***Project Phase IV Report***

***On­­­***

**Image Processing in Python**

**Submitted for the requirement of**

**Project course**

BACHELOR OF ENGINEERING

**COMPUTER SCIENCE & ENGINEERING**

****

**Submitted to: Rohini Bawa(e12228) Submitted By: Sahul Kr. Parida(20BCS4919)**

**Project Teacher(Supervisor) Aditya Raj(20BCS4955)**

**Kumar Shubham(20BCS4965)**

**Shaswat Rai(20BCS4905)**



**Co Supervisor Signature**

**Rakesh Sahu (E12311)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**CHANDIGARH UNIVERSITY, GHARUAN**

**June 2022**

### 

**ABSTRACT**

Vision is a major source of information for human beings. ‘Earlier it was impossible to achieve but due to the development of new technologies it has been made possible’. Image processing has its impact on communication devices also. By digital image processing we can enhance the image, extract the text from image, edges of images can be detected and we can apply other effects also. We can get any details about the images. There are many applications of digital image processing. Almost this technique is use in every field, medical field, robotics., neural networking, also useful in Crime branch for investigation.

The project mainly deals with one of the application of digital image processing that is detecting edges in the images.

Image Processing includes changing the nature of an image in order to improve its pictorial information for human interpretation, for autonomous machine perception. Digital image processing is a subset of the electronic domain wherein the image is converted to an array of small integers, called pixels, representing a physical quantity such as scene radiance, stored in a digital memory, and processed by computer or other digital hardware. Interest in digital image processing methods stems from two principal applications areas: improvement of pictorial information for human interpretation; and processing of image data for storage, transmission, and representation for autonomous machine perception. Edges characterize boundaries and edge detection is one of the most difficult tasks in image processing hence it is a problem of fundamental importance in image processing. In this paper investigates different steps of digital image processing like, a high-speed non-linear Adaptive median filter implementation is presented. Then Adaptive Median Filter solves the dual purpose of removing the impulse noise from the image and reducing distortion in the image.

The Image Processing Toolbox software is a collection of functions that extend the capability of the MATLAB numeric computing environment. The toolbox supports a wide range of image processing operations on the given image.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
| Sr no. | Topic | Page No. |
| 1 | Introduction | 4 |
| 2 | Literature Review | 4 |
| 3 | Problem Definition | 5 |
| 4 | Objectives | 5 |
| 5 | Feature/characteristics identification | 6-7 |
| 6 | Use of Modern tools in design and analysis | 8 |
| 7 | Constraints Identification | 9-10 |
| 8 | Analysis of features and finalization subject to constraints | 11-13 |
| 9 | Design selection | 14-18 |
| 10 | Source code and Output | 19-23 |
| 11 | Image Segmentation | 24-34 |
| 12 | Project Future Scope | 35 |
| 13 | Conclusion | 36 |
| 14 | References | 36 |

**1.Introduction**

**Image processing Python Project** is one form of signal processing for which the input is an image, such as photographs or frames of video; the output of image processing can be either an image or a set of characteristics or parameters related to the image. Image processing is a technique which involves different operation that can be performed on the image. Image processing plays a vital role in photography, computer science, and many other fields.

Image processing has the lot of application in different fields like medical, computer science etc. In this project, we have implemented some of the image processing operations which will help the user to understand the concept of image processing. Therefore, the scope of this project is increasing.

The purpose of this project is to allow users to explore how digitized images can be processed using computer software.

**2.Literature Review**

Many techniques of digital image processing were developed in ,1960s at the Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Laboratories, University of Maryland. The cost of processing of image was very high. But that changed in 1970s, when digital image processing grow rapidly as cheaper computer and hardware became available. Images then could be processed in real time. With the development of fast computers available in 200s, digital image processing has become one of the most common form of image processing. It is used because it is not only the most versatile method, but also the cheapest.

**3.Problem Definition**

Digital image processing is concerned with processing of an image. Image processing is a method to perform operations on images like enhancing images, extracting text from image, detecting edge of image and many other operations. In digital image processing we take an image and convert that image in different forms. Like if we take color image we can convert it into grey image. In this both the input and output is an image. Usually Image Processing System includes treating images as two dimensional signals while applying already set signal processing methods to them.

Today, it is rapidly growing technology. It forms core research area within engineering and computer science disciplines too. Image processing has its wide applications in robotics, machine learning, neural networking, signal processing, medical field, graphics and animations and in many other fields.

**4.Objectives**

The objective of our project is mainly concerned with basic operations on the image. So that user can get required a visual appearance. Image processing is the use of computer algorithm to perform, improve or change some quality of the image. Now a day, it is important in different fields.

The Python Image Processing Library adds image processing capabilities to the Python Interpreter. This library provides extensive file format, an efficient internal representation, and fairly powerful image processing tools. Actually, the Image Library designed for fast access.

The Python Image Library is ideal for image archival and batch processing application. Here our project is providing an image processing tool through which user can give desired effects to an image. It accepts input via mouse and keyboard and gives output on the screen.

So, just one click by mouse, you will get the image with the expected result. In this project, we are providing different image processing tool with GUI so it makes easy for the user to give desired effects to the image.

**5. Feature/characteristics identification**

**5.1 Objectives**

Main objective of Image processing project is to extract important data from images. Using this extracted information description, interpretation and understanding of the scene can be provided by the machine. Main point of image processing is to modify images in to desired manner. This system allows users to take hard copy of the image using printer routines and provides option for users to store file in to disk in different formats. In other words, image processing is called as altering and analysing pictorial information of images. In our daily life we come across different type of image processing best example of image processing in our daily life is our brain sensing lot of images when we see images with eyes and processing is done is very less time.

In existing system there are many techniques which are available for extracting information from images but there is no exact processing is defined. In proposed system we will come across different new techniques in image processing.

* 1. **Single entity**

The entire Python source code for this project will be prepared by Sahul Kumar Parida, who will also be acting as the Project Leader as well as Technical Lead. He will also be responsible for contributing to the overall objectives of the project and the specific team deliverables.

The different Python libraries, modules required for the project like OpenCV, Scikit-Image, SciPy, Pillow or PIL (Python Imaging Library) and NumPy as well as collecting content and resources for the project will be done by Aditya Raj, who will also be managing the deliverables of the project.

The Presentation and Synopsis for the project will be prepared by Kumar Shubham, who will also look into the Hardware and Software Configuration and all the necessary requirements required to implement our project successfully.

And last, but not the least the Project Report will be prepared by Shaswat Rai, who will also be acting as the Project Manager. He will be responsible to plan and schedule tasks and provide a framework for our project activities. His primary purpose will be helping the group to prepare, execute, and control all aspects of a project with optimal time management.

**5.3 Life Span**

Python becomes an apt choice for such Image processing tasks. This is due to its growing popularity as a scientific programming language and the free availability of many State of Art Image Processing tools in its ecosystem. So the life span could last for next 10 years or so.

**5.4 Require Funds**

We don’t require any funds for this project because it’s completely online. We get all the modules and libraries for this project online in web. So we don’t require any funds.

**5.5 Life Cycle**

System development life cycle is a sequence of events carried out by analyst, designers and users to develop and important an information system.

Activities are carried out in different stages.

The phases are as follows:

1.Preliminary Investigation (Feasibility Study)

2.Determination of System Requirement

3.Design of the System

4.Design of The System

5.Development of Software

6.System Testing

7.System Implementation

8.System Maintenance

**5.6 Project Planning and Task Definition**

The entire Python source code for this project will be prepared by Sahul Kumar Parida, who will also be acting as the Project Leader as well as Technical Lead. He will also be responsible for contributing to the overall objectives of the project and the specific team deliverables.

The different Python libraries, modules required for the project like OpenCV, Scikit-Image, SciPy, Pillow or PIL (Python Imaging Library) and NumPy as well as collecting content and resources for the project will be done by Aditya Raj, who will also be managing the deliverables of the project.

The Presentation and Synopsis for the project will be prepared by Kumar Shubham, who will also look into the Hardware and Software Configuration and all the necessary requirements required to implement our project successfully.

And last, but not the least the Project Report will be prepared by Shaswat Rai, who will also be acting as the Project Manager. He will be responsible to plan and schedule tasks and provide a framework for our project activities. His primary purpose will be helping the group to prepare, execute, and control all aspects of a project with optimal time management.

**6. Use of Modern tools in design and analysis**

List of software used:

* Python 3.7.0 IDLE installed in Windows/Linux.
* Virtual Studio Code for fixing the bugs in Python 3.7.0 IDLE and running the python code more efficiently.
* Discord for screen sharing and meetings for discussing about the project.
* Microsoft PowerPoint for doing the PPTs so as to conduct our presentation.
* Github- Here we have uploaded our code, presentation and working of the Image Processing.

**Why am I using Virtual Studio Code instead of Python IDLE?**

I am using Virtual Studio Code 64-Bit 2020 version in my system. VS Code is lightweight and should easily run on 32-bit or 64-bit system.

System requirements for running Virtual Studio Code are:

1. 1.6 GHz or faster processor
2. 1 GB of RAM
3. Microsoft .NET Framework 4.5.2 is required for VS Code. If the user is using Windows 7, we have to make sure .NET Framework 4.5.2 is installed.

According to me, the main reason I am using Virtual Studio Code because it has everything which any programmer expects from any code editor with some additional and useful features. It’s lightweight, fast, open source and cross-platform nature along with other features gives it an extra edge over any other editor.

1. It comes with an in-built debugger which is also one of its key features. It helps in accelerating any programmer’s edit, compile and debug loop.
2. Visual Studio Code also provides features for code management like Go to Definition, Peek definition, Find all references and rename Symbol.

**7. Constraints Identification**

This section defines the project plan for ‘EDGE DETECTION OF IMAGE’. The functionality of the system, scope of the project and other information about specifications is included in this plan.

**7.1 Purpose**

The purpose of this project is to take the input image and detect the edges in that image.

**7.2 Intended Audience**

It is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations

**7.3 Intended Use**

This software can be used to develop any application using different algorithms, data analysis, medical field, scientific and engineering graphics, solving mathematical problems using functions

**7.4 Scope**

The scope of software is vast. It can be used for designing video, electric circuits. Can be use in medical industry, automobile industry. Due to the fast development in the computer technology , future of the image- processing is going to be more flexible. Digital image processing has wide range of applications. In future it has been expected to be less expensive. Due to advancement of this technology there will be millions of robots in world. Advances in the image- processing and artificial intelligence will involve spoken commands, translation of languages, recognizing the finger prints, tracking of people and things, diagnosing medical conditions, performing surgery and automatically driving all forms of transport. With an increasing power of modern computing, the concept of computation can go beyond the present limits and in future, image processing technology will advance. The future trend in remote sensing will towards improved sensors that record the same scene in many spectral channels. Graphics data is becoming important nowadays in image processing app1ications. In future image processing technique will play import role in space also.

**7.5 Definition and acronyms**

MATLAB – Matrix Laboratory

**7.6 Benefits**

1. Processing of images are faster. It requires less time to process the image. There is no need of films and other photographic equipment.
2. Interactive method for detecting face, recognizing fingerprints, detecting cancer.
3. It is ecofriendly process since it does not require chemicals while processing images.
4. We can change the quality of image. We can compress, enhance, quality of image produced are good.
5. Image can be made in any required format.
6. Nowadays each and every book is available on the digital stage. The demand and needs of people are changing so having optimized digital book is need of today’s generation so the digital image processing plays vital role in publishing world.
7. Errors in images can easily be rectified.
8. It analyse blood cells and their composition in our body.
9. Through this technique robot can detect their visions.
10. It’s also helps in pattern recognition.

**7.7 Risks**

1. It is more time consuming.
2. It is cost effective.
3. More complex program is required for implement digital image processing.

**8. Analysis of features and finalization subject to constraints**

For implementing digital image processing we can write code in any language like python, java and any other language. We can also use the software ‘MATLAB’ which offers lots of feature to programmers for exploring their imagination and can make any projects related to digital image processing by using their coding skills.

Some features of MATLAB are:

1. It has several in built commands and math functions that help in performing mathematical calculations.
2. It also provides an interactive environment for design and problem solving.
3. MATLAB features a family of add on application specific solutions called toolboxes. Toolboxes allow us to learn and apply special technology.
4. We can add toolboxes for signal -processing, control -systems, neural- networks, and many other areas.
5. It has facilities for displaying vectors as graphs, and printing these graphs.
6. It includes high level function for two dimensional and three dimensional data visualization, image processing, and presentation graphics.
7. We can create GUI with the help of GUI guide to make our project more effective.
8. GUI guide contains many inbuilt functions through which we can make many unintrols like pushbutton, toggle buttons, lists, checkboxes, axes.
9. It provides many functions and algorithms with external applications and languages such as C, Java, .NET.

**SYSTEM REQUIREMENTS**

The software requires the following to be in the system.

Toolboxes

Minimum 5GB space required

## ANALYSIS AND FUNDAMENTALS OF IMAGE PROCESSING STEPS

### Image Acquisition

Image acquisition is the first step in image processing. This step is also known as preprocessing in image processing. It involves retrieving the image from a source, usually a hardware-based source.

### Image Enhancement

Image enhancement is the process of bringing out and highlighting certain features of interest in an image that has been obscured. This can involve changing the brightness, contrast, etc.

### Image Restoration

Image restoration is the process of improving the appearance of an image. However, unlike image enhancement, image restoration is done using certain mathematical or probabilistic models.

### Color Image Processing

Color image processing includes a number of color modeling techniques in a digital domain. This step has gained prominence due to the significant use of digital images over the internet.

### Wavelets and Multiresolution Processing

Wavelets are used to represent images in various degrees of resolution. The images are subdivided into wavelets or smaller regions for data compression and for pyramidal representation.

### Compression

Compression is a process used to reduce the storage required to save an image or the bandwidth required to transmit it. This is done particularly when the image is for use on the Internet.

### Morphological Processing

Morphological processing is a set of processing operations for morphing images based on their shapes.

### Segmentation

Segmentation is one of the most difficult steps of image processing. It involves partitioning an image into its constituent parts or objects.

### Representation and Description

After an image is segmented into regions in the segmentation process, each region is represented and described in a form suitable for further computer processing. Representation deals with the image’s characteristics and regional properties. Description deals with extracting quantitative information that helps differentiate one class of objects from the other.

### Recognition

Recognition assigns a label to an object based on its description.

**9. Design selection**

User interface

This prototyping of this GUI is done with the help of GUI Guide.

* After creating the above GUI, the image is taken as input
* Convert the coloured input image in gray scale image.
* From the list choose the suitable edge detector of your choice and click on apply button.
* Final image which detect the edges in the input image.

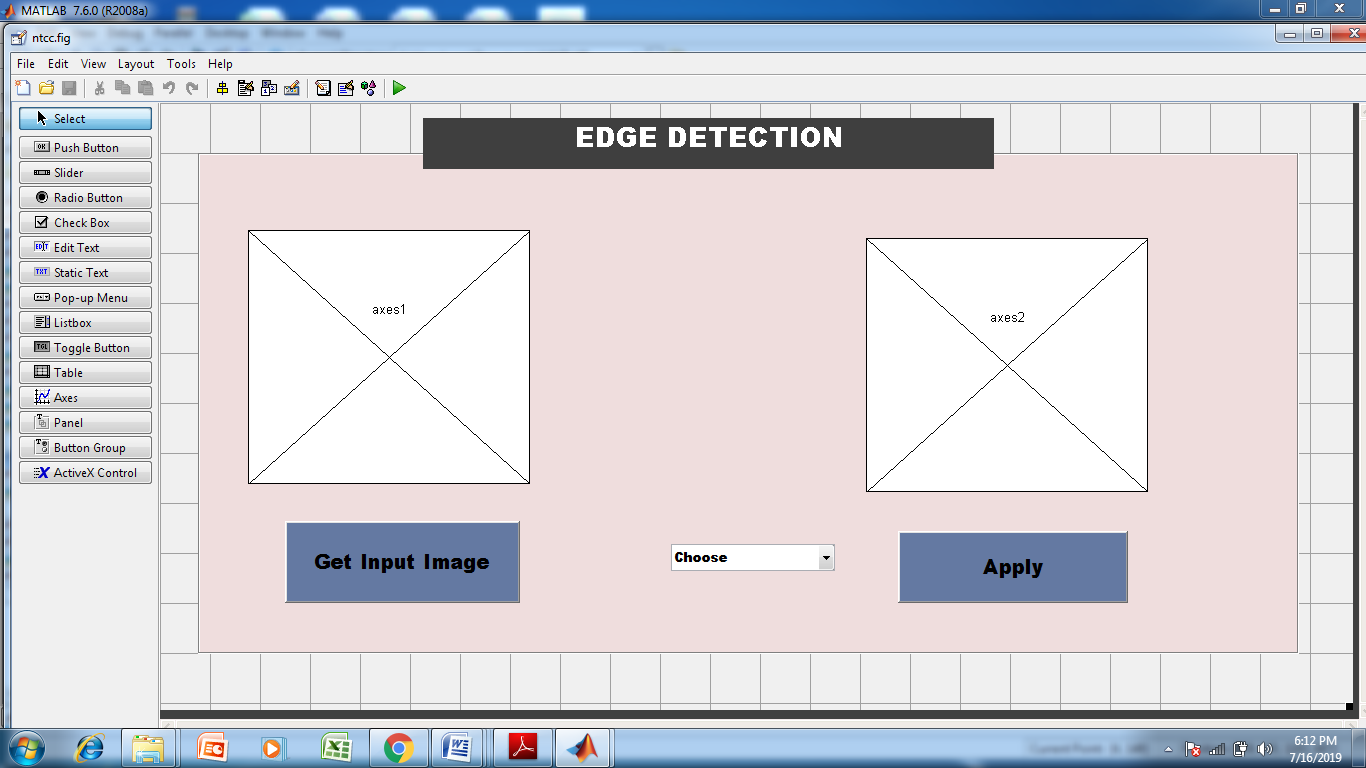


Fig 1 designing of GUI

Image processing is the process of transforming an image into a digital form and performing certain operations to get some useful information from it. The image processing system usually treats all images as 2D signals when applying certain predetermined signal processing methods.

There are five main types of image processing:

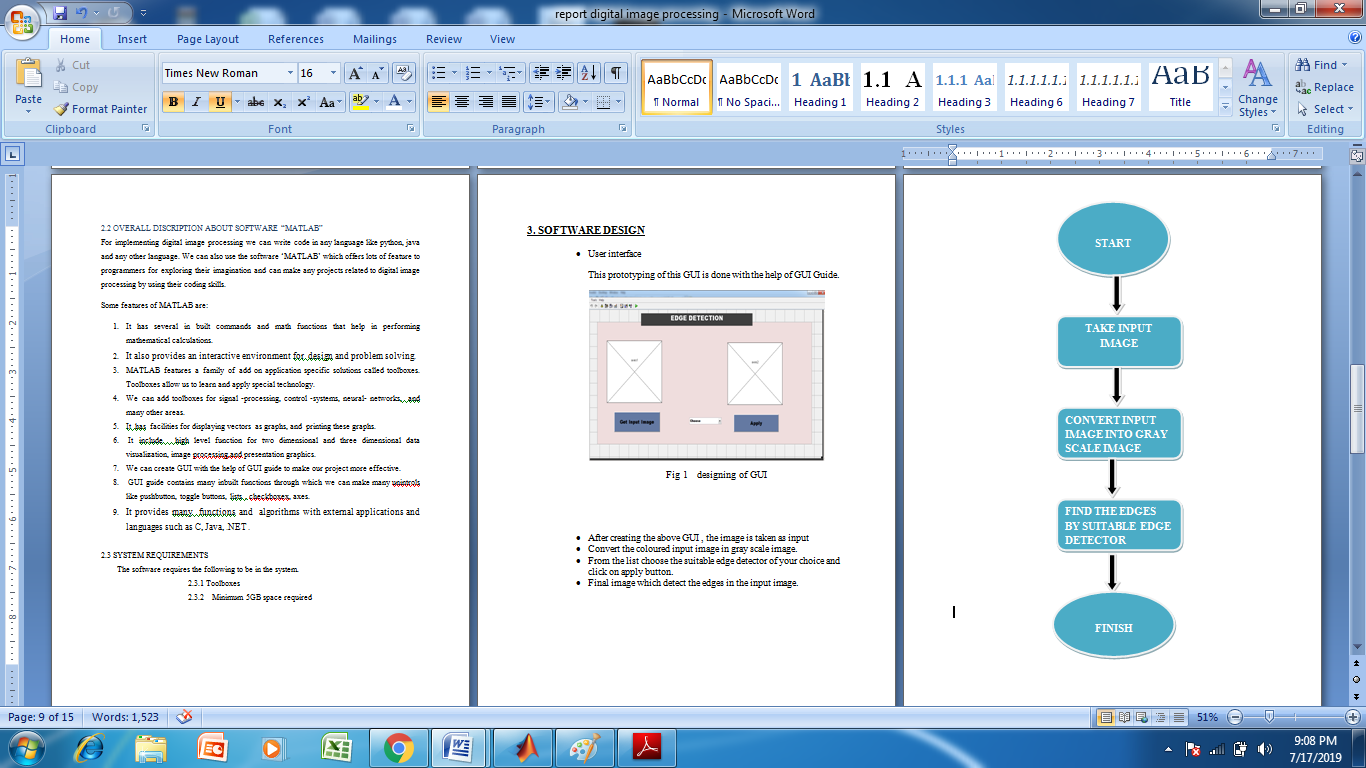
* Visualization - Find objects that are not visible in the image
* Recognition - Distinguish or detect objects in the image
* Sharpening and restoration - Create an enhanced image from the original image
* Pattern recognition - Measure the various patterns around the objects in the image
* Retrieval - Browse and search images from a large database of digital images that are similar to the original image

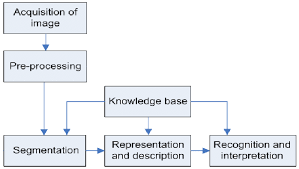
Before we jump into image processing, we need to first understand what exactly constitutes an image. An image is represented by its dimensions (height and width) based on the number of pixels. For example, if the dimensions of an image are 500 x 400 (width x height), the total number of pixels in the image is 200000.

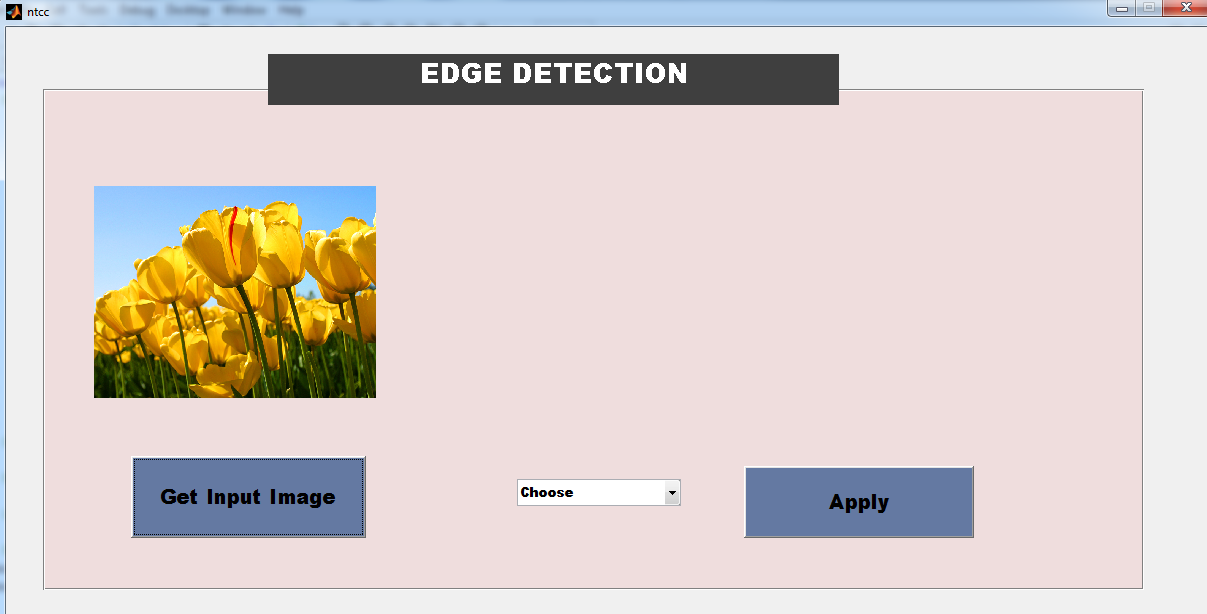
This pixel is a point on the image that takes on a specific shade, opacity or colour. It is usually represented in one of the following:

* Grayscale - A pixel is an integer with a value between 0 to 255 (0 is completely black and 255 is completely white).
* RGB - A pixel is made up of 3 integers between 0 to 255 (the integers represent the intensity of red, green, and blue).
* RGBA - It is an extension of RGB with an added alpha field, which represents the opacity of the image.

Image processing requires fixed sequences of operations that are performed at each pixel of an image. The image processor performs the first sequence of operations on the image, pixel by pixel. Once this is fully done, it will begin to perform the second operation, and so on. The output value of these operations can be computed at any pixel of the image.





fig 2 input image

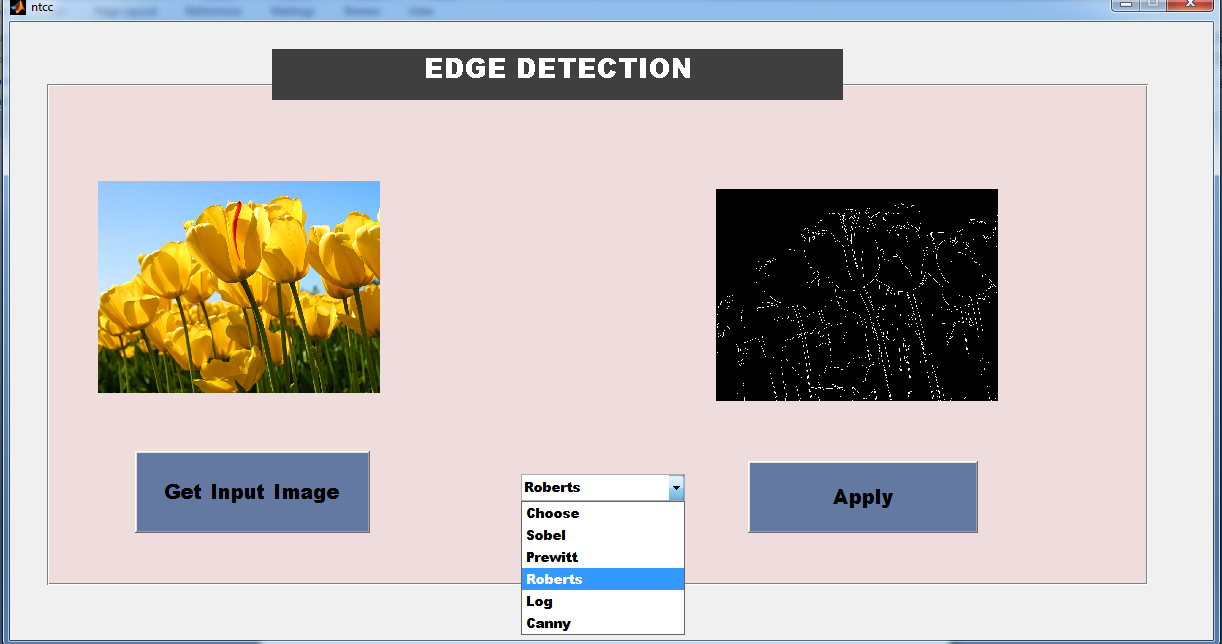


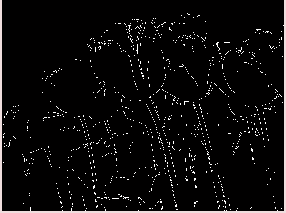
Fig 3 Apply any operator

Input Image Gray Image

Sobel Prewitt

Log Canny

**10. Source code and Output**

mapper.py

import numpy as np

def mapp(h):

    h = h.reshape((4,2))

    hnew = np.zeros((4,2),dtype = np.float32)

    add = h.sum(1)

    hnew[0] = h[np.argmin(add)]

    hnew[2] = h[np.argmax(add)]

    diff = np.diff(h,axis = 1)

    hnew[1] = h[np.argmin(diff)]

    hnew[3] = h[np.argmax(diff)]

    return hnew

Scanner.py

import cv2

import numpy as np

import mapper

image=cv2.imread("test\_img.jpg")   #read in the image

image=cv2.resize(image,(1300,800)) #resizing because opencv does not work well with bigger images

orig=image.copy()

gray=cv2.cvtColor(image,cv2.COLOR\_BGR2GRAY)  #RGB to Gray Scale

cv2.imshow("Gray Scale",gray)

blurred=cv2.GaussianBlur(gray,(5,5),0)  #(5,5) is the kernel size and 0 is sigma that determines the amount of blur

cv2.imshow("Blur",blurred)

edged=cv2.Canny(blurred,30,50)  #30 MinThreshold and 50 is the MaxThreshold

cv2.imshow("Canny",edged)

contours,hierarchy=cv2.findContours(edged,cv2.RETR\_LIST,cv2.CHAIN\_APPROX\_SIMPLE)  #retrieve the contours as a list, with simple apprximation model

contours=sorted(contours,key=cv2.contourArea,reverse=True)

#the loop extracts the boundary contours of the page

for c in contours:

    p=cv2.arcLength(c,True)

    approx=cv2.approxPolyDP(c,0.02\*p,True)

    if len(approx)==4:

        target=approx

        break

approx=mapper.mapp(target) #find endpoints of the sheet. Passing the target image

to mapper.py

pts=np.float32([[0,0],[800,0],[800,800],[0,800]])  #map to 800\*800 target window

op=cv2.getPerspectiveTransform(approx,pts)  #get the top or bird eye view effect

dst=cv2.warpPerspective(orig,op,(800,800))

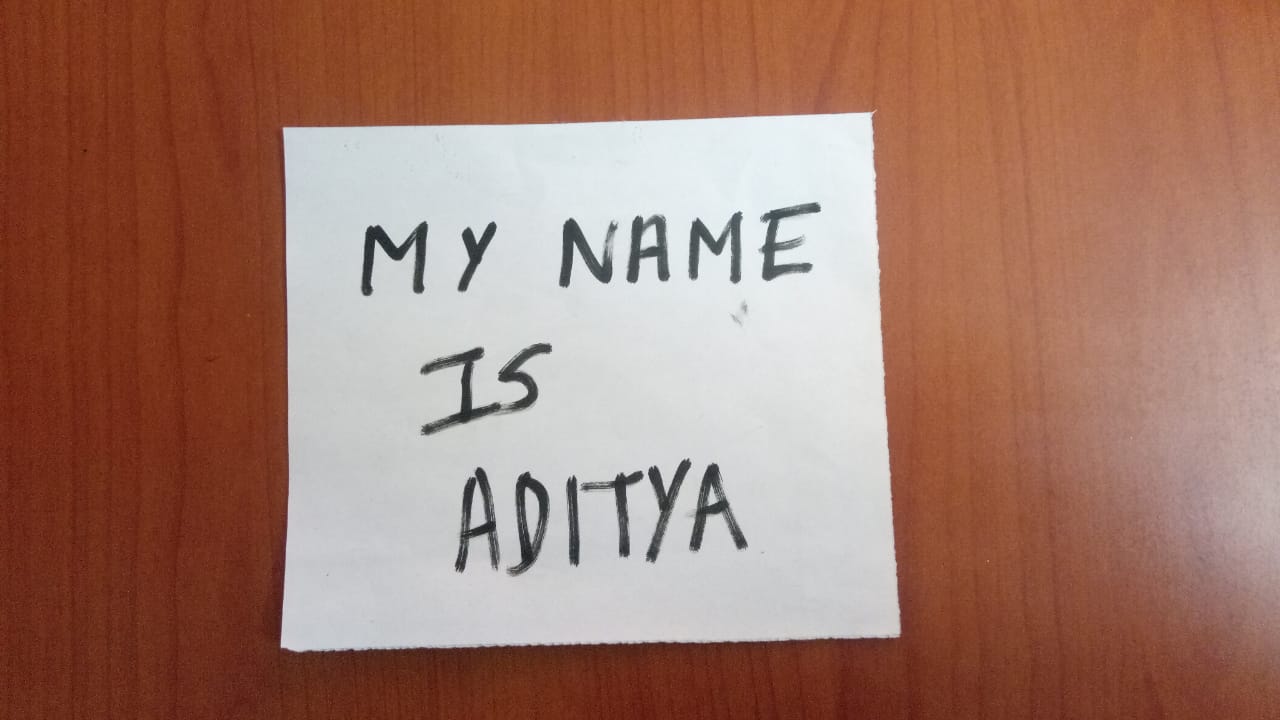
cv2.imshow("Scanned",dst)

# press q or Esc to close

cv2.waitKey(0)

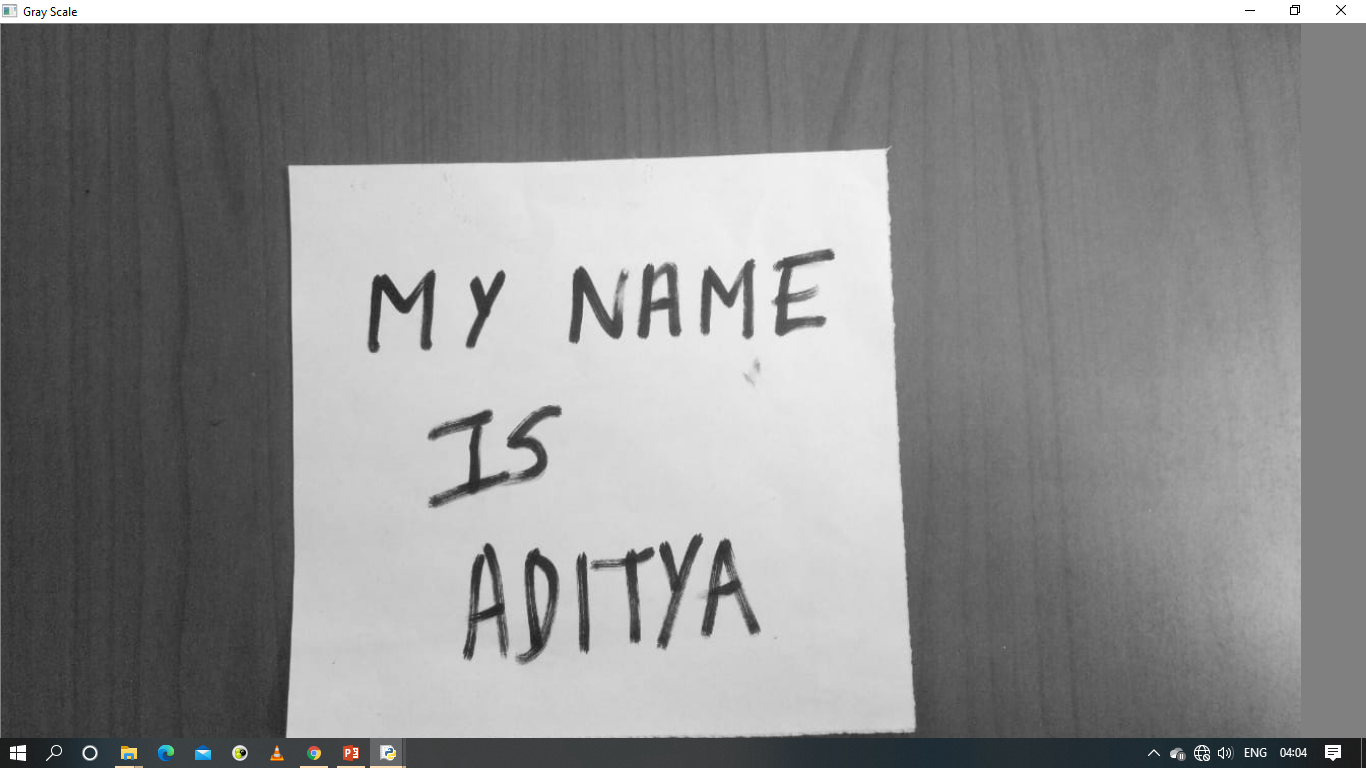
cv2.destroyAllWindows()

Original Image

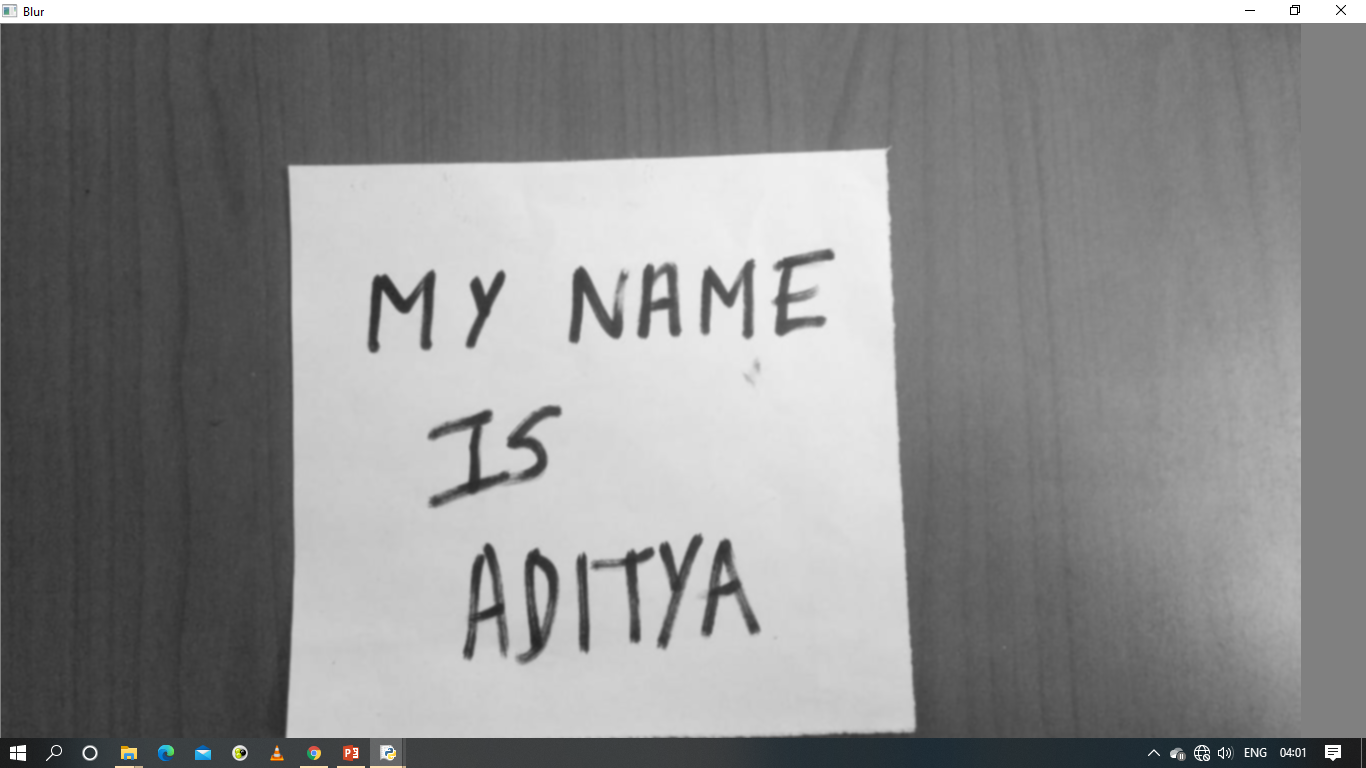


**Output after running the Python Code for Scanner.py**

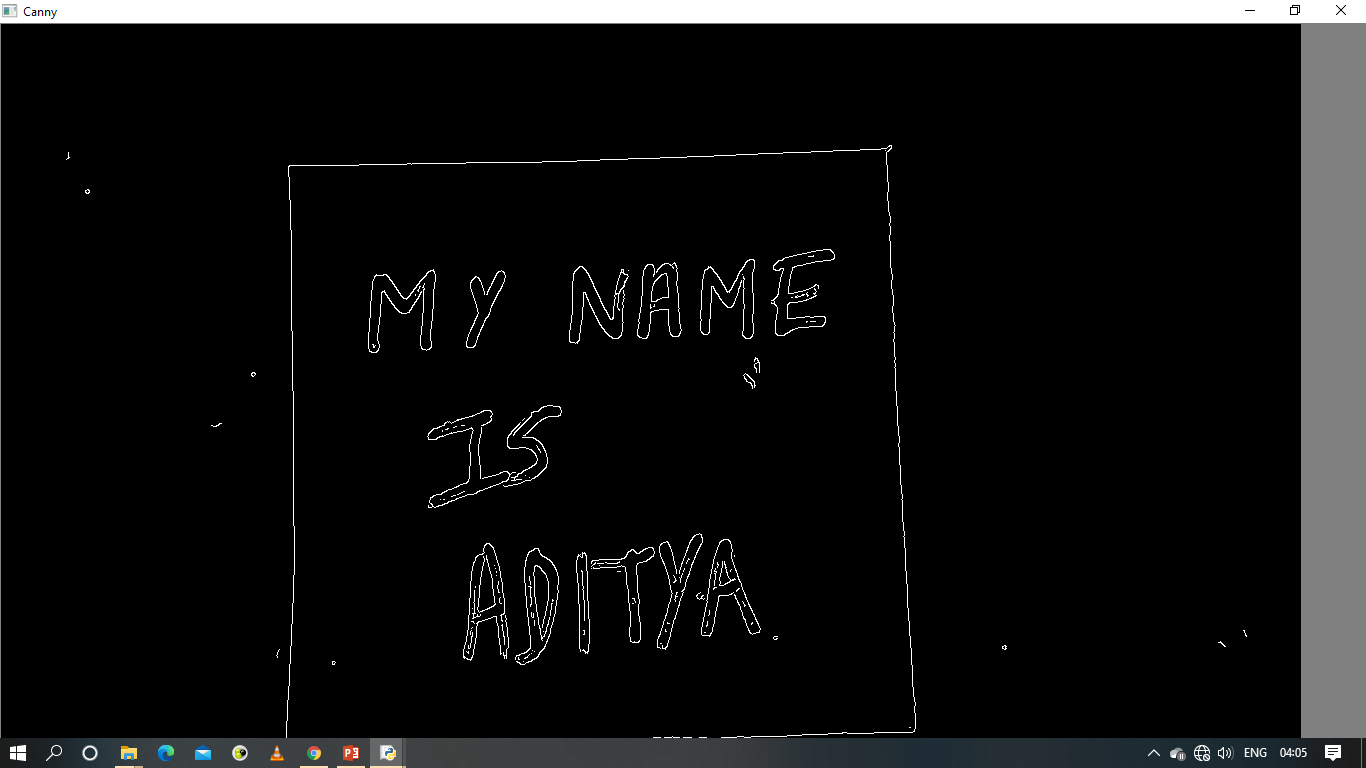
First we get the gray scale image entitled “Gray Scale” as output.



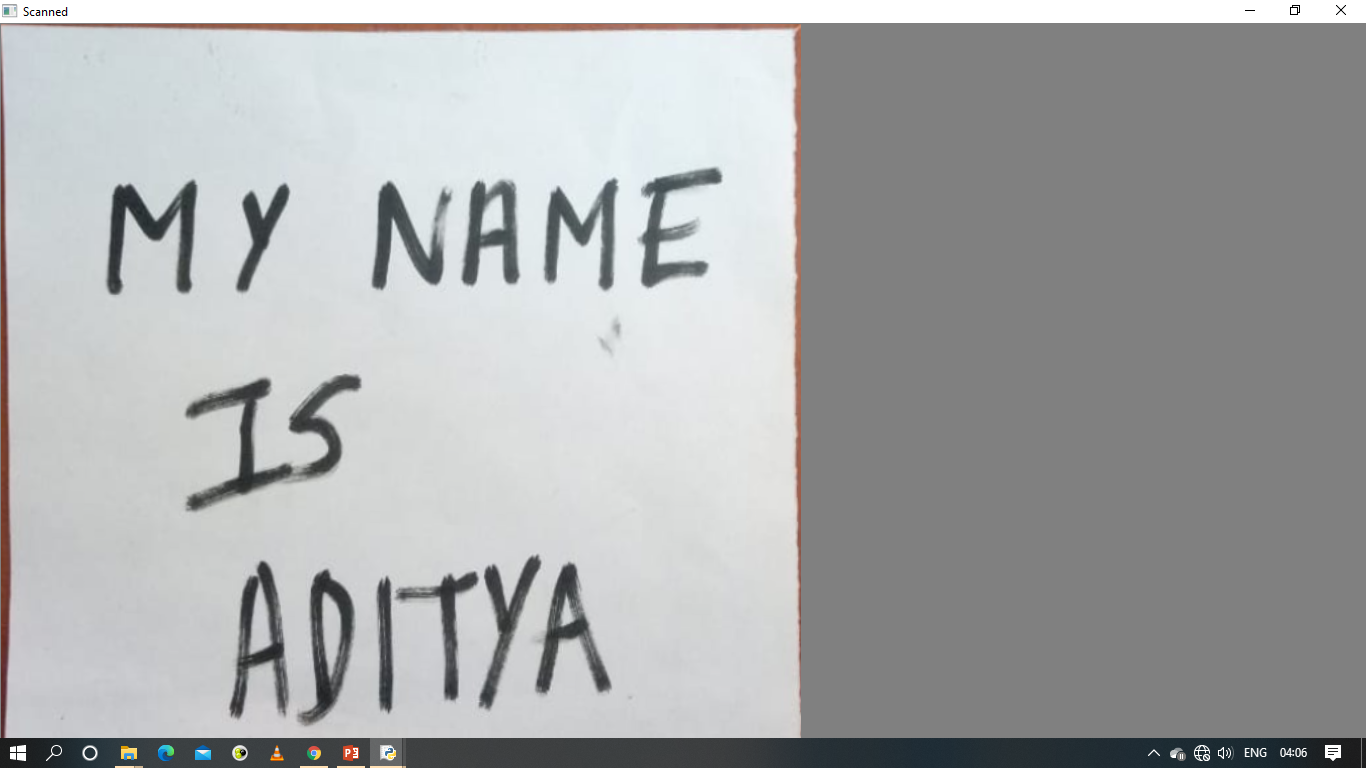
Secondly we get the Blurred Image entitled “Blur”.



Thirdly we get the Canny Detected Image entitled “Canny”.



Fourthly we get the Scanned Image entitled “Scanned” which is the final output.



**10. Image Segmentation**

Some computer vision problems require deeper understanding of the contents in the images. Classification and object detection may not be suitable to solve these problems. The need for an effective technique to solve some critical computer vision problems gives rise to the invention of Image Segmentation.

Every image is made up of a group of pixel values. Image Segmentation is the task of classifying an image at the pixel level. A machine is able to analyse an image more effectively by dividing it into different segments according to the classes assigned to each of the pixel values present in the image.

The distinct technique employed in Image Segmentation makes it applicable in solving critical computer vision problems. These are problems that require detailed information about the objects present in an image, details that cannot be provided by classifying the entire image or providing bounding boxes for the objects present in the image. Some of the major contributions of Image Segmentation include:

-Efficient vision system for driverless cars for an effective road scene’s understanding.

-Medical Image Segmentation provides segmentation of body parts for performing diagnostic tests.

* Satellite images' analysis.

Before performing this experiment, we need to know what Image Segmentation actually is.

* Every image is made up of a group of pixel values. Image Segmentation is the task of classifying an image at the pixel level.
* A machine is able to analyse an image more effectively by dividing it into different segments according to the classes assigned to each of the pixel values present in the image.

There are two major types of Image Segmentation:

**Semantic Segmentation**: Objects classified with the same pixel values are segmented with the same colormaps.

**Instance Segmentation**: It differs from semantic segmentation because different instances of the same object are segmented with different color maps.

**Our requirements for the experiment:**

Before performing this experiment, we need to download certain libraries and packages to implement our code successfully.

**tensorflow-gpu**

**opencv-python**

**scikit-image**

**pillow**

**pixellib**

Install PixelLib and its dependencies:

Pixellib is a library for performing segmentation of objects in images and videos. It supports the two major types of image segmentation:

1.Semantic segmentation

2.Instance segmentation

PixelLib supports two deep learning libraries for image segmentation which are Pytorch and Tensorflow.

Install the latest version tensorflow (tensorflow 2.0) with: pip3 install tensorflow

Install opencv-python with: pip3 install opencv-python

Install scikit-image with: pip3 install scikit-image

Install Pillow with: pip3 install pillow

Install Pixellib: pip3 install pixellib

**Implementation of Semantic Segmentation with PixelLib**

**Source Code:**

# The class for performing semantic segmentation is imported from pixelLib and we created an instance of the class.

import pixellib

from pixellib.semantic import semantic\_segmentation

segment\_image = semantic\_segmentation()

# In the code below we loaded the xception model trained on pascal voc for segmenting objects.

# The model can be downloaded from Github link mentioned below this comment line

#https://github.com/ayoolaolafenwa/PixelLib/releases/download/1.1/deeplabv3\_xception\_tf\_dim\_ordering\_tf\_kernels.h5

segment\_image.load\_pascalvoc\_model("deeplabv3\_xception\_tf\_dim\_ordering\_tf\_kernels.h5")

# Next we load the function to perform segmentation on an image. The function takes two parameters:

# path\_to\_image: - this is the path to the image to be segmented.

# output\_image\_name: - this is the path to save the segmented image. It will be saved in your current working directory.

# segment\_image.segmentAsPascalvoc("path\_to\_image", output\_image\_name = "path\_to\_output\_image")

segment\_image.segmentAsPascalvoc("sample.jpg", output\_image\_name = "output.jpg")

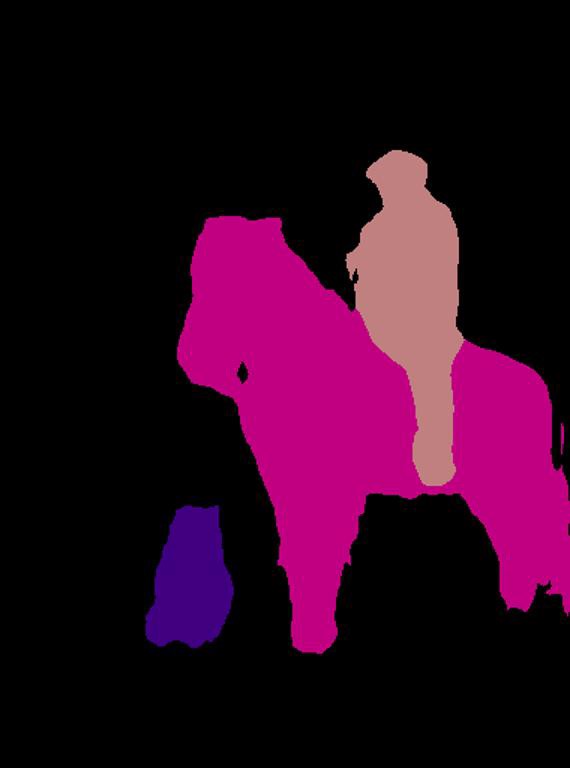
**Output:**

**sample.jpg**



The saved image after segmentation, the objects in the image are segmented.

**output.jpg**



Next we added the extra parameter **overlay** and set it to **true** and we obtained an image with a segmentation overlay on the objects.

**Source Code:**

import pixellib

from pixellib.semantic import semantic\_segmentation

import cv2

segment\_image = semantic\_segmentation()

segment\_image.load\_pascalvoc\_model("pascal.h5")

segmap, segoverlay = segment\_image.segmentAsPascalvoc("sample.jpg", overlay= True)

cv2.imwrite("img.jpg", segoverlay)

print(segoverlay.shape)

**Output:**

**sample.jpg**



**img.jpg**



**Implementation of Instance Segmentation with PixelLib**

Instance segmentation with PixelLib is based on Mask R-CNN framework.

**Source Code:**

import pixellib  
from pixellib.instance import instance\_segmentation  
  
segment\_image = instance\_segmentation()  
segment\_image.load\_model("mask\_rcnn\_coco.h5")   
segment\_image.segmentImage("sample2.jpg", output\_image\_name = "image\_new.jpg")

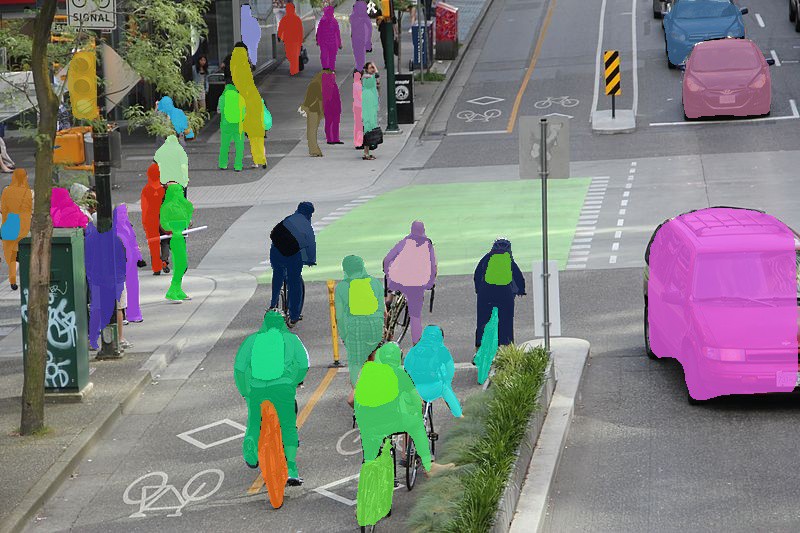
**Output:**

sample2.jpg



This is the saved image in your current working directory.

image\_new.jpg



We can implement segmentation with bounding boxes. This can be achieved by modifying the code.

We added an extra parameter **show\_bboxes** and set it to **true**, the segmentation masks are produced with bounding boxes.

The Mask R\_CNN model is trained on Microsoft Coco dataset, a dataset with 80 common object categories. The model can perform instance segmentation on these object categories.

**A list of the object categories present in Coco dataset:**

[‘BG’, ‘person’, ‘bicycle’, ‘car’, ‘motorcycle’, ‘airplane’, ‘bus’, ‘train’, ‘truck’, ‘boat’, ‘traffic light’, ‘fire hydrant’, ‘stop sign’, ‘parking meter’, ‘bench’, ‘bird’, ‘cat’, ‘dog’, ‘horse’, ‘sheep’, ‘cow’, ‘elephant’, ‘bear’, ‘zebra’, ‘giraffe’, ‘backpack’, ‘umbrella’, ‘handbag’, ‘tie’, ‘suitcase’, ‘frisbee’, ‘skis’, ‘snowboard’, ‘sports ball’, ‘kite’, ‘baseball bat’, ‘baseball glove’, ‘skateboard’, ‘surfboard’, ‘tennis racket’, ‘bottle’, ‘wine glass’, ‘cup’, ‘fork’, ‘knife’, ‘spoon’, ‘bowl’, ‘banana’, ‘apple’, ‘sandwich’, ‘orange’, ‘broccoli’, ‘carrot’, ‘hot dog’, ‘pizza’, ‘donut’, ‘cake’, ‘chair’, ‘couch’, ‘potted plant’, ‘bed’, ‘dining table’, ‘toilet’, ‘tv’, ‘laptop’, ‘mouse’, ‘remote’, ‘keyboard’, ‘cell phone’, ‘microwave’, ‘oven’, ‘toaster’, ‘sink’, ‘refrigerator’, ‘book’, ‘clock’, ‘vase’, ‘scissors’, ‘teddy bear’, ‘hair drier’, ‘toothbrush’]

**Source Code:**

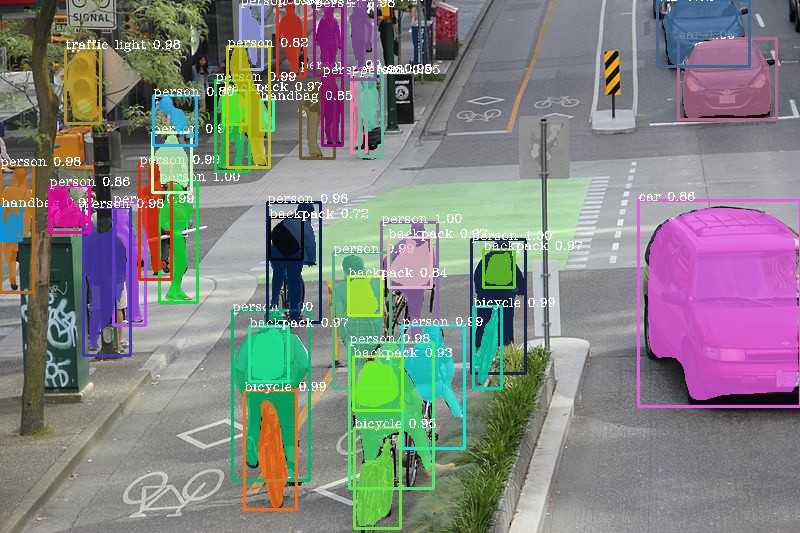
import pixellib  
from pixellib.instance import instance\_segmentation  
  
segment\_image = instance\_segmentation()  
segment\_image.load\_model("mask\_rcnn\_coco.h5")  
segment\_image.segmentImage("sample2.jpg", output\_image\_name = "new\_image2.jpg", show\_bboxes = True)

**Output:**

sample2.jpg



new\_image2.jpg



**Project Future Scope**

Our project is mainly concerned with basic operations on the image. So that user can get a requested visual appearance. Image processing is the use of computer algorithm to perform, improve or change some quality of the image. Nowadays, it is important in different fields.

The Python Image Processing Library adds image processing capabilities to the Python Interpreter. This library provides extensive file format, an efficient internal representation, and fairly powerful image processing tools. Actually, the Image Library designed for fast access.

The Python Image Library is ideal for image archival and batch processing application.  Here our project is providing an image processing tool through which user can give desired effects to an image. It accepts input via mouse and keyboard and gives output on the screen.

So, just one click by mouse, you will get the image with the expected result. In this project, we are providing different image processing tool with GUI so it makes easy for the user to give desired effects to the image. Image processing is booming technology.

Image processing has the lot of application in different fields like medical, traffic signals, computer science etc. In this project, we have implemented some of the image processing operations which will help the user to understand the concept of image processing. Therefore, the scope of this project is increasing.

The purpose of this project is to allow users to explore how digitized images can be processed using computer software.

Due to the fast development in the computer technology , future of the image- processing is going to be more flexible. Digital image processing has wide range of applications. In future it has been expected to be less expensive. Due to advancement of this technology there will be millions of robots in world. Advances in the image- processing and artificial intelligence will involve spoken commands, translation of languages, recognizing the finger prints, tracking of people and things, diagnosing medical conditions, performing surgery and automatically driving all forms of transport. With an increasing power of modern computing, the concept of computation can go beyond the present limits and in future, image processing technology will advance. The future trend in remote sensing will towards improved sensors that record the same scene in many spectral channels. Graphics data is becoming important nowadays in image processing app1ications. In future image processing technique will play import role in space also.

**Conclusion**

In a discpline of computer vision digital image processing is becoming popular day by day.

Edge detection is the initial step of recognizing object. Edges describe the boundaries of the object that is useful for identification of objects that are presented in the scene such as X-ray image. Edge detection is mostly use in image – segmentation . But all the edge detection techniques are not same. They are different from each other. In this paper we have made attempt to detect the edges in the input images using different methods. This different edge detection technique is implemented through the software ‘MATLAB R2008a’. It has been observed that canny edge detection is more superior than prewitt , sobel , log , roberts edge detector.

**References**

[1] Hands-On Image Processing with Python by Sandipan Dey

[2] Computer Vision: Algorithms and Applications (Texts in Computer Science) 2011th Edition

by Richard Szeliski

[3] Computer Vision: Models, Learning, and Inference 1st Edition by Simon J. D. Prince

[4] Computer Vision: A Modern Approach 2nd Edition by David Forsyth

[5] Introductory Techniques for 3-D Computer Vision by Emanuele Trucco

[6] Multiple View Geometry in Computer Vision 2nd Edition by Richard Hartley