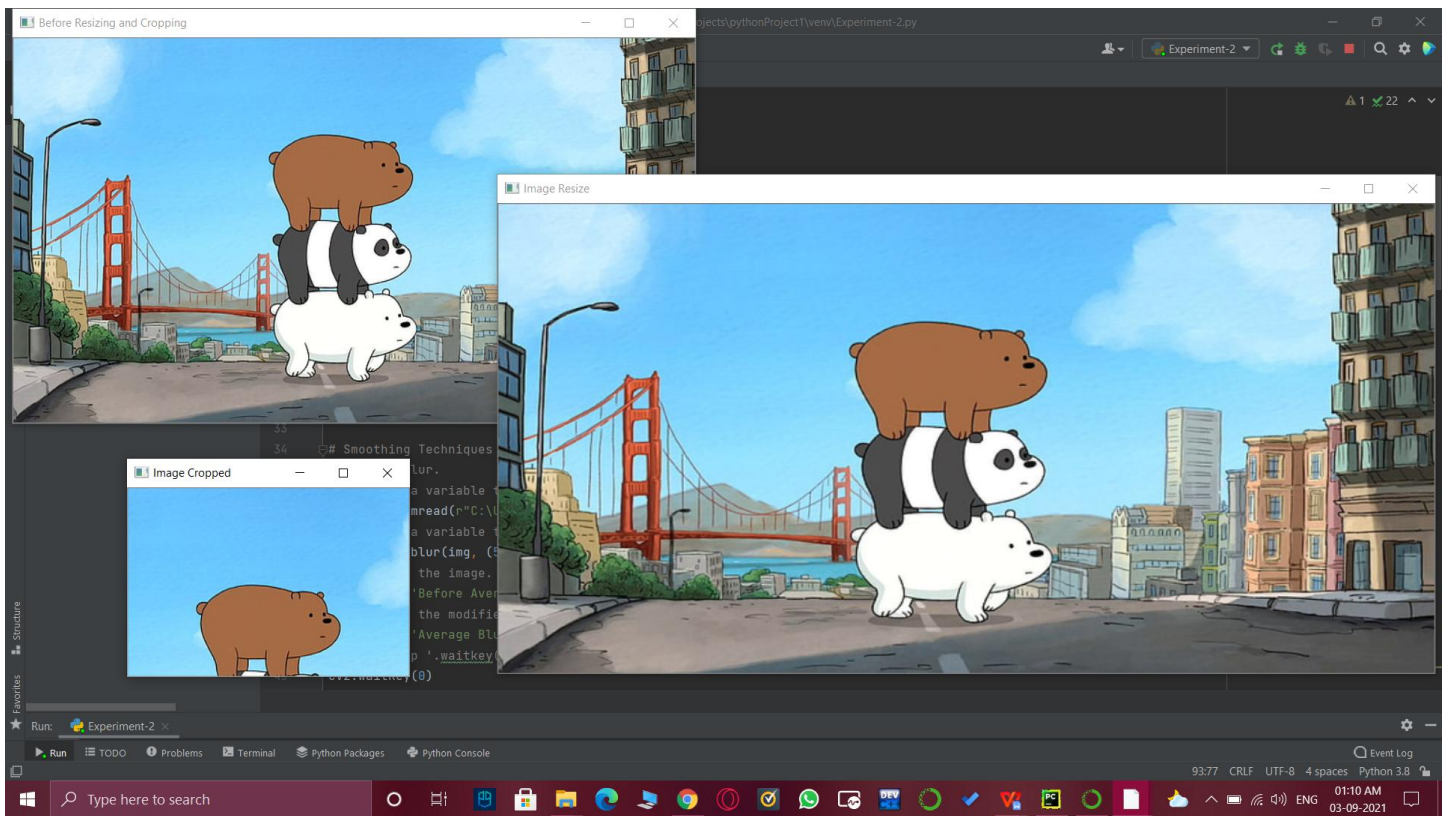
```
1 # 18BCS6201-CV Practical-2 (Gauri Prabhakar) (AI-ML-2)(B)
2 # Aim: To demonstrate the use of different image processing functions using python and OpenCv.
3
4 # Importing necessary modules.
5 import cv2
6 import numpy as np
7
8 # Converting an image to grayscale.
9 # Creating a variable to store the original image.
10 image = cv2.imread(r"C:\Users\gaurn\PycharmProjects\pythonProject1\venv\Images\panda.jpg")
11 # Creating a variable to store the grayscale image.
12 grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
13 # Returning the modified image.
14 cv2.imshow('Grayscale', grayImage)
15 # Setting up '.waitkey()' to wait for a specific time until any key is pressed.
16 cv2.waitKey(0)
17 # Destroying all windows.
18 cv2.destroyAllWindows()
19
20
21 # Edge Detection on an image.
22 # Creating a variable to store the original image.
23 img = cv2.imread(r"C:\Users\gaurn\PycharmProjects\pythonProject1\venv\Images\panda.jpg")
24 # Creating a variable to store the edge detected image.
25 edge_det = cv2.Canny(img, 1, 30)
26 # Returning the modified image.
27 cv2.imshow('Edge Detection', edge_det)
28 # Setting up '.waitkey()' to wait for a specific time until any key is pressed.
29 cv2.waitKey(0)
```











5. Learning outcomes (What I have learnt):

- Open CV modules.
- Grayscale images, edge detection.
- Average, Median and Gaussian Blur.
- Erosion and Dilation of an image.
- Cropping and resizing an image.
- Different shapes and text on an image.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			

