Text recognition using OpenCV

Submission Document:

Team B:

Bhavina Chechani

Bhumika Chechani

Harekrishna Ray

Riken Prajapati

Index

1. Introduction to OpenCV

- OpenCV[1][2] is an open source library which includes several computer vision algorithms. It is basically a computer vision and machine learning library.
- It supports languages like python, c++, java etc and is available on different platforms including Windows, Linux, OS X, Android, and iOS.
- It was started by Intel in 1999. It was later supported by Willow Garage and now maintained by Itseez.
- OpenCV-python is a library in python to solve computer vision problems.
- It can also be said as an image processing library.
- Using it, one can process images and videos to identify objects, faces, or even handwriting of a human and many more.

2. Image Processing

- It is a method to perform some operation on an image to extract any useful information from the image.
- Image Processing[4] is the analysis of a digitized image, especially in order to improve its quality.
- It is a signal processing which has input as image and output is image or characteristics according to requirement.
- Image is just a two-dimensional matrix, defined by the mathematical function f(x, y) at any point which gives the pixel value at that point.
- Where x and y are plane coordinates and amplitude, which tells the intensity or grey level of the image at that point.
- Pixel value describes how bright the image is and colour.
- Computer reads the image as a range of the values between 0 and 255.
- And for colour image, there are 3 primary channels i.e red, green and blue.

Terms regarding image processing:

- **Object recognition:** It is a computer vision technique to define objects in images or videos.It consists of identifying, recognising and locating objects within an image.
- This process have three main task:
  - Classification**:** Algorithm produces a list of object categories present in an image.
  - Single-object localization**:** With list of objects it also shows axis-aligned bounding box indicating position and scale of one instance of each object category.
  - Detection**:** With list of objects it also shows axis-aligned bounding box indicating position and scale of every instance of each object category.
- **Colour Detection:** Colour detection algorithm identifies image pixel in image which match a specific colour or range of colour. It can be done with RGB color space.
- Steps:

    - Define the upper and lower limits for your pixel values.
    - Function cv2.inRange method which returns a mask, specifying which pixel falls into the specified lower and upper range.
    - Then to apply this on our image cv2,bitwise_and function is used.

- **Morphological Processing:** In particular, noises and textures can be distorted when the binary region is produced by a simple threshold, so it helps in removing the imperfections in detecting the object.

3. Shape Detection

- [5]Shape detection is an important and basic part of image processing.
- It refers to modules that deal with identifying and detecting shapes of parts of an image.
- It includes detecting images which differ in brightness, color or texture.
- Following is the code for image detection , finding its color, area and centroid. https://github.com/Abhiyanta-Community/Trainee-B-JAN2021/blob/Task-2.1/Riken/task%202.1.py

Code Explanation

- Reading an image using imread() function.
- cvtColor() function is used to convert the image to grayscale from BGR. Using this function we can convert images in required space like HSV, GRAY etc.
- Using canny() function we find the edges of the image, it takes 3 parameters i.e the source image, max and min threshold.
- Now contour is the curve formed by joining all the continuous points along the boundary and findContours() and drawContours() function is to detect and draw the contours.
- Finding Shape type : Firstly finding the perimeter of contour by using function arcLength(). This obtained perimeter is passed to approxPolyDP() function which finds the vertices of the shape which can be compared and accordingly the shape is detected, for example 4:square, 3: triangle etc.
- Finding Centroid : First finding the moment using moment() function and then can find centroid using the moments.
- Finding Area : Using contourArea() function we find the area the parameter passed is the contour obtained by findContour() function.
- Finding Color : Getting the bounding rectangle using boundingRect() around contour then finding average BGR color inside the bounding rectangle using mean() function.

Output

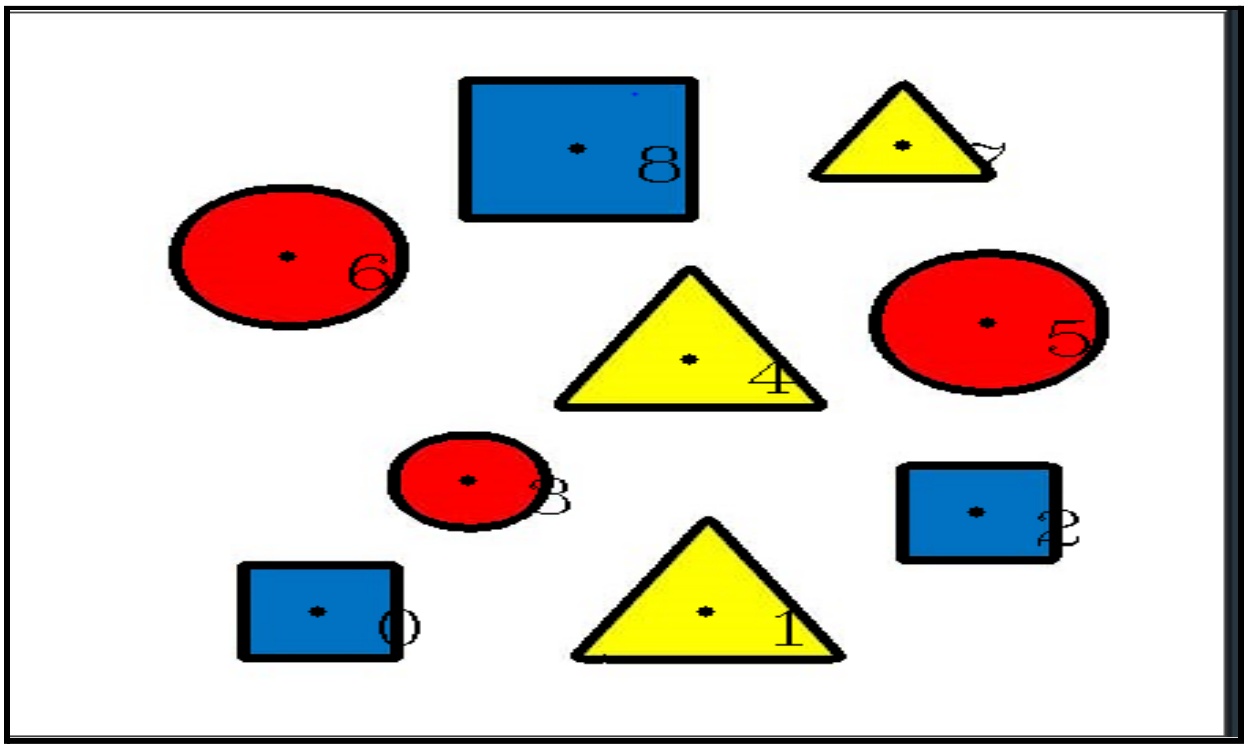

Fig1: Output for test case1

```
Console 1/A
ShapeCentroid: [372, 186]
ShapeArea: 5519.0
6
ShapeType: Circle
ShapeColor[BGR]: [ 59  58 234   0]
ShapeCentroid: [116, 146]
ShapeArea: 5515.5
7
ShapeType: Triangle
ShapeColor[BGR]: [128 226 226   0]
ShapeCentroid: [341, 79]
ShapeArea: 1984.5
8
ShapeType: Square
ShapeColor[BGR]: [179 107   7   0]
ShapeCentroid: [222, 81]
ShapeArea: 6889.0
```

Fig2: Result of test case 1

:

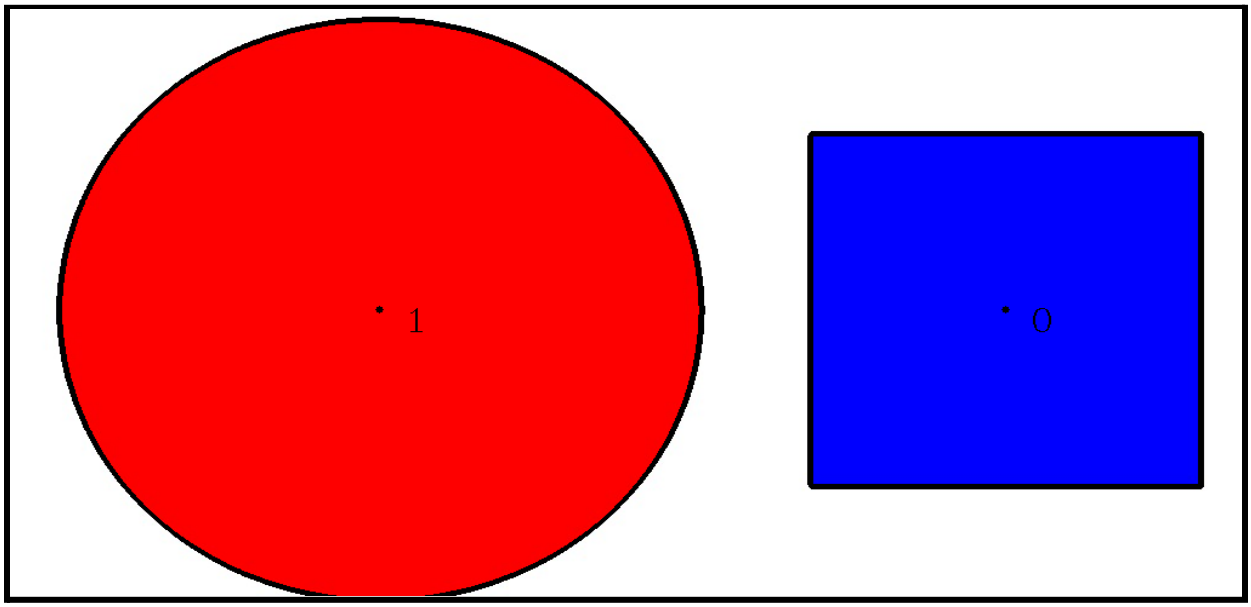

Fig3: Output of test case 2

```
In [2]: runfile('D:/Image processing/Shape detection(1)/task2.1.py',
wdir='D:/Image processing/Shape detection(1)')
0
ShapeType: Square
ShapeColor[BGR]: [252   1   1   0]
ShapeCentroid: [905, 475]
ShapeArea: 103359.0
1
ShapeType: Circle
ShapeColor[BGR]: [ 55  55 253   0]
ShapeCentroid: [389, 475]
ShapeArea: 220076.0
```

Fig4: Result of test case 2

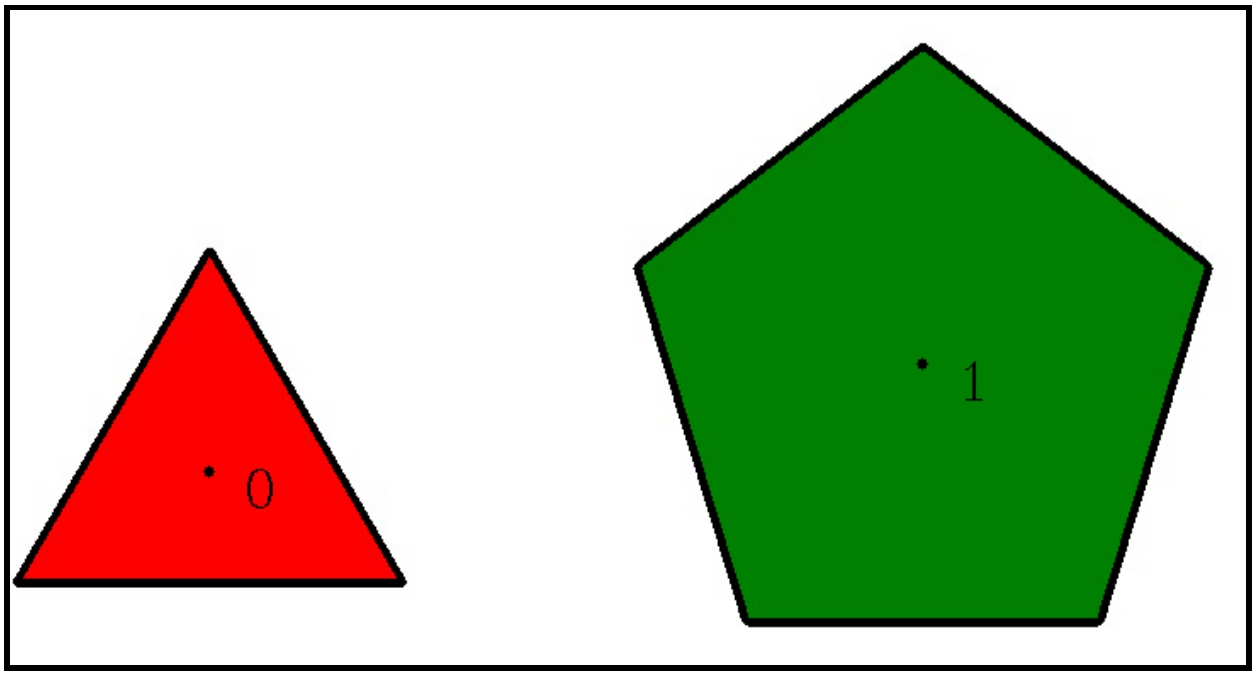

Fig5: Output of test case 3

```
IPython 7.12.0 -- An enhanced Interactive Python.

In [1]: runfile('D:/Image processing/Shape detection(1)/task2.1.py',
wdir='D:/Image processing/Shape detection(1)')
0
ShapeType: Triangle
ShapeColor[BGR]: [127 127 252   0]
ShapeCentroid: [167, 392]
ShapeArea: 21224.0
1
ShapeType: pentagon
ShapeColor[BGR]: [ 80 167  79   0]
ShapeCentroid: [587, 332]
ShapeArea: 74755.0
```

Fig6: Result of test case 3

4. Introduction to Pytesseract

- Tesseract[7][8] is an open source OCR engine.
- The OCR system is used to transform a two dimensional image of text into machine readable text.
- Python-tesseract is an OCR tool for python.
- It reads the text written in image and recognise it whether it be handwritten or computer text.
- It can read all types of image like png, jpeg, etc with the help of Pillow and Leptonica imaging libraries.

5. Text Recognition

- Its main aim is to identify and capture all the unique words using different languages from written text characters.
- Basic steps[3]:
  - **Preprocessing of the Image** : It includes rescaling, binarization, noise removal, deskewing the image so that the required output can be obtained.
  - **Text Localization**: It is a specified form of object detection which automatically computes the bounding boxes for every region of text in an image. That can be done with OCR software.
  - **Character Segmentation**[6] **:** In Character segmentation the text lines, words characters are segmented properly for recognition. We use Hough transform based technique for line and word segmentation from digitized images.
  - **Character Recognition**: It is for recognition of printed or written text characters by a computer
- Following is the link for the program for number plate detection https://github.com/Abhiyanta-Community/Trainee-B-JAN2021/blob/Task-2.2/Riken/task2.2.py

Code Explanation

- Reading the image and resizing it with the help of imutil.resize() function.
- Converts the image to grayscale using cvtColor() function and finds its edges with the help of Canny() function.
- Using findContour() and drawContour() function draw the contour for the whole image.
- Now we sort the contour with respect to the area in descending order, read the first 20 values and draw the contours for new values.
- In for loop first finding perimeter of contour using arcLength() function.
- Now finding edges in contour with the help of approxPolyDP() function.
- Using the convexHull() function we convert the irregular contour to regular contour.
- Find the number of edges with 4 corners and if it is 4 then it enters the if loop.
- Using boundingRect() function find the rectangle across that particular contour and crop the image.
- Now passing this cropped image to pytesseract.image_to_string() function the text can be obtained.

Output

Fig7: Output for case1

Fig8: Output of cropped image for case1

```
In [1]: runfile('D:/Python1/OCR/task2.2.py', wdir='D:/Python1/OCR')
Number is : HR26DK8337
```

Fig7: Output on console for case1

6. Troubleshooting

- Faced problem in detecting edges of shapes in noisy images. Used cv2.Canny() to detect the clean edges of shapes.
- Were only detecting Square and Rectangle in Quadrilaterals. But then used np.arccos() function to find angle and found length of each edge using distance formula to differentiate between rhombus, parallelogram and trapezoid.

7. Conclusion

- Using OpenCV detected objects from an image using different functions also to find the area, centroid of that object.
- Learn to build a basic Automatic number Plate Recognition System using OpenCV and Pytesseract library python.
- Basic method applied like morphological operation, thresholding, bitwise operation, and contours to localize a number plate in an image.

8. References

1) https://docs.opencv.org/master/
2) https://www.geeksforgeeks.org/opencv-overview/
3) https://www.geeksforgeeks.org/text-detection-and-extraction-using-opencv-and-ocr/
4) https://www.udentify.co/Blog/12/2019/introduction-to-image-processing/
5) https://2020.robotix.in/tutorial/imageprocessing/shape_detection/#:~:text=Shape%20detection%20is%20an%20important,in%20brightness%2Ccolor%20or%20texture.&text=There%20are%20various%20advanced%20means,better%20algorithms%20for%20the%20same
6) https://deepomatic.com/en/what-is-object-recognition-and-how-you-can-use-it
7) https://nanonets.com/blog/ocr-with-tesseract/
8) https://pypi.org/project/pytesseract/