JSS Mahavidyapeetha
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
MYSURU 570 006

Mini project report submitted in partial fulfillment of curriculum prescribed for Software Project Management(CS640) for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering
By

Sl. No.	**USN**	**Name**
01	01JST17CS052	Eeshaan Achar
02	01JST17CS097	Navneet Gajanan Hosmane

Under the guidance of

Dr. Trisiladevi C. Nagavi
Assistant Professor,
Dept. of CS&E,
JSSS&TU MYSURU

JSS Mahavidyapeetha
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
MYSURU 570 006

CERTIFICATE

This is to certify that the work entitled **"Object Recognition"** is a bonafied work carried out **by Eeshaan Achar, Navneet Gajanan Hosmane** in partial fulfillment of the award of the degree of **Bachelor of Engineering in Computer Science and Engineering of JSS Science and Technology, Mysuru during the year 2020**. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Digital Image Processing (CS662) course.

Course In Charge and Guide

Dr. Trisiladevi C. Nagavi
Assistant Professor
Dept. of CS & E
JSS STU Mysore.

Place: Mysuru **Date:**

Abstract

Vision is the most advanced of our senses, so it is not surprising that images play the single most important role in human perception. Interest in digital image processing methods stems from two principal application areas: improvement of pictorial information for human interpretation; and processing of image data for storage, transmission, and representation for autonomous machine perception. However, unlike humans, who are limited to the visual band of the electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate on images generated by sources that humans are not accustomed to associating with images. And, in modern days images are preferred ways of conveying messages since images are larger part of internet communication. Thus, digital image processing encompasses a wide and varied fields of applications. We will discuss object recognition using the basic operations involved in digital image processing.

Acknowledgements

We have put a lot of effort in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. we would like to extend our sincere thanks to all of them.

We are highly indebted to Dr. Trisiladevi C. Nagavi ma'am for their guidance and constant supervision as well as for providing necessary information regarding the project and also for their support in completing the project.

We would like to express our gratitude towards our parents for their kind co-operation and encouragement which help me in completion of this project.

Our thanks and appreciations also go to our team in developing the project and people who have willingly helped us out with their abilities.

Contents

1 Introduction

Today object recognition is one of the popular topics and takes more attention day by day. There is a wide application area for object recognition fields like industry (quality control etc.), medical image possessing, military applications, in the field explorations, autonomous vehicles (cars, robots, etc.). Even though many applications mentioned here involve deep learning, object recognition acts as a foundation for all of these applications.

Although computer vision, image processing, image analysis, robot vision and machine vision are very close fields, the difference of them should be realized. Main differences between them are stated below [5, 6]:

Computer Vision: Computer vision tends to focus on the 3D scene projected onto one or several images, for example reconstruction structure of the 3D scene from one or several images. Computer vision often based on more or less complex assumptions about the scene described in an image.

Machine Vision: Machine vision tends to focus on applications, mainly in industry, e.g., vision based autonomous robots and systems for vision based inspection or measurement. This imply that image sensor technologies and control theory often are integrated with the processing of image data to control a robot and that real-time processing is emphasized by means of efficient implementations in hardware and software.

Image Analysis: Image analysis tends to focus on 2D images, how to transform one image to another, e.g., by pixel-wise operations such as contrast enhancement, local operations such as edge extraction or noise removal, or geometrical transformations such as rotating the image. This characterization implies that image processing/analysis neither require assumptions nor produce interpretations about the image content.

Pattern Recognition: Pattern recognition is a field which uses various methods to extract information from signals in general, mainly based on statistical approaches. A significant part of this field is devoted to applying these methods to image data.

Imaging: This technique basically focus on the process of producing images, but sometimes also deals with processing and analysis of images. For instance, medical imaging contains lots of work on the analysis of image data in medical applications.

1.1 Introduction to problem domain

An object recognition system finds objects in the real world from an image of the world, using object models which are known a priori. This task is surprisingly difficult. Humans perform object recognition effortlessly and instantaneously. Algorithmic description of this task for implementation on machines has been very difficult. With this project, we will try to introduce some techniques that have been used for object recognition in many applications.

The object recognition problem can be defined as a labeling problem based on models of known objects. Formally, given an image containing one or more objects of interest (and background) and a set of labels corresponding to a set of models known to the system, the system should assign correct labels to regions, or a set of regions, in the image. The object recognition problem is closely tied to the segmentation problem: without at least a partial recognition of objects, segmentation cannot be done, and without segmentation, object recognition is not possible. Since, machine learning is required for labeling it is beyond the scope of our project.

1.2 Statement of the problem

To detect the objects present in an image with utmost efficiency using fundamental steps in digital image processing.

1.3 Objectives of the project

Object detection involves detecting instances of objects from a particular class in an image. The goal of object detection is to detect all instances of objects from a known class, such as people, cars or faces in an image. Typically only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored. In our project we are applying basic image processing techniques to detect objects present in the frame. Combining this with deep learning concepts will yield automated systems like autonomous car, weather forecasting, Artificial Intelligence and so on.

The below picture describes fundamental steps involved in digital image processing. It does not mean that every image processing should involve all of these steps. But, these are the basic operations which are helpful in providing efficient ways to obtain information from digital images. So in our project, with some of these operations we will implement object recognition.

1.4 Applications

Object recognition technology has matured to a point at which exciting applications are becoming possible. A number of applications are listed below:

1.4.1 Autonomous Vehicles

In recent years, autonomous/self-driving cars have drawn much interest as a topic of research for both academia and industry. For a car to be a truly autonomous, it must make sense of the environment through which it is driving. The autonomous car must be able to both localize itself in an environment and identify and keep track of objects (moving and stationary). The car gets information about the environment using exteroceptive sensors such LiDAR (Light Detecting And Ranging), cameras, inertial sensors, and GPS. The information from these sensors can be used together and fused to

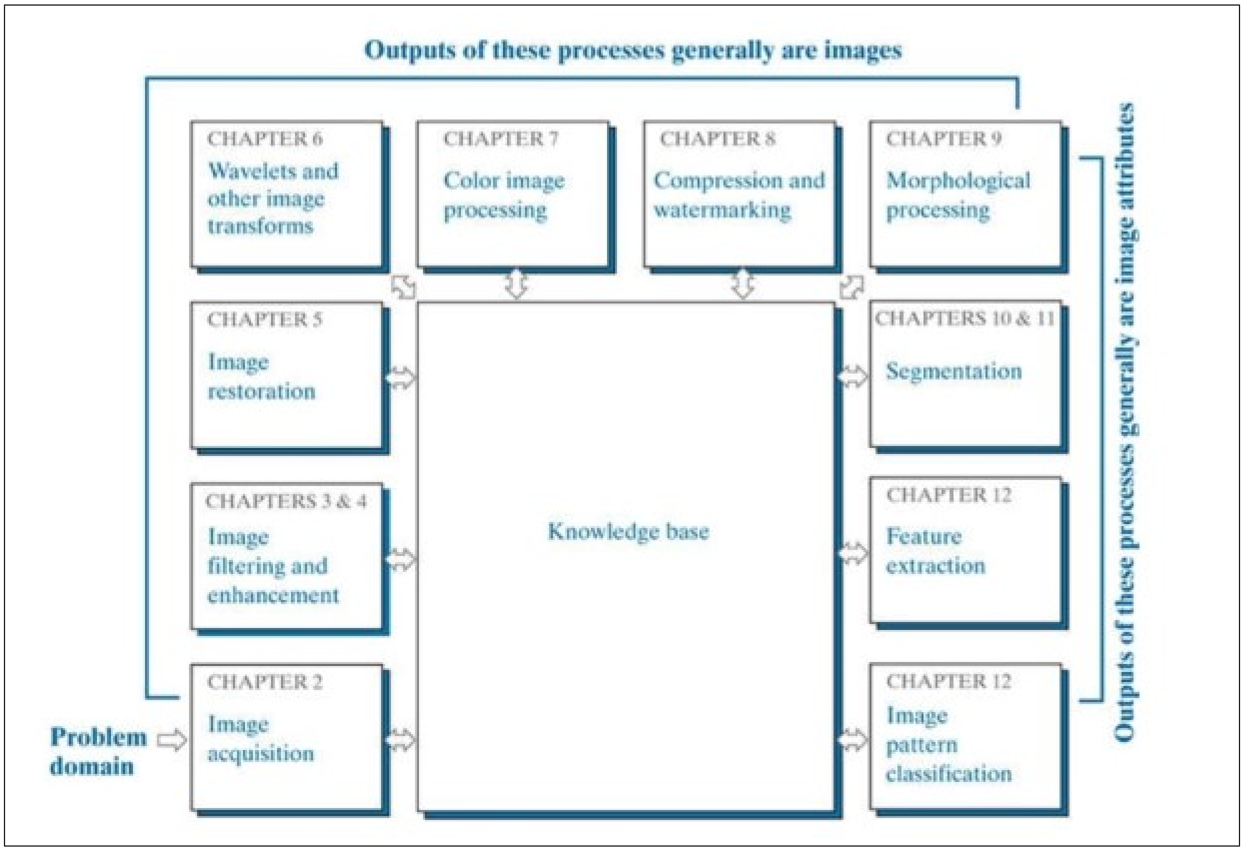


Figure 1: Fundamentals steps in digital image processing.(Source : Figure 1.23 from the book Digital Image Processing (3rd Edition) by Rafael C. Gonzalez,Richard E. Woods)

localize the car and track objects in its environment, allowing it to travel successfully from one point to another.

1.4.2 Forensics

A well known application of object detection is face detection, that is used in almost all the mobile cameras. Another application of this is in crime detection, where the face of criminal is matched with millions of other faces to get any information. Handwriting recognition (HWR), also known as Handwritten Text Recognition (HTR), is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, photographs, touch-screens and other devices. Artificial intelligence can be used to train a machine in recognizing alphabets, numerals and other handwritten text within emails, bank cheques, official documents or images.

1.4.3 Surveillance systems

Safety on the job site continues to be an area of concern for the heavy-duty equipment industry as operator visibility is often restricted due to the large size of the equipment. Because of this, object detection and other safety systems have become more prevalent on heavy equipment within recent years. Other than that, surveillance systems are used in child safety equipment.

1.4.4 Satellite imagery

Satellite imagery has already been used in applications such as, study of Land Use and Land Cover (LULC) change detection to identify illegal deforestation, or to inform agricultural projects;the use of satellite imagery to study refugee settlements; or its analysis to help with rescue operations after natural catastrophes such as floods or fires.

2 Tools and technology used

2.1 Python 3.7.1

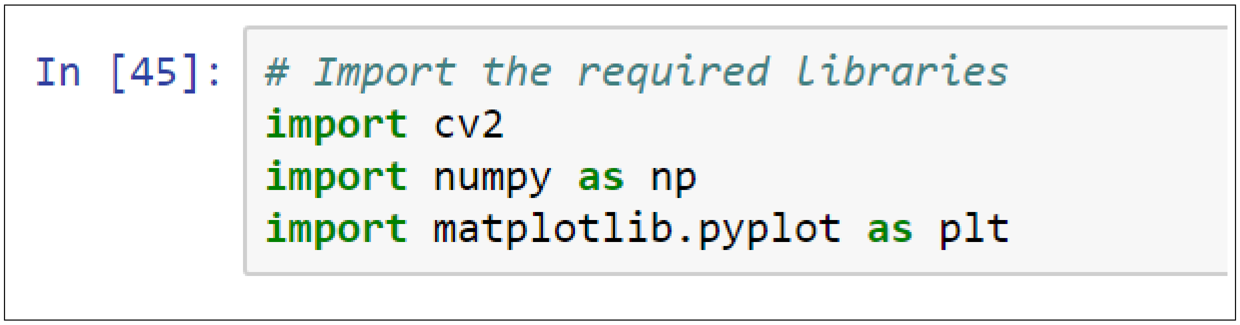
Python is an interpreted, high-level, general-purpose programming language. Python provides all the required functionalities to implement object recognition system. Python is compatible with numerous platforms, so a developer doesn't have to bother about any issues which often occurs with other languages. Many python libraries like numpy, matplotlib and openCV are used in this project.

2.2 Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Jupyter supports over 40 programming languages, including Python, R, Julia, and Scala. Jupyter Notebooks are powerful, versatile, shareable and provide the ability to perform data visualization in the same environment.

3 System design and implementation

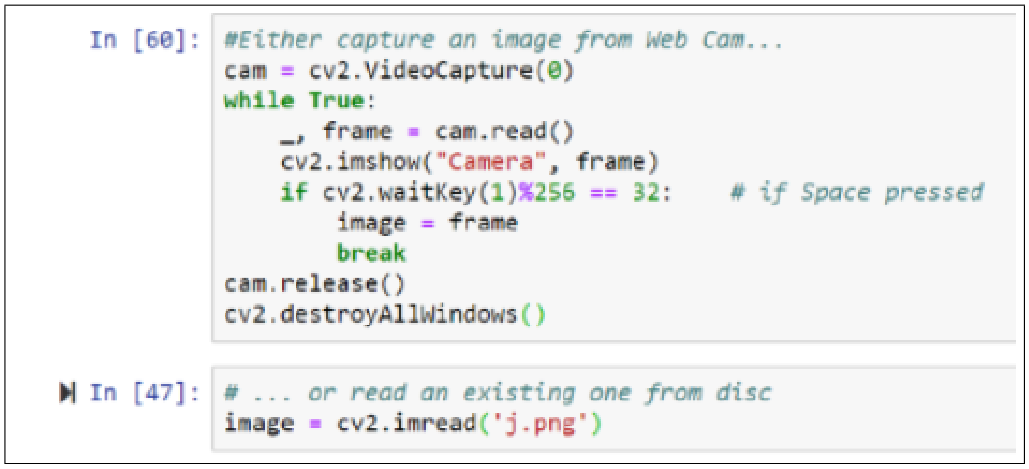
We imported 3 libraries at the beginning of the code : openCV is imported as cv2, numpy as np and matplotlib.pyplot as plt. openCV is a huge open-source library used to process images and videos. Numpy is a library which supports large, multidimensional arrays and matrices, along with large collection of high-level mathematical functions to operate on these arrays. Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.



```
In [45]: # Import the required libraries
         import cv2
         import numpy as np
         import matplotlib.pyplot as plt
```

Figure 2: screenshot taken from libraries imported in our code.

After importing all the libraries, we need an image to work on. So we can either capture an image from web camera or read an existing one from the disk. The required image is stored in a variable named image.

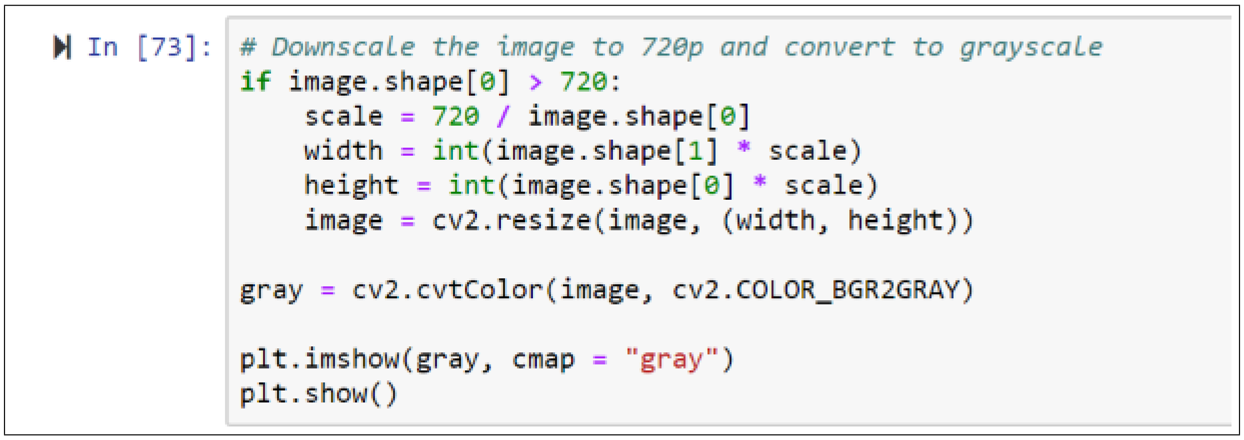


```
In [60]: #Either capture an image from Web Cam...
         cam = cv2.VideoCapture(0)
         while True:
             _, frame = cam.read()
             cv2.imshow("Camera", frame)
             if cv2.waitKey(1)%256 == 32:    # if Space pressed
                 image = frame
                 break
         cam.release()
         cv2.destroyAllWindows()

In [47]: # ... or read an existing one from disc
         image = cv2.imread('j.png')
```

Figure 3: Input image is obtained.

To increase the efficiency of the object recognition system, we downscale the image to 720p in the next step. Along with that, we also convert color image to grayscale. And store the same in gray.



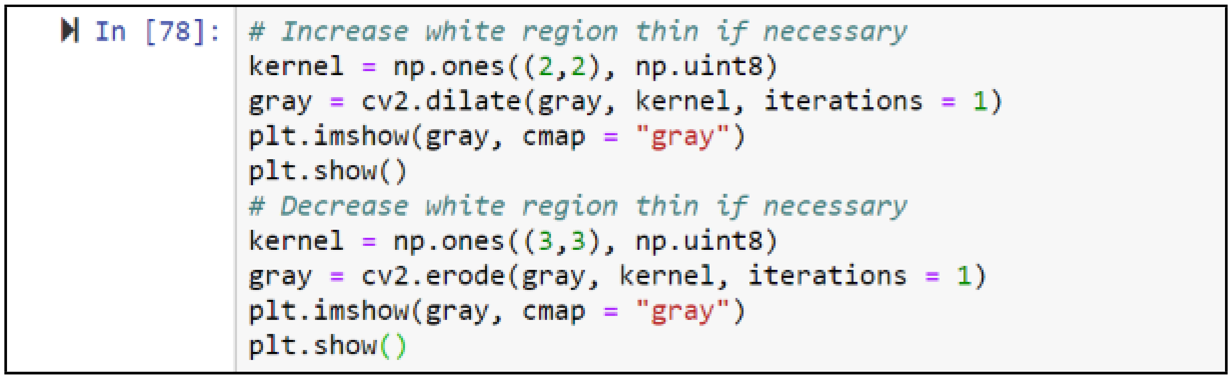
```
In [73]: # Downscale the image to 720p and convert to grayscale
         if image.shape[0] > 720:
             scale = 720 / image.shape[0]
             width = int(image.shape[1] * scale)
             height = int(image.shape[0] * scale)
             image = cv2.resize(image, (width, height))

         gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

         plt.imshow(gray, cmap = "gray")
         plt.show()
```

Figure 4: Resizing the image obtained.

After the above steps, we perform morphological operations, which are a set of operations that process images based on shapes. Morphological operations apply a structuring element to an input image and generate an output image. The most basic morphological operations are two: Erosion and Dilation. In openCV both dilate and erode functions are available.



```
In [78]: # Increase white region thin if necessary
         kernel = np.ones((2,2), np.uint8)
         gray = cv2.dilate(gray, kernel, iterations = 1)
         plt.imshow(gray, cmap = "gray")
         plt.show()
         # Decrease white region thin if necessary
         kernel = np.ones((3,3), np.uint8)
         gray = cv2.erode(gray, kernel, iterations = 1)
         plt.imshow(gray, cmap = "gray")
         plt.show()
```

Figure 5: Code for Dilation and Erosion.

Dilation - This operation consists of convoluting an image "A" with some kernel (B), which can have any shape or size, usually a square or circle. The kernel B has a defined anchor point, usually being the center of the kernel. As the kernel B is scanned over the image, we compute the maximal pixel value overlapped by B and replace the image pixel in the anchor point position with that maximal value. As you can deduce, this maximizing operation causes bright regions within an image to "grow" (therefore the name dilation).

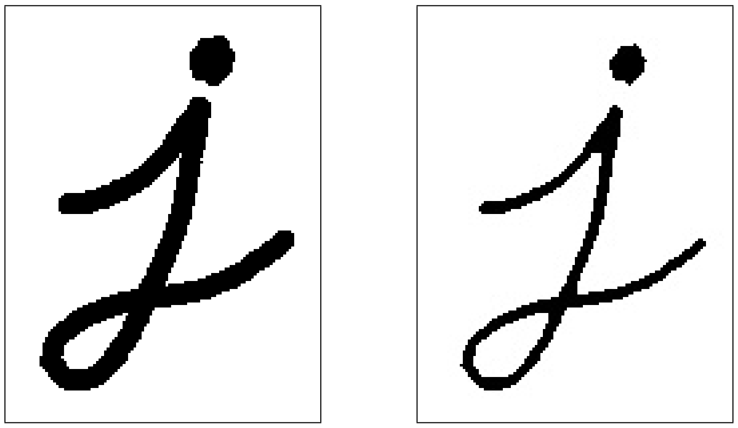

(a) Original Image. (b) Dilated Image.

Figure 6: The same image before and after Dilation.

<u>Erosion</u> - What this does is to compute a local minimum over the area of the kernel. As the kernel B is scanned over the image, we compute the minimal pixel value overlapped by B and replace the image pixel under the anchor point with that minimal value.

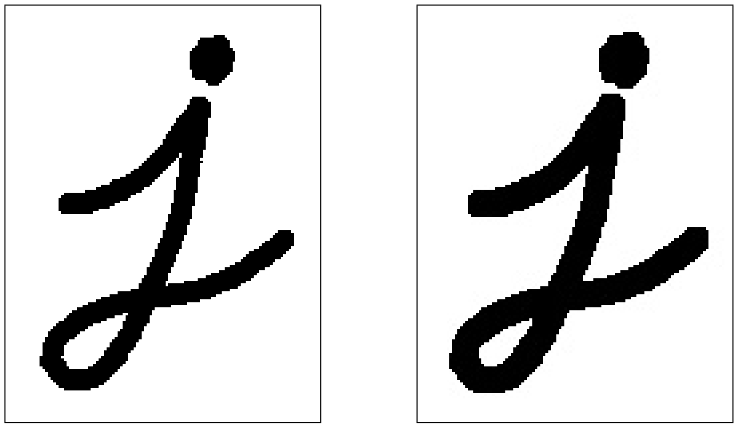
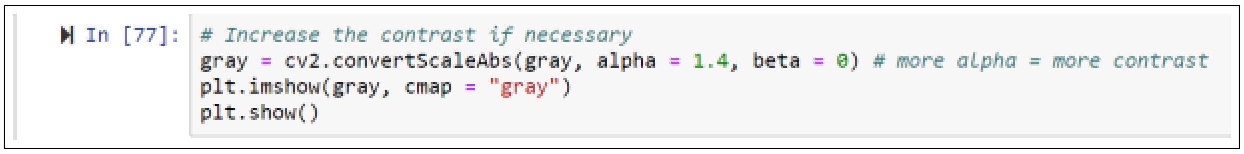

(a) Original Image. (b) Eroded Image.

Figure 7: The same image before and after Erosion.

If morphological processing causes decrease in sharpness of the image, then we increase the contrast of the image. Contrast is the distinction between lighter and darker areas of an image, and it refers to making more obvious the objects or details within an image. Increasing contrast on an image will increase the difference between light and dark areas so light areas will become lighter and dark areas will become darker. So, the details in the image will be more detailed. For this convertScaleAbs method is used by passing alpha value as the parameter. More the alpha value, more contrast is obtained.



```
In [77]: # Increase the contrast if necessary
         gray = cv2.convertScaleAbs(gray, alpha = 1.4, beta = 0) # more alpha = more contrast
         plt.imshow(gray, cmap = "gray")
         plt.show()
```

Figure 8: Code for increasing the contrast.

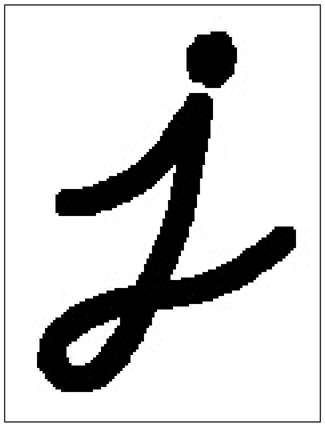
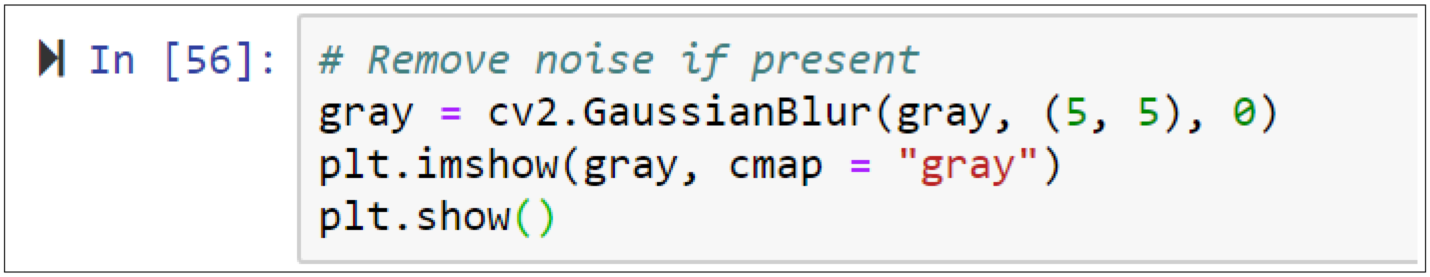


Figure 9: The image after applying contrast.(The image could have undergone either erosion or dilation. In this case it's erosion.)

If there noise is present in the image, it will be removed using GaussianBlur function. This function has a Gaussian filter that removes the high-frequency components from an image.



```
In [56]: # Remove noise if present
         gray = cv2.GaussianBlur(gray, (5, 5), 0)
         plt.imshow(gray, cmap = "gray")
         plt.show()
```

Figure 10: Code for noise removal.

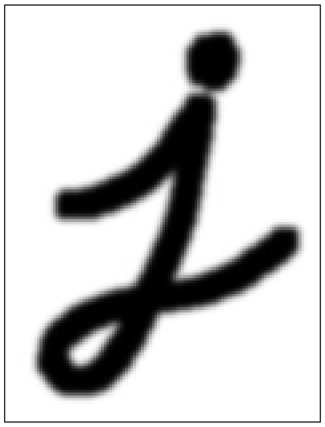
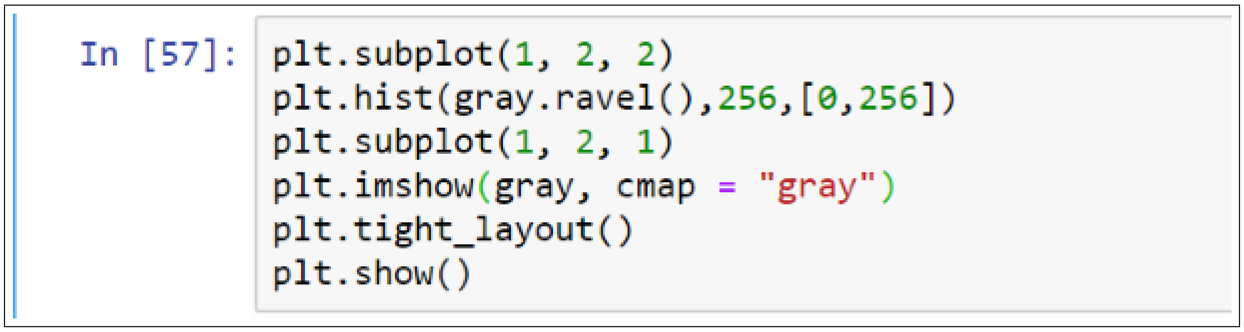


Figure 11: After applying noise removal to fig.9.

We plot pixel value Vs intensity graph to determine the threshold to be set



```
In [57]: plt.subplot(1, 2, 2)
         plt.hist(gray.ravel(),256,[0,256])
         plt.subplot(1, 2, 1)
         plt.imshow(gray, cmap = "gray")
         plt.tight_layout()
         plt.show()
```

Figure 12: Code for Histogram plot on obtained image.

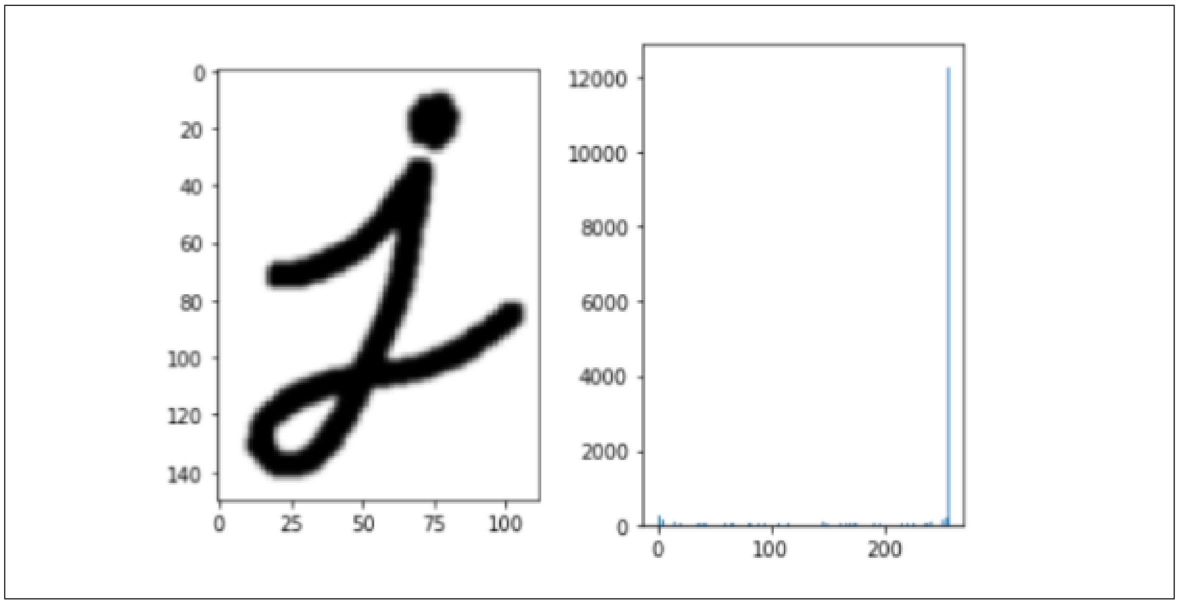
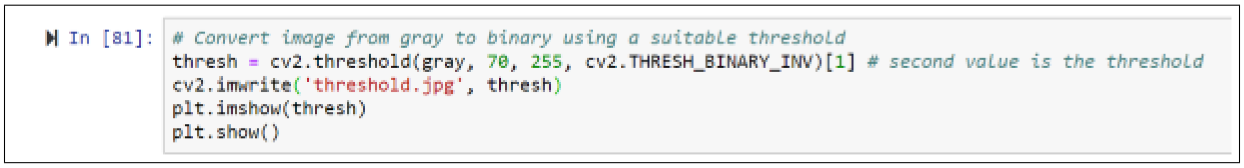


Figure 13: Histogram plot for fig.11.

To convert the image from gray to binary a suitable threshold is used. This operation is performed using the function threshold present in openCV.



```
In [81]: # Convert image from gray to binary using a suitable threshold
         thresh = cv2.threshold(gray, 70, 255, cv2.THRESH_BINARY_INV)[1] # second value is the threshold
         cv2.imwrite('threshold.jpg', thresh)
         plt.imshow(thresh)
         plt.show()
```

Figure 14: Code for threshold.

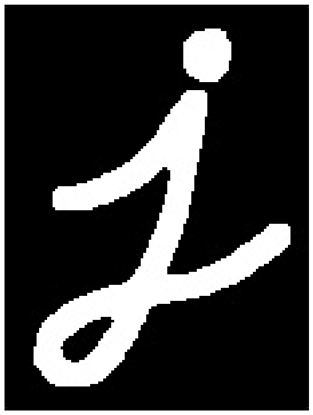
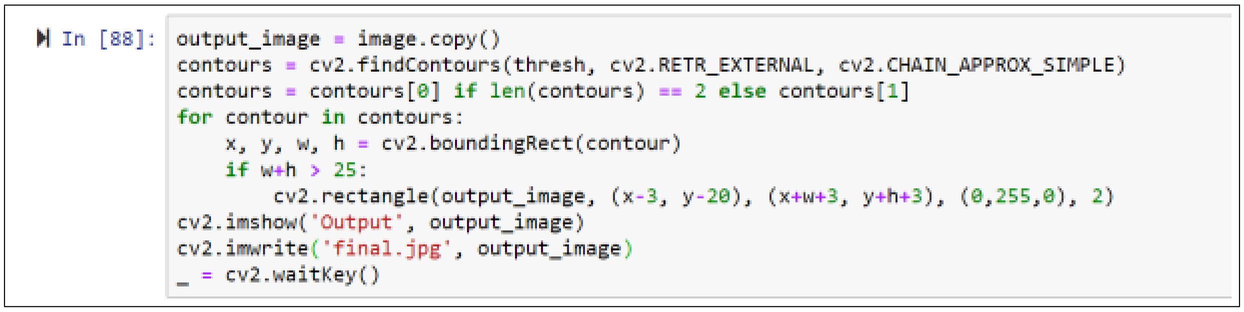


Figure 15: Image obtained after all the above operations.

After getting a binary image we find the contours between different objects and the background. And based on that we draw a rectangular box around the object. By doing the same many number of times, a neural network learns to detect objects.



```
In [88]: output_image = image.copy()
         contours = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
         contours = contours[0] if len(contours) == 2 else contours[1]
         for contour in contours:
             x, y, w, h = cv2.boundingRect(contour)
             if w+h > 25:
                 cv2.rectangle(output_image, (x-3, y-20), (x+w+3, y+h+3), (0,255,0), 2)
         cv2.imshow('Output', output_image)
         cv2.imwrite('final.jpg', output_image)
         _ = cv2.waitKey()
```

Figure 16: Code for threshold.

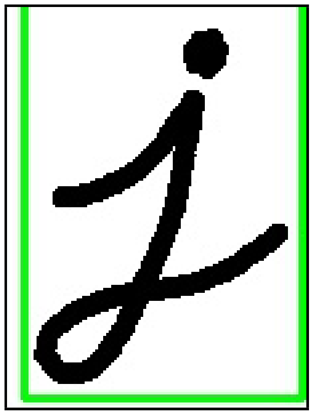


Figure 17: Final object detected in the image.

4 System testing and results analysis

System testing is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.So, in this section we will apply our code to different objects and test how efficiently it works against all these circumstances.

Handwriting Recognition: We tested our code with Handwritten notes and the following are the results we obtained.

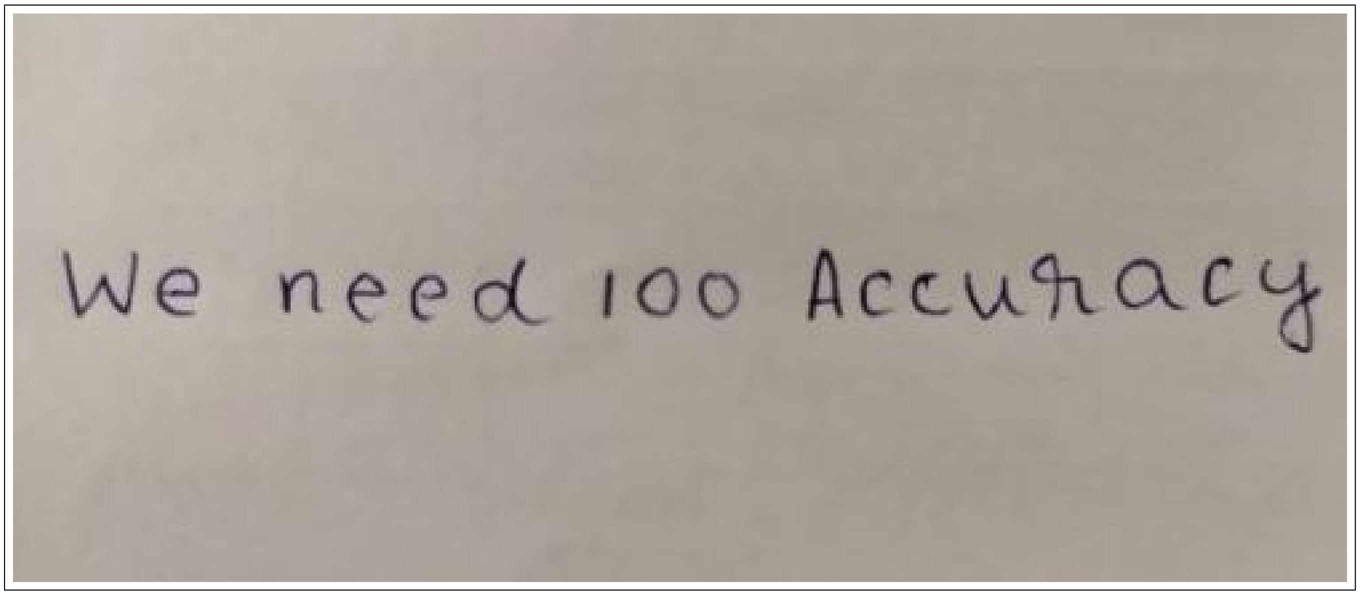


Figure 18: Original Image.

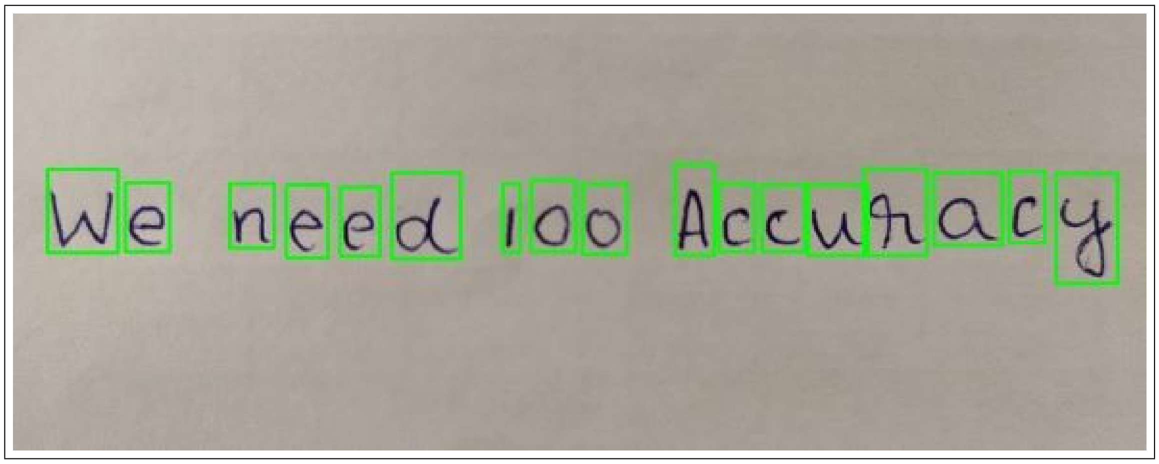


Figure 19: Histogram plot for fig.11.

The problem in recognizing the handwriting with the current version of the code is that, we have to set the threshold value manually to convert from grayscale to binary image. So, for different images different threshold values have to be set. So for the below image if we keep the threshold as previous value, then some of the contours are not detected. We had set threshold value as 120.

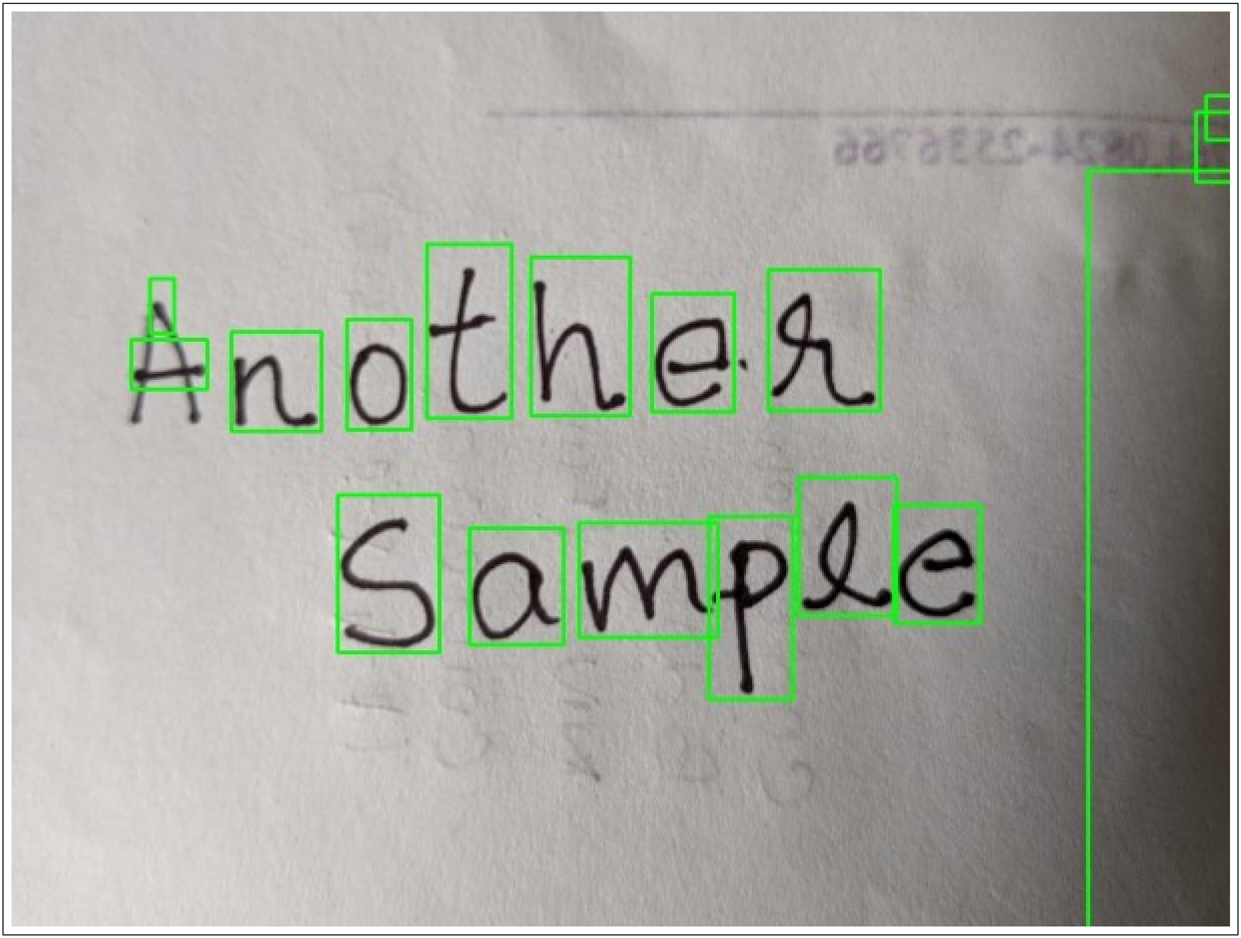


Figure 20: Original Image.

By adjusting the threshold to be 160, we get the following image which is much more accurate.

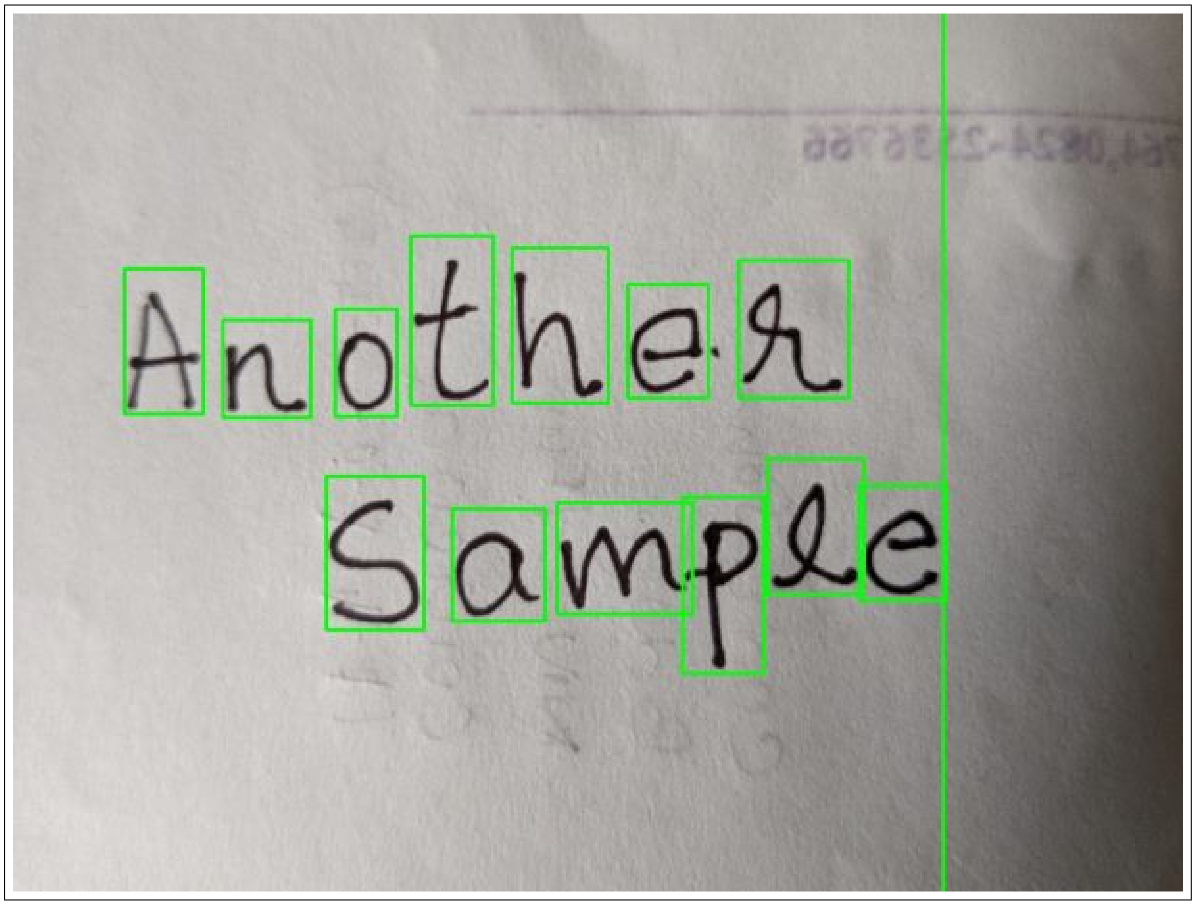


Figure 21: Output Image.

Safety Systems: This section tests our code against child safety systems. If a child is spotted approaching any water bodies or any other dangerous places, object recognition techniques come in handy.

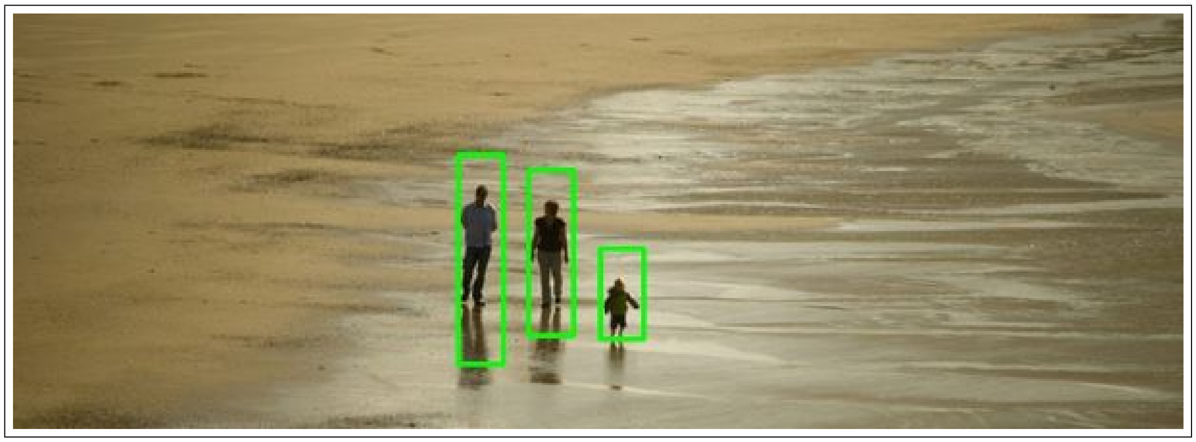


Figure 23: Output Image.

Figure 22: Original Image.

Item detection in grocery stores: In grocery or any stationary stores, we can bill the items automatically with the help of object recognition and detection technique.

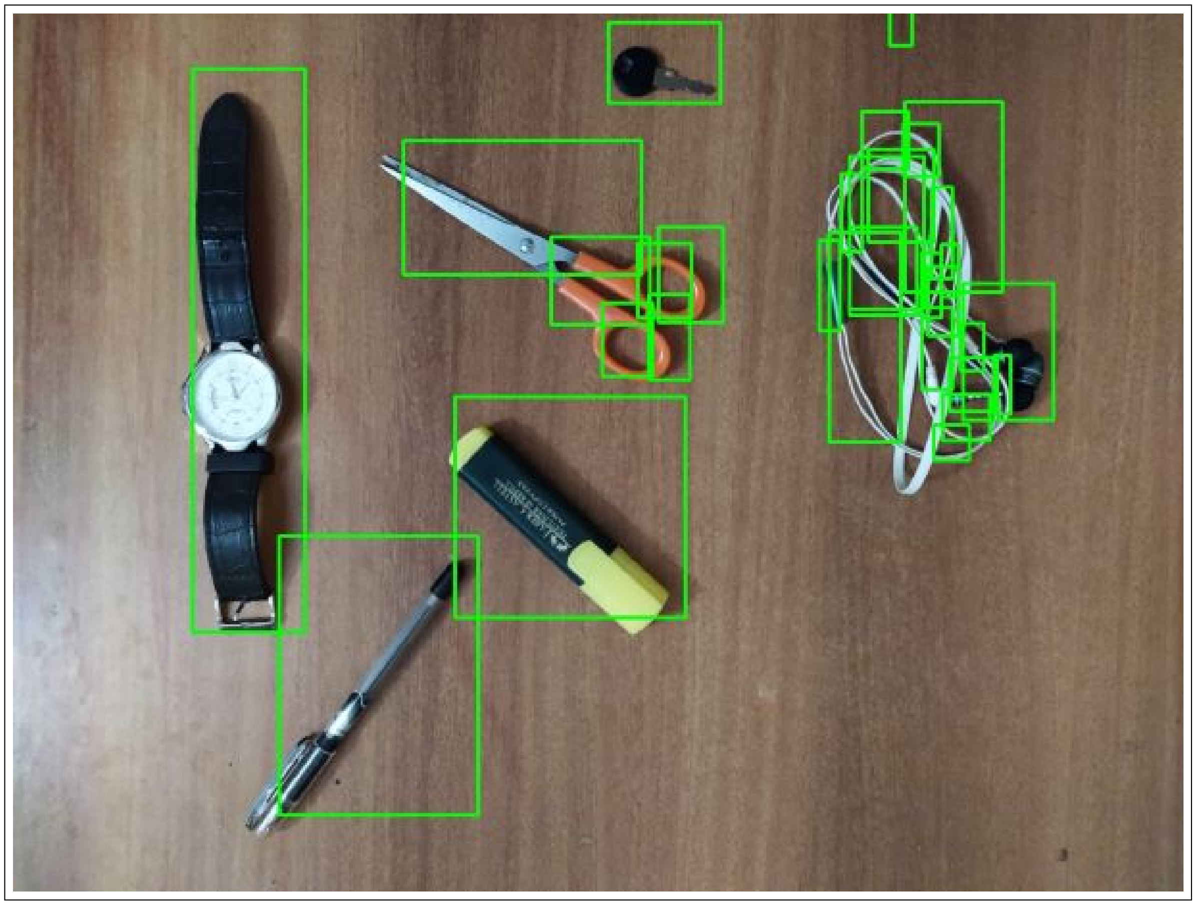


Figure 25: Output Image.

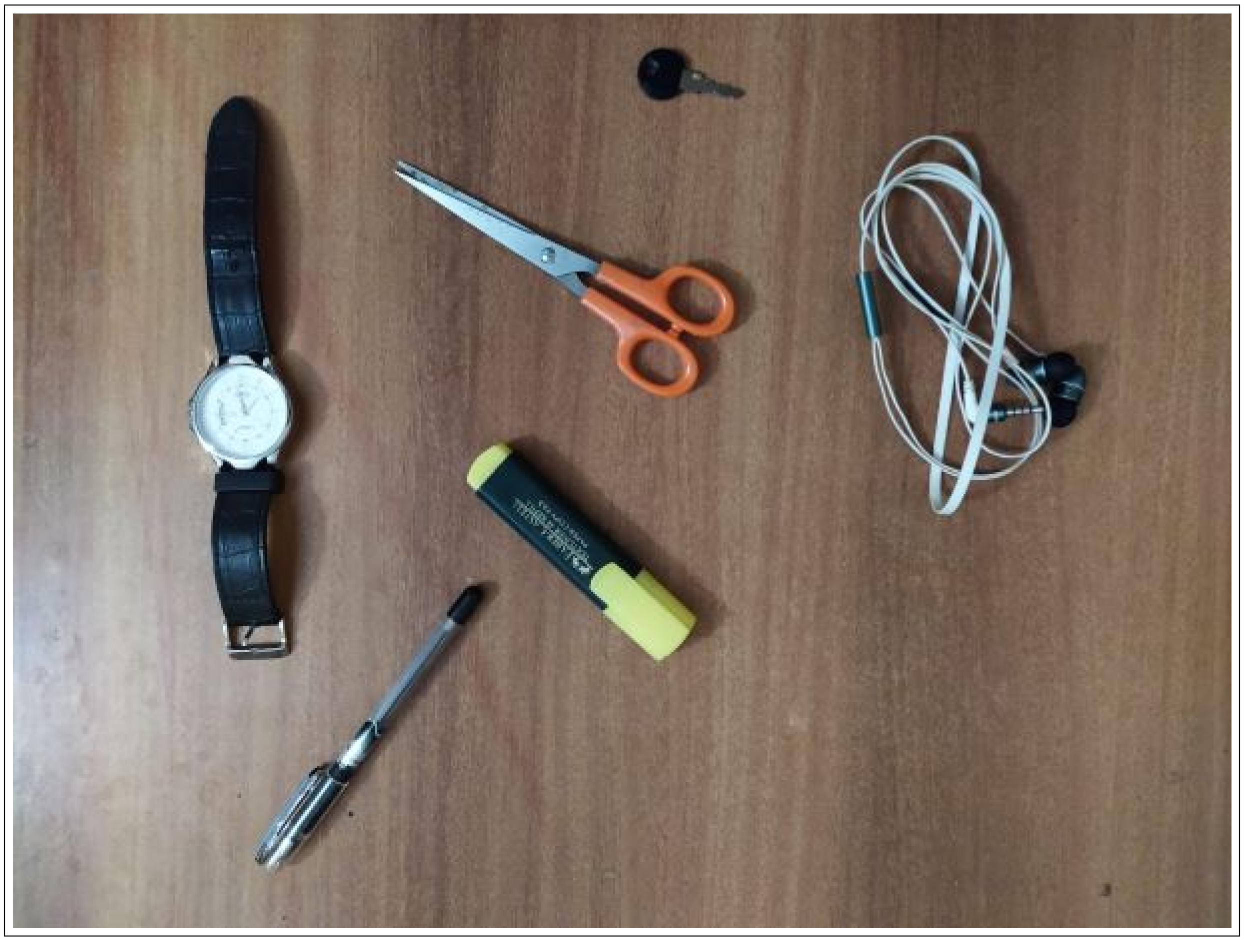


Figure 24: Original Image.

As you can see in the above fig.25 some part of the objects are not clearly segmented. This is due to threshold value which we set. By automating the process of selecting threshold, this problem can be solved. And also if we train the system with a deep network, then the algorithm will be more efficient.

5 Conclusion

Object detection is a key ability required by most computer and robot vision systems. The latest research on this area has been making great progress in many directions. Since the humanity is fast pacing towards an automated world, object detection acts as an ”eye” for this world. Although great progress has been observed in the last years, and some existing techniques are now part of many consumer electronics (e.g., face detection for auto-focus in smartphones) or have been integrated in assistant driving technologies, we are still far from achieving human-level performance, in particular in terms of open-world learning. Finally, we need to consider that we will need object detection systems for nano-robots or for robots that will explore areas that have not been seen by humans, such as depth parts of the sea or other planets, and the detection systems will have to learn to new object classes as they are encountered. In such cases, a real-time open-world learning ability will be critical.

The proposed object recognition system uses grayscale image and discards the color information. The color information in the image can be used for recognition of the object. Color based object recognition plays vital role in Robotics.

References

[1] Digital Image Processing (4th Edition) by Rafael C. Gonzalez,Richard E. Woods

[2] OpenCV - Image Processing,
`https://opencv.org/`

[3] Overleaf - LaTeX documentation
`https://www.overleaf.com/learn`