Practical-4

Student Name: Gauri Prabhakar
UID: 18BCS6201
Branch: 18AITAIML-2
Section/Group: B
Semester: 7
Date of Performance: 9th September, 2021
Subject Name: Computer Vision Lab
Subject Code: CSF - 432

1. Aim/Overview of the practical:

To detect contours and shape drawn within a given image using python and OpenCV.

2. Task to be done:

To detect contours and shape drawn within a given image using python and OpenCV and the explanation.

3. Steps to be followed:

1. Importing necessary modules.

2. Importing custom module.

3. Defining a function to detect contours.

4. Finding contours in a binary image using the function 'findContours()'.

5. It takes the image, a flag which returns only the parents contours (contour retrieval mode) and 'CHAIN_APPROX_NONE' to store all the contour points.

6. Sometimes there are shapes within shapes (parent and children) so to depict the relationship we use hierarchy.

7. Iterating through the contours.

8. Storing the contour area.

9. Returning the sub-image.

10. Storing the contour area.

11. Returning the contour area.

12. If area is greater than 500 we enter the loop to draw contour.

13. Drawing contour using the 'drawContours()' function.

14. It takes the image, list of contour points, index of contours, color of the contour and thickness of the contour as arguments.

15. Creating a variable to store the arclength of the contours using the function 'arcLength()'.

16. Returning the arc length.

17. Approximating the shape of the contours.

18. Returning the length of approx.

19. Creating a variable to store the number of items in 'approx'.

20. Drawing an approximate rectangle around the binary image.

21. Checking the number of items in the 'approx' to classify the contour into different shapes.

22. Setting up the properties of the rectangle drawn above.

23. Inserting text to label contours.

24. Defining the function 'stackImages()' to stack input images.

25. Using 'len()' to return the number of items in the 'imgArray' object which is used to store 1-D and 2-D images as an array.

26. Returning the number of rows.

27. Returning the number of columns.

28. Returning the image array in literal format.

29. Checking if we have a multilayer image.

30. The 'isinstance()' function returns true or false.

31. It takes the the columns and the list as an argument.

32. Storing the width and height of the image array.

33. Returning the width and height of the image array.

34. If 'rowsAvailable' evaluates to True:

35. Horizontally stacking the image.

36. Vertically stacking the image.

37. If 'rowsAvailable' evaluates to False:

38. Horizontally stacking the image.

39. Vertically stacking the image.

40. Writing driver code to trigger the stacks and perform contouring.

41. Creating a variable path which stores the path of the target image.

42. Creating a variable to store the image using the '.imread()' function.

43. Creating a variable to store the grayscale image using the function '.cvtColor()'.

44. Creating a variable to store the gaussian blurred image using the function '.GausianBlur()'.

45. Creating a variable to store the edge detected image using the function '.Canny()'.

46. Retrieving contours from the edge detected image.

47. Creating a blank image.

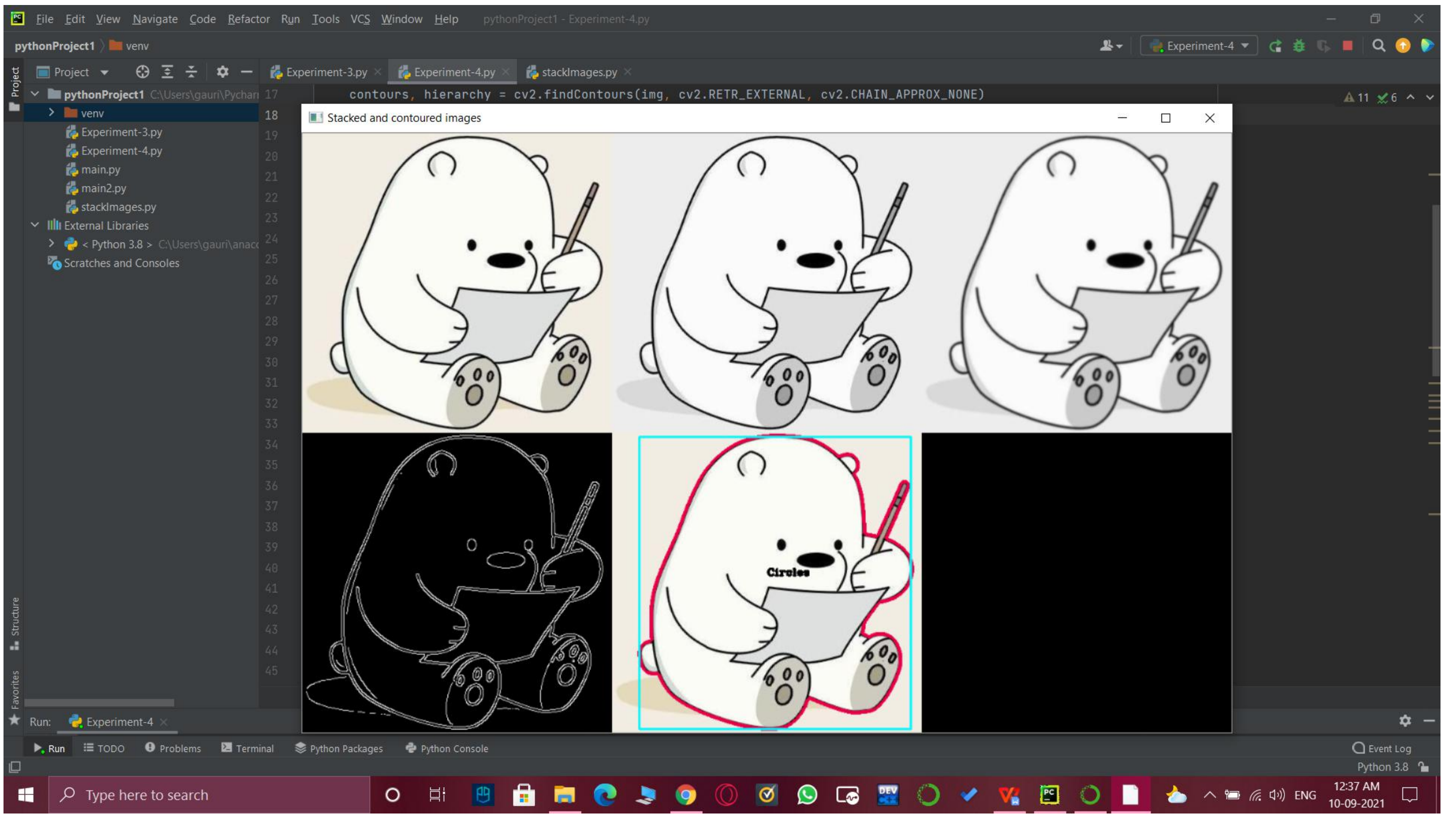
48. Stacking the images using 'stackImages.py'.

49. Returning the stacked and contoured images.

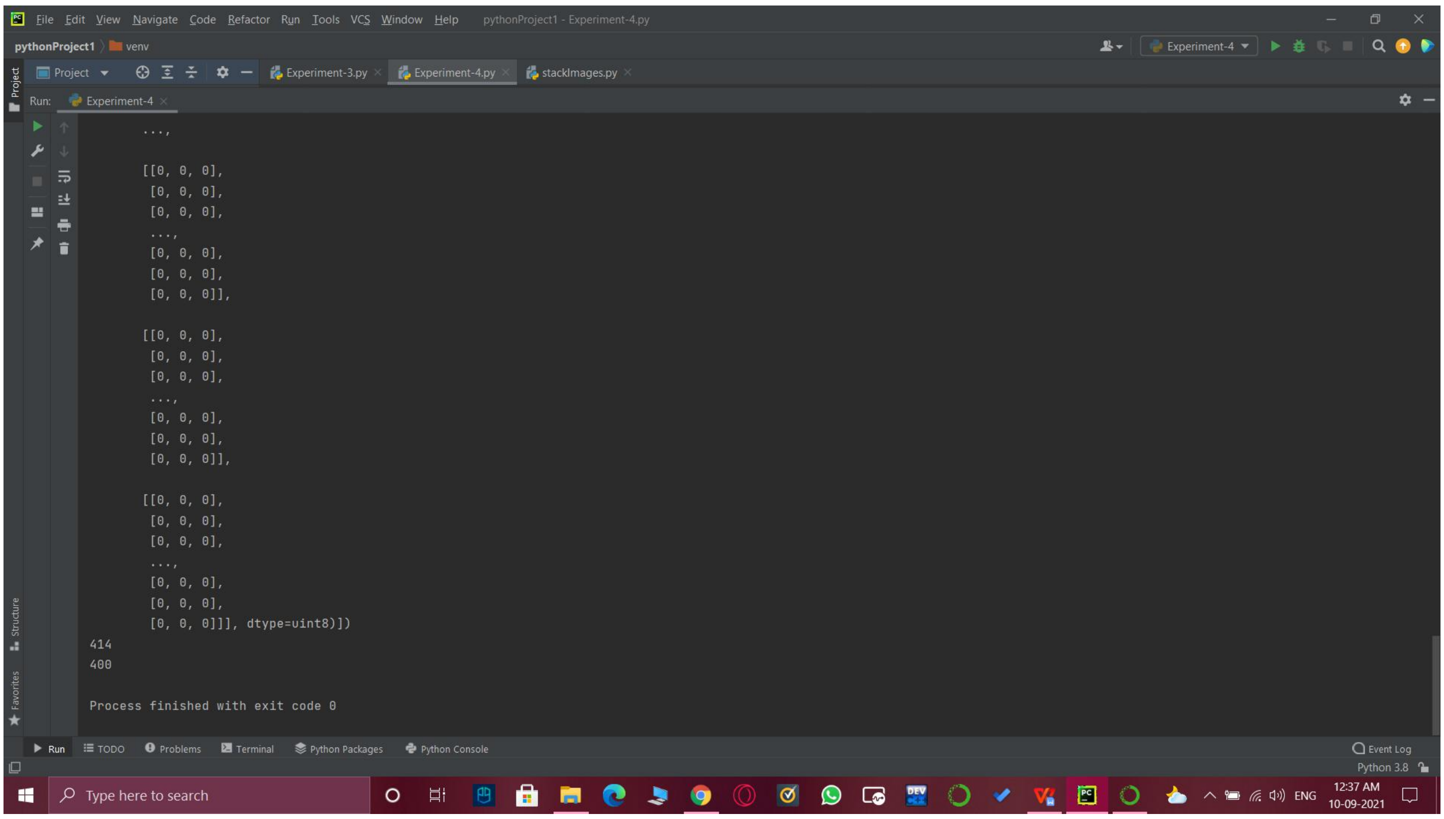
50. Setting up '.waitkey()' to wait for a specific time until any key is pressed.

51. Destroying all windows.

4. Result/Output/Writing Summary:



```
contours, hierarchy = cv2.findContours(img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
```



```
        ...,

        [[0, 0, 0],
         [0, 0, 0],
         [0, 0, 0],
         ...,
         [0, 0, 0],
         [0, 0, 0],
         [0, 0, 0]],

        [[0, 0, 0],
         [0, 0, 0],
         [0, 0, 0],
         ...,
         [0, 0, 0],
         [0, 0, 0],
         [0, 0, 0]],

        [[0, 0, 0],
         [0, 0, 0],
         [0, 0, 0],
         ...,
         [0, 0, 0],
         [0, 0, 0],
         [0, 0, 0]]], dtype=uint8)])
414
400

Process finished with exit code 0
```

5. Learning outcomes (What I have learnt):

- Open CV modules.
- Detect contours and label them.
- What is hierarchy.
- Grayscale images.
- Gaussian Blur.
- Edge Detection.
- How to stack images.
- Vertically, horizontally stacked images.

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.			