****

**St. PETER’s ENGINEERING COLLEGE**

# Department of Computer Science and Engineering

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with ‘A’ Grade) Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100

# 2019-2023

**Text extraction from images**

A Mini Project Report Submitted

In partial fulfillment of the requirements for the award of the degree of

# Bachelor of Technology in Computer Science and Engineering

## by

|  |  |
| --- | --- |
| **MADALA.SUPRAJA** | **19BK1A05E9** |
| **VADE GEETHIKA** | **19BK1A05H0** |
| **KONDOJU SRUJAN** | **19BK1A05E5** |

Under the esteemed guidance of

## T .RAVI KUMAR

**Assistant Professor**

****

**St. PETER’s ENGINEERING COLLEGE**

(UGC-AUTONOMOUS)

(Autonomous Institution- UGC, Govt. of India)

(Affiliated to JNTUH, Hyderabad, Approved by AICTE, NBA &NAAC with ‘A’ Grade) Maisammaguda, Kompally, Dhulapally, Secunderabad – 500100

# CERTIFICATE

This is to certify that this is the bonafide record of the project entitled “Text extraction from images”, submitted by Student Madala Supaja (19BK1A05E9), Vade Geethika (19BK1A05H0) and Kondoju Srujan (19BK1A05E5) of B.Tech in the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science and Engineering, Department of CSE during the year 2021-2022. The results embodied in this project report have not been submitted to any other university or institute for the award of any degree or diploma.

**Internal Guide Head of the Department**

**T. Ravi Kumar Dr Amjan shaik**

**Assiatant Professor Professor**

**External Examiner**

# DECLARATION

We hereby declare that the project titled “**Text extraction from images**” submitted to St. Peter’s Engineering College (UGC Autonomous), affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a result of original research carried-out in this thesis. It is further declared that the project report or any part thereof has not been previously submitted to any University or Institute for the award of degree.

**MADALA.SUPRAJA(19BK1A05E9)**

**VADE.GEETHIKA(19BK1A05H0)**

**KONDOJU.SRUJAN(19BK1A05E5)**

## ACKNOWLEDGEMENT

We sincerely express our deep sense of gratitude to Dr. T.Ravi Kumar for his valuable guidance constant encouragement and cooperation during all phases of the project.

It is a great opportunity to render I was sincere thanks to Dr. Amjan Shaik Professor and Head of the Department of computer science and engineering for her timely guidance highly interactive attitude which helped us a lot in successful execution of the project.

We are also extremely thankful to our Principal Dr. K. Sreelatha, who stood as an inspiration behind this project and heartfelt thanks for her endorsement and valuable suggestions.

We are greatly indebted to our project coordinator Dr. Deepan professor prefer providing bear valuable advice, constructive suggestions, positive attitude and encouragement without which it would not be impossible to complete this project. Behold that we can build upon the experience and knowledge that we have gained an make a valuable contribution towards the growth of the country in coming future.

**M.SUPRAJA (19BK1A05E9)**

**V.GEETHIKA (19BK1A05H0)**

**K.SRUJAN (19BK1A05E5)**

## ABSTRACT

Text Extraction plays a major role in finding vital and valuable information. Text extraction

involves detection, localization, tracking, binarization, extraction, enhancementandrecognition of the text from the given image. These text characters are difficult to be detected and recognized due to their deviation of size, font, style, orientation, alignment, contrast, complex colored, textured background. Due to rapid growth of available multimedia documents and growing requirement for information, identification, indexing and retrieval, many researches have been done on text extraction in images. Several techniques have been developed for extracting the text from an image. EasyOCR is a framework in the proposed methodology which includes image pre processing deep learning model recognition, and image post processing.

**Keywords: Text Recognition, Text Extraction, EasyOCR, Pre-processing, Post-processing,**

**Feature Extraction.**

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| S.No. | Title | Page No |
| 1 | Introduction |  |
|  | 1.1 Purpose, Aim, Objectives |  |
|  | 1.2 Background of the project |  |
|  | 1.3 Scope of the project |  |
|  | 1.4 Modules Description |  |
| 2 | System analysis |  |
|  | 2.1 Hardware and Software requirements |  |
|  | 2.1.1 Hardware requirements |  |
|  | 2.1.2 Software requirements |  |
|  | 2.2 Software requirement specification |  |
|  | 2.2.1 SRS |  |
|  | 2.2.2 Role of SRS |  |
|  | 2.2.3 Scope |  |
|  | 2.2.4 Exixting system |  |
|  | 2.2.4.1 Drawbacks of existimg system |  |
|  | 2.2.5 Proposed system |  |
|  | 2.2.5.1 Image Processing |  |
|  | 2.2.5.2 Text Recognition software |  |
|  | 2.2.5.3 OCR |  |
|  | 2.2.5.4 Pre-processing Module |  |
|  | 2.2.5.5 Noise removal |  |
|  | 2.2.5.6 Filtering |  |

|  |  |  |
| --- | --- | --- |
|  | 2.2.5.7 Segmentation |  |
|  | 2.2.5.8 Feature Extraction and classification |  |
|  | 2.2.5.9 Post-processing Module |  |
| 3 | Technologies |  |
|  | 3.1 Python |  |
|  | 3.2 Development tools |  |
| 4 | System Design and UML diagrams |  |
|  | 4.1 Software design |  |
|  | 4.2 Architecture |  |
|  | 4.3 Unified modeling language |  |
|  | 4.3.1 Building blocks of UML |  |
|  | 4.3.1.1 Things in the UML |  |
|  | 4.3.1.2 Relationship in the UML |  |
|  | 4.3.2 UML diagrams |  |
|  | 4.3.2.1 Data flow diagram |  |
|  | 4.3.2.2 Flowchart diagram |  |
|  | 4.3.2.3 Use case diagram |  |
|  | 4.3.2.4 Sequence diagram |  |
|  | 4.3.2.5 Deployment diagram |  |
|  | 4.3.2.6 Block diagram |  |
| 5 | Input/Output design |  |
|  | 5.1 Input design |  |
|  | 5.2 Output design |  |
| 6 | Implementation |  |
| 7 | Testing |  |
|  | 7.1 Testing Objectives |  |
|  | 7.2 Testing methodologies |  |
|  | 7.3 User training |  |
|  | 7.4 Maintainence |  |
|  | 7.5 Testing strategy |  |
|  | 7.5.1 System testing |  |
|  | 7.5.2 Unit testing |  |
| 8 | Output Screens |  |
| 9 | Conclusion and Future Scope |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Name** | **Page no** |
| 1.1 | EasyOCR Framework | 3 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF OUTPUTS**

**FIGURE.NO NAME PG.NO**

**8.1 HOME SCREEN 47**

**82. SETTINGS 47**

**8.3 HELP 48**

**8.4 CUSTOM MESSAGE HELP 48**

### CHAPTER 1 INTRODUCTION

**1.1 PURPOSE, AIM, OBJECTIVES:**

Digital images are getting popular rapidly. Every day, many images have been generated by many groups like students, engineer, doctors, according to their varying needs. They can access images based on its primitive features or associated text. Text present in such images can provide meaningful information. The current technology is restricted to extracting text against clean backgrounds. There is a need for a system to extract text from general backgrounds. We aim to retrieve the content and summarize the visual information automatically from images.

Text extraction can be achieved by applying text detection that identifies image parts containing text, text localization finds the exact position of the text, text segmentation separates the text from its background and binarization process converts the coloured images into binary. On this binary image, character recognition is applied to convert it into ASCII text.

* Text extraction is used in creating e-books from scanned books, image searching from a collection of visual data etc.
* It is very useful for describing the contents of an image;
* It can be easily extracted compared to other semantic contents;
* It enables applications such as keyword-based image search, and text-based image indexing.

* 1. **BACKGROUND OF THE PROJECT**

This project aims to develop a software which is used to extract the text from the backgrounds. The current technology is restricted to extracting text against clean backgrounds. There are various applications in which text extraction is useful. The role of text detection is to find the image regions containing only text that can be directly highlighted to the user or fed into an optical character reader module for recognition. The information from these image documents would give higher efficiency and ease of access if it is converted to text form. EasyOCR is a framework in the proposed methodology which includes image pre processing deep learning model recognition, and image post processing. This experiment proves that the high in accuracy and efficiently compared to previous methods. There is also a possibility in finding solutions for extracting text from unformatted documents and updating it automatically into the database. The aim of this project is to explore the task of classifying handwritten text, invoices, vehicles detection and many more into an understandable text.

* 1. **SCOPE OF THE PROJECT**

Text extraction from document images is one of the most widely studied topics in Image. Analysis and Optical Character Recognition. These extractions of document images can be used for document analysis, content analysis, document retrieval and many more. Many complex text extracting processes Maximization Likelihood (ML), Edge point detection, Corner point detection etc. are used to extract text documents from images. In this article, the corner point approach was used. To extract document from images we used a very simple approach based on EasyOCR.Text extraction is used in creating e-books from scanned books, image searching from a collection of visual data etc. It can be easily extracted compared to other semantic contents. This method is very fast and versatile, it can be used to detect various languages, handwriting and even images with a lot of noise and blur. Even though it is a very simple program the precision of this method is closer or higher than 90%. In conclusion, this method helps in more accurate and less complex detection of text from document images.

**1.4. MODULES DESCRIPTION:**

Modules used in our project are

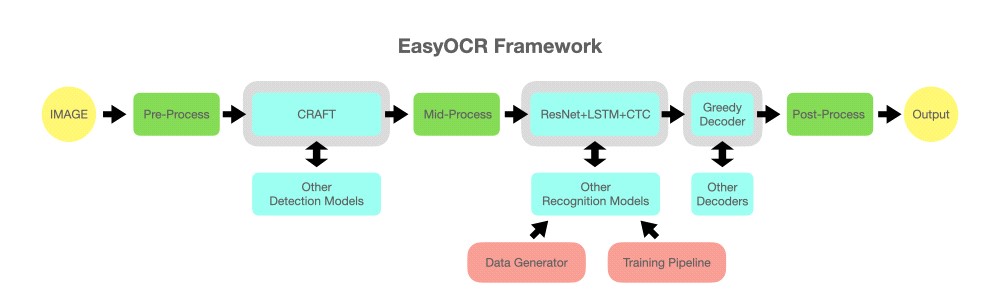
* EasyOCR
* Matplotlib
* Opencv-python
* Numpy

**EasyOCR:**

OCR is formerly known as Optical Character Recognition which is revolutionary for the digital world nowadays. OCR is actually a complete process under which the images/documents which are present in a digital world are processed and from the text are being processed out as normal editable text.

OCR is a technology that enables you to convert different types of documents, such as scanned paper documents, PDF files, or images captured by a digital camera into editable and searchable data.

EasyOCR is actually a python package that holds PyTorch as a backend handler. EasyOCR like any other OCR (tesseract of Google or any other) detects the text from images but in my reference, while using it I found that it is the most straightforward way to detect text from images also when high end deep learning library (PyTorch) is supporting it in the backend which makes it accuracy more credible. EasyOCR supports 42+ languages for detection purposes. EasyOCR is created by the company named Jaided AI company.



**FIG 1.1 EasyOCR Framework**

PyTorch is an [open source](https://en.wikipedia.org/wiki/Open_source) [machine learning](https://en.wikipedia.org/wiki/Machine_learning) [framework](https://en.wikipedia.org/wiki/Software_framework) based on the [Torch](https://en.wikipedia.org/wiki/Torch_(machine_learning)) library, used for applications such as [computer vision](https://en.wikipedia.org/wiki/Computer_vision) and [natural language processing,](https://en.wikipedia.org/wiki/Natural_language_processing) primarily developed by [Meta AI.](https://en.wikipedia.org/wiki/Meta_AI) Although the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) interface is more polished and the primary focus of development.

PyTorch defines a class called Tensor (torch.Tensor) to store and operate on homogeneous multidimensional rectangular arrays of numbers. PyTorch Tensors are similar to [NumPy](https://en.wikipedia.org/wiki/NumPy) Arrays, but can also be operated on a [CUDA-](https://en.wikipedia.org/wiki/CUDA)capable [Nvidia](https://en.wikipedia.org/wiki/Nvidia) GPU. PyTorch supports various sub-types of Tensors.

**PyTorch provides two high-level features:**

* Tensor computing (like [NumPy)](https://en.wikipedia.org/wiki/NumPy) with strong acceleration via [graphics processing units](https://en.wikipedia.org/wiki/Graphics_processing_unit) (GPU)
* [Deep neural networks](https://en.wikipedia.org/wiki/Deep_neural_networks) built on a tape-based [automatic differentiation](https://en.wikipedia.org/wiki/Automatic_differentiation) system

**Opencv-python:**

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as [Numpy](https://www.geeksforgeeks.org/python-numpy/) which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

In OpenCV, the CV is an abbreviation form of a computer vision, which is defined as a field of study that helps computers to understand the content of the digital images such as photographs and videos.

The purpose of computer vision is to understand the content of the images. It extracts the description from the pictures, which may be an object, a text description, and threedimension model, and so on. For example, cars can be facilitated with computer vision, which will be able to identify and different objects around the road, such as traffic lights, pedestrians, traffic signs, and so on, and acts accordingly.

* It is used to overlay images with respective recognized texts later.
* It creates bounding boxes on the text.

Computer vision allows the computer to perform the same kind of tasks as humans with the same efficiency. There are a two main task which are defined below:

* Object Classification - In the object classification, we train a model on a dataset of particular objects, and the model classifies new objects as belonging to one or more of your training categories.
* Object Identification - In the object identification, our model will identify a particular instance of an object.

**Matplotlib:**

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

Matplotlib was originally written by [John D. Hunter.](https://en.wikipedia.org/wiki/John_D._Hunter) Since then, it has an active development community and is distributed under a [BSD-style license.](https://en.wikipedia.org/wiki/BSD_licenses) Michael Droettboom was nominated as matplotlib's lead developer shortly before John Hunter's death in August 2012 and was further joined by Thomas Caswell. Matplotlib is a [NumFOCUS](https://en.wikipedia.org/w/index.php?title=NumFOCUS&action=edit&redlink=1) fiscally sponsored project.

It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks.

Matplotlib along with NumPy can be considered as the open-source equivalent of MATLAB.

**Numpy:**

NumPy, which stands for [Numerical Python,](https://www.mygreatlearning.com/blog/python-numpy-tutorial/) is a library consisting of multidimensional array objects and a collection of routines for processing those arrays.

Using NumPy, mathematical and logical operations on [arrays c](https://www.mygreatlearning.com/blog/what-is-an-array-learn-more-in-one-read/)an be performed.

NumPy is a Python package. It stands for ‘Numerical Python’. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Numeric, the ancestor of NumPy, was developed by Jim Hugunin. Another package Numarray was also developed, having some additional functionalities. In 2005, Travis Oliphant created NumPy package by incorporating the features of Numarray into Numeric package. There are many contributors to this open-source project.

**Operations using NumPy**

Using NumPy, a developer can perform the following operations −

* Mathematical and logical operations on arrays.
* Fourier transforms and routines for shape manipulation.
* Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

NumPy – A Replacement for MatLab. It is open-source, which is an added advantage of NumPy.

**CHAPTER 2**

### SYSTEM ANALYSIS

In this chapter, we will discuss and analyze about the developing process of Text extraction from images including software requirement specification (SRS) and comparison between existing and proposed system. The functional and non-functional requirements are included in SRS part to provide complete description and overview of system requirement before the developing process is carried out. Besides that, existing vs. proposed provides a view of how the proposed system will be more efficient than the existing one.

**2.1 HARDWARE AND SOFTWARE REQUIREMENTS**

**2.1.1 HARDWARE REQUIREMENTS:**

Processor : Intel core3 or higher.

RAM : Minimum of 4GB or higher.

Hard disk : 1TB or above.

**2.1.2 SOFTWARE REQUIREMENTS:**

Technology/Language : Python

System : Windows, Linux, Mac.

Software tool : EasyOCR

**2.2 SOFTWARE REQUIREMENT SPECIFICATION:**

**2.2.1 SRS:**

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

* Problem/Requirement Analysis:

The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.

* Requirement Specification:

Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

**2.2.2 ROLE OF SRS:**

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

**2.2.3 SCOPE:**

This document is the only one that describes the requirements of the system. It is meant for the use by the developers, and will also be the basis for validating the final delivered system. Any changes made to the requirements in the future will have to go through a formal change approval process. The developer is responsible for asking for clarifications, where necessary, and will not make any alterations without the permission of the client.

**2.2.4 EXISTING SYSTEM:**

Off-line handwriting recognition involves the automatic conversion of text in an image into letter codes which are usable withincomputer and text-processing applications. The data obtained by this form is regarded as a static representation of handwriting. Off-line handwriting recognition is comparatively difficult, as different people have different handwriting styles. And, as oftoday, OCR engines are primarily focused on machine printed text and ICR for hand "printed" (written in capital letters) text. Pranob, K. Charles, V.Harish, M.Swathi ,CH. Deepthi : A Review on the Various Techniques used for Optical CharacterRecognition. In this project the design of OCR systems are discussed. It presents the techniques that are slow which provide better results in nature and also the fast techniques that provide inefficient results. The first prominentpiece of OCR software was invented by Ray Kurzweil in 1974 as the software allowed for recognition for any font. This software used a more developed use of the matrix method (pattern matching). Essentially, this would compare bitmaps of the templatecharacter with the bitmaps of the read character and would compare them to determine which character it most closely matchedwith.

Papers Referred:

1. “Extracting Text from Image Document and Displaying ITS Related Information”, K. N. Natei, journal of Engineering Research and Application:

Image Text is the text information embedded or written in image of different form. Image text canbe found in captured images, scanned documents, magazines, newspapers, posters etc. These image texts are highly availablenowadays and they are very important in representing, describing and transferring information which help peoples incommunication, solving problems, availability, creation of new types of jobs, cost effectiveness, productivity, globalization andcultural gap etc. The information from these image documents would give higher efficiency and ease of access if it is convertedto text form. The process by which Image Text converted into plain text that computer can recognize its ASCII character is TextExtraction.

1. “Text Recognition using Image Processing”, International journal of Advanced Research in Computer Science: TextRecognition using OCR involves computer system designed to translate images of typewritten text into machine editable text orto translate pictures of characters into a standard encoding scheme representing them. OCR began as a field of research inartificial intelligence and computational vision. Text Recognition used in official task in which the large data have to type likepost offices, banks, colleges etc., in real life applications where we want to collect some information from text written image.

**2.2.4.1 DRAWBACKS OF EXISTING SYSTEM:**

The drawbacks of the project are

1)Sensitive to variations in sizing.

2)The distinctions between each individual's way of writing.

3)Inefficient Results with less probability

**2.2.5 PROPOSED SYSTEM:**

In the devised model as shown the idea proposed is to take in a number of images of documents like identity proofs of individualsand classify them into classes, such as passport and license. Once, the images are classified, they are subjected to the textextraction module. The text data are extracted from the classified images. The extracted credentials from the images are thenstored in the database. Text Extraction Text extraction is implemented using the EasyOCR module which contains an PyTorch module which acts as a backend handler. PyTorch is a prerequistie for the EasyOCR module. This module is required for extracting text from images, it is a technology that enables us to convert different types of documents, such as scanned paper documents, pdf files or images captured by a digital camera into editable and searchable data.

Optical Character Recognition (OCR), OCR helps to scan printed documents and retrieve texts from images. It makes it possible to extract text from an image and edit it without having to type it all manually.

**Benefits of OCR in text extraction:**

Beyond reasonable doubt, products that use OCR technology gives room for speed and accuracy in extracting data from images amongst other advantages.

* **Usability**: The converted text from an image can be edited and used for your benefit.
* **Saves time**: OCR has helped to eliminate the manual task of storing files and extracting data. With OCR, data is extracted and saved in no distant time.
* **Image to text conversion**: OCR provides a major benefit of converting images into text for different purposes.
* **Data security**: OCR and data security hand in hand. With OCR, the data extracted is stored securely and can be easily accessed at any time.

The use of OCR is relevant in so many industries for different reasons.

* **Banking:** OCR is used to capture account information, detect fraudand ensure seamless flow of operation.
* **Legal**: OCR is used to digitise printed documents.
* **Healthcare**: OCR is used to extract reports from data like X-rays and hospital records.
* **Business:** OCR is used to extract serial codes from phones.

**2.2.5.1 Image Processing:**

Image processing is analysis and manipulation of a digitized image, so as to enhance its quality with the help of mathematical operations by using any kind of signal processing where the input is a picture or an image or a video frame. The output of image processing will be either a picture or set of characters or parameters associated with the given input image. This is a set of computational techniques for analyzing, enhancing, compressing and reconstructing image.

Image Processing is set of computational techniques for analyzing, enhancing, compressing, and reconstructing images. Its main components are importing, in which an image is captured through scanning or digital photography; analysis and manipulation of the image, accomplished using various specialized software applications; and output. Image processing has extensive applications in many areas, including astronomy, medicine, industrial robotics, and remote sensing by satellites. Image Processing provides a comprehensive set of reference-standard algorithms and workflow apps for image analysis, visualization, and algorithm development.

Image Processing can interactively segment image data, compare image registration techniques, and batch-process large data sets.

Image processing generally involves three steps:

* Importing and loading the image by using image acquisition tools.
* Analyzing and manipulating image to extract the information.
* Output the result. The result might be the image or a picture altered in some way or it may be a report based on analysis of the image

**2.2.5.2 Text Recognition Software:**

Text recognition software is an automatic tool that recognizes text within image files, so that you do not have to retype it yourself. This is especially important when working with scanned documents and image files that contain text that might not be recognized by the computer. Text recognition software, known as OCR, resolves this problem by making the documents text-searchable.

Text recognition software can be used for a variety of different purposes. It is especially popular for working with scanned documents. For example, sometimes you may need to work with a text that was generated before information was stored in digital format. Since you cannot access a computer-generated version of the text, you may need to scan the original document and use text recognition software to make it convenient to work with. Once we have detected or extracted area of target image, we can recognize the characters or text with the help of training data. The detected image area compared by available training dataset which outputs the exact character written in image Text in images can exhibit many variations with respect to the following properties:

1. **Geometry:**

Size: Although the text size can vary a lot, assumptions can be made depending on the application domain.

Alignment: The characters in the caption text appear in clusters and usually lie horizontally, although sometimes they can appear as non-planar texts as a result of special effects. This does not apply to Text, which can have various perspective distortions. Text can be aligned in any direction and can have geometric distortions.

Inter-character distance: Characters in a text line have a uniform distance between them.

1. **Colour:**

The characters in a text line tend to have the same or similar colours. This property makes it possible to use a connected component-based approach for text detection. Most of the research reported till date has concentrated on finding ‘text strings of a single colour (monochrome)’. However, video images and other complex colour documents can contain ‘text strings with more than two colours (polychrome)’ for effective visualization, i.e., different colours within one word.

1. **Motion:**

The same characters usually exist in consecutive frames in a video with or without movement. This property is used in text tracking and enhancement. Caption text usually moves in a uniform way: horizontally or vertically. Scene text can have arbitrary motion due to camera or object movement.

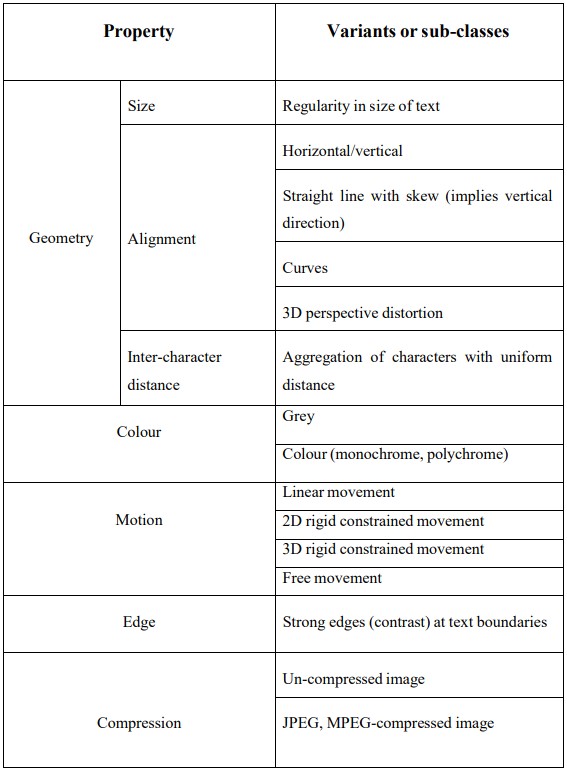
1. **Edge:**

Most caption and scene text are designed to be easily read, thereby resulting in strong edges at the boundaries of text and background.

• **Compression:**

Many digital images are recorded, transferred, and processed in a compressed format. Thus, a faster TIE (Text Information Extraction) system can be achieved if one can extract text without decompression.

**Properties of Text in Image:**



**Table 2.1: Properties of text in images**

**2.2.5.3 OCR:**

OCR (Optical Character Recognition) is the use of technology to distinguish printed or handwritten text characters inside digital images of physical documents, such as a scanned paper document. The basic process of OCR involves examining the text of a document and translating the characters into code that can be used for data processing.

OCR is sometimes also referred to as text recognition.

OCR systems are made up of a combination of hardware and software that is used to convert physical documents into machine-readable text. Hardware, such as an optical scanner or specialized circuit board is used to copy or read text while software typically handles the advanced processing. Software can also take advantage of artificial intelligence (AI) to implement more advanced methods of Intelligent Character Recognition (ICR), like identifying languages or styles of handwriting.

The major OCR technology providers began to tweak OCR systems to deal more efficiently with specific types of input. Beyond an application-specific lexicon, better performance may be had by taking into account business rules, standard expression, rich information contained in colour images. This strategy is called "Application-Oriented OCR" or "Customized OCR", and has been applied to OCR of invoices, screenshots, ID cards, driver licenses, and automobile manufacturing.

**2.2.5.4 Pre-processing Module:**

The paper document is generally scanned by the optical scanner and isconverted in to the form of a picture. At this stage we have the data in the form of image and this image can be further analysed so that’s the important information can be retrieved. The image resulting from the scanning process may contain a certain amount of noise. Depending on the resolution on the scanner and the success of the applied technique for thresholding, the characters may be smeared or broken. Some of these defects, which may later cause poor recognition rates, can be eliminated by using a pre-processor to smooth the digitized characters. The smoothing implies both filling and thinning. Filling eliminates small breaks, gaps and holes in the digitized characters, while thinning reduces the width of the line. The most common techniques for smoothing, moves a window across the binary image of the character, applying certain rules to the contents of the window. So, to improve quality of the input image, few operations are performed for enhancement of image such as noise removal, normalization, binarization etc.

**2.2.5.5 Noise Removal:**

Image noise is an unavoidable side-effect occurring as a result of image capture, more simply understood as inaudible, yet inevitable fluctuations. In a digital camera, if the light which entersthe lensmisaligns with the sensors, itwill create image noise. Even 17 if noise is not so obviously visible in a picture, some kind of image noise is bound to exist. Every type of electronic device receives and transmits some noise and sends it on to what it is creating. When the images are transmitted over channels, they are corrupted with impulse noise due to noisy channels. This impulse noise consists of large positive and negative spikes. The positive spikes have values much larger than the background and thus they appear as bright spots, while the negative spikes have values smaller than the background and they appear as darker spots. Both the spots for the positive and negative spikes are visible to the human eye. Also, Gaussian type of noise affects the image.

**2.2.5.6 Filtering:**

Filters are required for removing noises before processing. There are lots of filters in the paper to remove noise. They are of many kinds as linear smoothing filter, median filter, wiener filter and Fuzzy filter. In this filtering technique, the three primaries (R, G and B) are done separately. It is followed by some gain to compensate for attenuation resulting from the filter. The filtered primaries are then combined to form the coloured image. This process is very simple. This approach shown in figure below as.

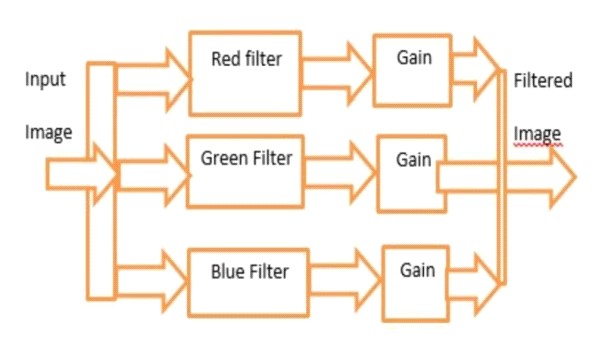


FIG 2.1 Filtering

**2.2.5.7 Segmentation:**

Image segmentation is the process of partitioning a digital image into multiple segments (sets of pixels, also known asimage objects). The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyse. There are various approaches for text and picture segmentation in image document namely region based approach and texture based approach. In the region based approach, we consider each pixel in the image and assign it to a particular region or object. This approach is basically divided into two subcategories: edge based [2, 3, 12] and connected component based. The text-regions in a document image can be detected either by region based and texture-based methods. They are relatively independent of changes in text size and orientation, but having difficulties with complex images with non-uniform backgrounds, for example, if a text string touches a graphical object in the original image, they may form one connected component in the resultant binary image.

**2.2.5.8 Feature Extraction and Classification:**

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.

The objective of feature extraction is to capture the essential characteristics of the symbols, and it is generally accepted that this is one of the most difficult problems of pattern recognition. The most straight forward way of describing a character is by the actual raster image. Another approach is to extract certain features that still characterize the symbols, but leaves out the unimportant attributes.

The techniques for extraction of such features are often divided into three main groups, where the features are found from:

* The distribution of points.
* Transformations and series expansions.
* Structural analysis.

The different groups of features may be evaluated according to their sensitivity to noise and deformation and the ease of implementation and use. The criteria used in this evaluation are the following:

* Robustness:

1. Noise. Sensitivity to disconnected line segments, bumps, gaps, filled loops etc.
2. Distortions. Sensitivity to local variationslike rounded corners, improper protrusions, dilations and shrinkage.
3. Style variation. Sensitivity to variation in style like the use of different shapes to represent the same character or the use of serifs, slants etc.
4. Translation. Sensitivity to movement of the whole character or its components.
5. Rotation. Sensitivity to change in orientation of the characters.

• Practical use:

1. Speed of recognition.
2. Complexity of implementation.
3. Independence - The need of supplementary techniques.

Feature extraction includes several convolution layers followed by max-pooling and an activation function. The classifier usually consists of fully connected layers.

Image classification is acceptance of the input image and the following definition of its class. This is a skill that people learn from their birth and are able to easily determine that the image in the picture is an elephant. But the computer sees the pictures quite differently.



Fig 2.2: Difference of the view of seeing between Human and Computer

Instead of the image, the computer sees an array of pixels. For example, ifimage size is 300 x 300. In this case, the size of the array will be 300x300x3. Where 300 iswidth, next 300 is height and 3 is RGB channel values. The computer is assigned a value from 0 to 255 to each of these numbers. This value describesthe intensity of the pixel at eachpoint.

**2.2.5.9 Post-processing Module:**

Post processing attempts to increase the quality of a mask image. Post processing is performed with the help of Morphology. The morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. It is the final step of recognition system being discussed. It prints the corresponding characters which were recognized in the structured text form which is done by the calculation of equivalent ASCII value using recognition index of the test samples.

Grouping:

The result of plain symbol recognition on a document, is a set of individual symbols. However, these symbols in themselves do usually not contain enough information. Instead we would like to associate the individual symbols that belong to the same string with each other, making up words and numbers. The process of performing this association of symbols into strings, is commonly referred to as grouping. The grouping of the symbols into strings is based on the symbols’ location in the document. Symbols that are found to be sufficiently close are grouped together. For fonts with fixed pitch the process of grouping is fairly easy as the position of each character is known. For typeset characters the distance between characters are variable. However, the distance between words are usually significantly larger than the distance between characters, and grouping is therefore still possible. The real problems occur for handwritten characters or when the text is skewed.

Error-detection and correction:

Up until the grouping each character has been treated separately, and the context in which each character appears has usually not been exploited. However, in advanced optical text recognition problems, a system consisting only of single-character recognition will not be sufficient. Even the best recognition systems will not give 100% percent correct identification of all characters, but some of these errors may be detected or even corrected by the use of context. There are two main approaches, where the first utilizes the possibility of sequences of characters appearing together. This may be done by the use of rules defining the syntax of the word, by saying for instance that after a period there should usually be a capital letter. Also, for different languages the probabilities of two or more characters appearing together in a sequence can be computed and may be utilized to detect errors. For instance, in the English language the probability of a “k” appearing after an “h” in a word is zero, and if such a combination is detected an error is assumed. Another approach is the use of dictionaries, which has proven to be the most efficient method for error detection and correction. Given a word, in which an error may be present, the word is looked up in the dictionary. If the word is not in the dictionary, an error has been detected, and may be corrected by changing the word into the most similar word. Probabilities obtained from the classification, may help to identify the character which has been erroneously classified. If the word is present in the dictionary, this does unfortunately not prove that no error occurred. An error may have transformed the word from one legal word to another, and such errors are undetectable by this procedure. The disadvantage of the dictionary methods is that the searches and comparisons implied are time-consuming.

### CHAPTER 3 TECHNOLOGIES

**3.1 PYTHON:**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

Python is often compared to other interpreted languages such as Java, JavaScript, Perl, Tcl, or Smalltalk. Comparisons to C++, Common Lisp and Scheme can also be enlightening. In this section I will briefly compare Python to each of these languages. These comparisons concentrate on language issues only. In practice, the choice of a programming language is often dictated by other real-world constraints such as cost, availability, training, and prior investment, or even emotional attachment. Since these aspects are highly variable, it seems a waste of time to consider them much for this comparison.

Python programs are generally expected to run slower than Java programs, but they also take much less time to develop. Python programs are typically 3-5 times shorter than equivalent Java programs. This difference can be attributed to Python's built-in high-level data types and its dynamic typing. For example, a Python programmer wastes no time declaring the types of arguments or variables, and Python's powerful polymorphic list and dictionary types, for which rich syntactic support is built straight into the language, find a use in almost every Python program. Because of the run-time typing, Python's run time must work harder than Java's. For example, when evaluating the expression a+b, it must first inspect the objects a and b to find out their type, which is not known at compile time. It then invokes the appropriate addition operation, which may be an overloaded user-defined method. Java, on the other hand, can perform an efficient integer or floating point addition, but requires variable declarations for a and b, and does not allow overloading of the + operator for instances of user-defined classes.

For these reasons, Python is much better suited as a "glue" language, while Java is better characterized as a low-level implementation language. In fact, the two together make an excellent combination. Components can be developed in Java and combined to form applications in Python; Python can also be used to prototype components until their design can be "hardened" in a Java implementation. To support this type of development, a Python implementation written in Java is under development, which allows calling Python code from Java and vice versa.

**Features:**

1. **Easy to Learn:**

Python is one of the most user-friendly programming languages. One can easily learn the basics of Python and become familiar with its syntax and be able to write basic programs in a few days. However, learning the advanced concepts and mastering python may take you some time. Compared to other languages like C, C++, Java, etc, Python is the easiest language to learn and master.

1. **Easy to Code:**

The syntax of Python is very easy. It generally comprises words from the English language. While writing the code it generally feels like giving the instructions to a child. Apart from this, unlike other programming languages (like C, C++, Java, etc., there is no need to take care of any opening or closing of brackets for defining the scope. In Python, we use indentation (spaces or Tabs) for the scope that makes the code look clean and impressive.

1. **Interpreted language:**

Python code is not compiled at once, converted to a .exe file, and then executed. Python is an interpreted language which means its code is executed line by line and not all at once like in other programming languages. This line-by-line execution also makes it easy to debug the code.

1. **Free and Open Source:**

Python is a free and open-source programming language, which means that it can be used for free of cost on any operating system and without any copyright issues. Anyone can download Python from its official website along with its libraries and documentation. You can download, but it also allows you to make your own modules or libraries and distribute them.

1. **Object-Oriented Language:**

[Object-Oriented Programming](https://www.scaler.com/topics/oops-concepts-in-python/) is a programming paradigm that is based on the concepts of classes and objects. Classes serve as a blueprint for objects, which contains the data and methods that act on that data. The object-oriented programming concept is focused on making reusable code with a good level of abstraction. One of the critical aspects of Python is Object-Oriented programming. Python supports object-oriented programming constructs like classes, data encapsulation, inheritance, polymorphism, etc. In Python, we can easily create and use classes, objects and can implement OOP constructs. Due to this approach, one can build efficient and powerful applications in Python.

1. **Cross-Platform Language:**

Python is a cross-platform language. Many times while downloading some software from some website you might have noticed a list of versions of that software compatible with different operating systems. In Python it’s not the case, once you write a python code on one machine or operating system then it can be run anywhere. For example, if we have made a python program on Mac then we can run the same code in Linux, Windows, or any other operating system without any changes. This is because the python code first is converted to an intermediate form known as Bytecode and is then executed.

1. **Extensive feature:**

Python has the capabilities to be extended and be a more versatile programming language. Python proves to be a versatile language as it covers a large area in software development applications due to its adaptability to various functionalities. We can compile the code in languages like C/C++, and then can use that in our python code which can be compiled and run anywhere. It allows the execution of the code written in other programming languages. This provides Python new capabilities and functionality by integrating other programming language’s code.

1. **High-Level Language:**

Python is a high-level programming language which means that users can easily write and understand or interpret the code. High-level programming language enables the programmer to write codes that are less independent of the specific machine type. Python is a programming language with very strong abstraction from the low-level constructs of the system or machine. While writing code, the developer need not be concerned about the architecture, memory management, or the underlying machine type.

1. **Databases Support:**

Today almost every application we need to develop certainly requires a database and here comes the Python Database API (DB-API) that provides an interface to almost most of the major commercial databases. Some of the databases supported by standard python are MySQL, PostgreSQL, Microsoft SQL, Oracle, Informix, etc. We need to import the interface for the particular database to use it. Using Python you can deal with both relational as well as non-relational databases.

1. **GUI Programming Support:**

When we use our computers or smartphones, How do we interact with them, or what do we see? Obviously, the screen has various icons of applications and when we open some applications. We see a nice visual representation that makes it easy for us to interact and use that application. This is what we call GUI or Graphical User Interface. It is one of the key aspects of the Python Programming language. Python has many GUI libraries like Tkinter, PyQT5, PSide2, etc. GUI provides a user to interact with the application and the system more easily and python GUI libraries have the features that make the development of graphics-intensive software easy and fast.

1. **Large Standard Library:**

Python consists of a bulk of libraries that are cross-platform and provides a rich set of modules and functions. These libraries are compatible with various operating systems like UNIX, Mac, windows, etc. Due to the large number of libraries we need not write code for every single thing instead import and use the functionality required. For example, if you need to access some websites and want to scrape data from them, then you don’t need to write the functions for request, response, and other things, from scratch. There are various libraries available for this purpose which you can use.

We can also install other packages that aren’t a part of the standard library if we need more functionality.

1. **Dynamic Typed Language:**

Python is a Dynamically-Typed language. Dynamically-typed means that, unlike other programming languages, in Python, we need not declare the data type (for example, int, float, double, char, etc.) of a variable explicitly. The data type of the variable is decided at the run time. Apart from this one variable can be used to store different types of data at different instances in the program. This feature of python saves a lot of time and helps us to avoid pitfalls that might have occurred if it required the datatype to be mentioned explicitly.

**Applications**

Python is no doubt a powerful, easy-to-use, and versatile language. It has very simple English-like syntax and has a huge community of awesome developers. Python is a general-purpose programming language that can be used for writing software in various domains.

These features of python have encouraged developers all around the globe to create some helpful libraries and frameworks which extend the application of python in almost every field.

In this article, I will share domains where Python is loved.

**Web Development**

Web development is a huge domain, and Python is one of the most preferred languages.

Python can be used in web development to create the backends for websites.

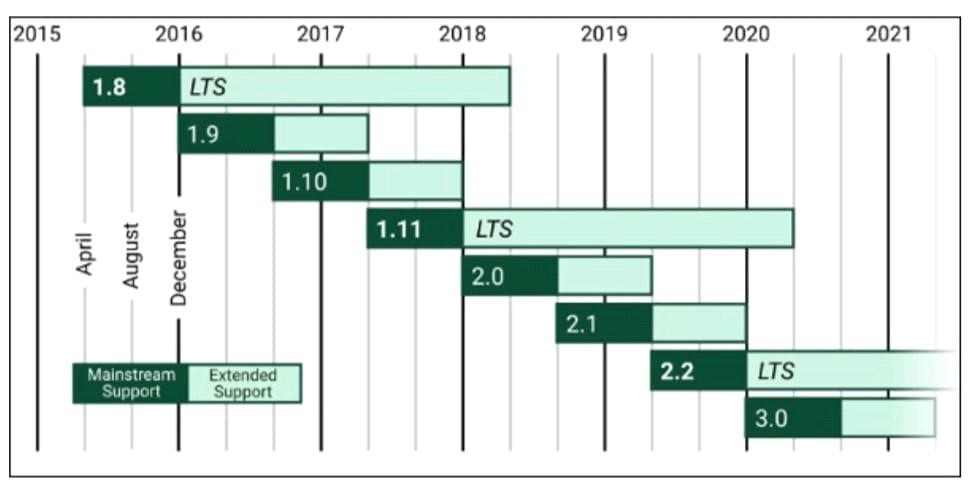
There are two parts of a website:

* Frontend: All the part of the website that is visible to you
* Backend: All the behind the scene work that is performed on your request A simple example to understand this can be Google.

When searching anything on google, all the results we see are called the Frontend, and the back end does the work that Google does to bring those results. You can only appreciate the backend but cannot see it (unless you work at Google, lol)

The most popular frameworks of Python for creating backends are:

* **Flask**
* **Django**



**FIG 3.1 Django Framework**

Both of these allow the creation of a fully functional website. They are easy to learn as they have great documentation.

Many big websites like Instagram, Netflix, and Spotify use Python to develop their product.

**Developing Games**

Python can be used for game development. It can be used to create both 2D and 3D games. One of the most popular modules for 2D game development is PyGame. It is simple, easy to learn, and has a huge community.

We can use it to create games like Flappy Bird, Tic Tac Toe, Minesweeper, etc.

**Creating GUI Apps**

Python can run on almost all platforms, making it a great choice to develop Graphical User Interface (GUI) based applications.

Tkinter, PyQt, and Kivy are some of the most famous libraries for this purpose.

Tkinter comes as an inbuilt module with Python by default. The other two can be installed very easily.

Some famous applications made with Python are Anki and Dropbox.

**Performing Complex Calculations**

This is yet another field where simple syntax and powerful libraries help.

There is a need for many complex calculations that need to be performed quickly in the research field. Many mathematicians and computer scientists have created libraries like SciPy and Numpy that enable us to do complex mathematical calculations very quickly.

Some of the most famous libraries used for performing complex mathematical calculations are

* **SciPy** is used for scientific computation
* **Numpy** supports the use of large multidimensional arrays and matrices
* **Pandas** is the most famous data analytics library

**Artificial Intelligence and Machine Learning**

AI and ML are buzzwords nowadays with huge potential, and many people are getting into these, and surely [Python](https://www.scaler.com/topics/python/) does not want to lag.

This field includes the use of libraries from different domains because, in AI and ML, there are many tasks like data collection, data analysis, and advanced scientific calculation.

Some most commonly used libraries are:

* **TensorFlow** by Google
* **Pytorch** by Facebook
* **Keras**
* **Scikit** **Learn**

These libraries are extremely powerful, and many big companies use them.

You can easily start learning these by following the official documentation by visiting their website.

**Software Development**

Python has many packages which aim to simplify some task or the other. Python is used for development and as a support language for management and testing.

The most popular tools and libraries for this purpose are SCons, Buildbot, Trac, etc.

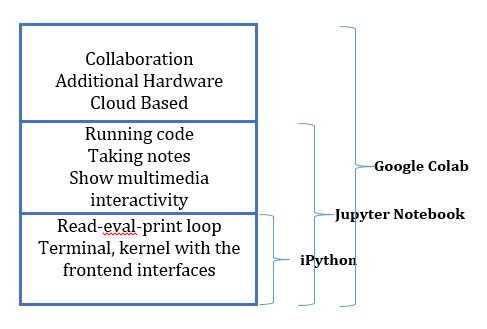
**3.2 DEVELOPMENT TOOLS:**

**COLLAB:**

Colabratory, or “Colab” for short, is a product from Google Reasearch. Colab allows anybody to write and execute arbitary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebbok service that requires no setup to use, while providing access free of charge to computing resources including GPUs. Resources in Colab are prioritized for interactive use cases. We prohibit actions associated with bulk compute, actions that negatively impact others, as well as actions associated with bypassing our policies.

As a developer, you can perform the following using Google Colab;

* Write and execute code in Python
* Create/Upload/Share notebooks
* Import/Save notebooks from/to Google Drive
* Import/Publish notebooks from GitHub
* Import external datasets
* Integrate PyTorch, TensorFlow, Keras, OpenCV
* Free Cloud service with free GPU



**FIG 3.2 Development tools**

**Reasons:**

* Pre-installed Libraries
* Saved on the Cloud
* Collaboration
* Free GPU and TPU Use

**Pre-Installed Libraries**

Anaconda distribution of Jupyter Notebook is shipped with several pre-installed data libraries, such as NumPy, Pandas, Matplotlib. Colab, on the other hand, provides more preinstalled machine learning libraries including Keras, PyTorch, TensorFlow.

**Saved on the Cloud**

When you opt to use a plain Jupyter notebook as your development environment, everything is saved in your local machine. If you are cautious about privacy, this may be a preferred feature for you. However, if you want your notebooks to be accessible to you from any device with a simple Google log-in, then Google Colab is the way to go. All of your Google Colab notebooks are saved under your [Google Drive account,](https://mjgarbade.com/) just like your Google Docs and Google Sheets files.

**Collaboration**

The other great feature that Google Colab offers is the collaboration feature. If you are working with multiple developers on a project, it is great to use Google Colab notebook.

Just like collaborating on a Google Docs document, you can co-code with multiple developers using a Google Colab notebook. Besides, you can also share your completed work with other developers.

**Free GPU and TPU Use**

It is an absolute no brainer to choose Google Colab instead of a local Jupyter notebook. Google Research lets you use their dedicated GPUs and TPUs for your personal machine learning projects. For some projects, the GPU and TPU acceleration make a huge difference even for some small projects. Besides, since it uses Google resources, the neural network optimization operations do not mess with your processors, and your cooling fan will not go crazy.

Google Colab is a specialized version of the Jupyter Notebook. It runs on the cloud and offers free computing resources. The relationship between iPython, Jupyter Notebook, and Google Colab is shown below;

**CHAPTER 4**

### SYSTEM DESIGN AND UML DIAGRAMS

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

**4.1 SOFTWARE DESIGN:**

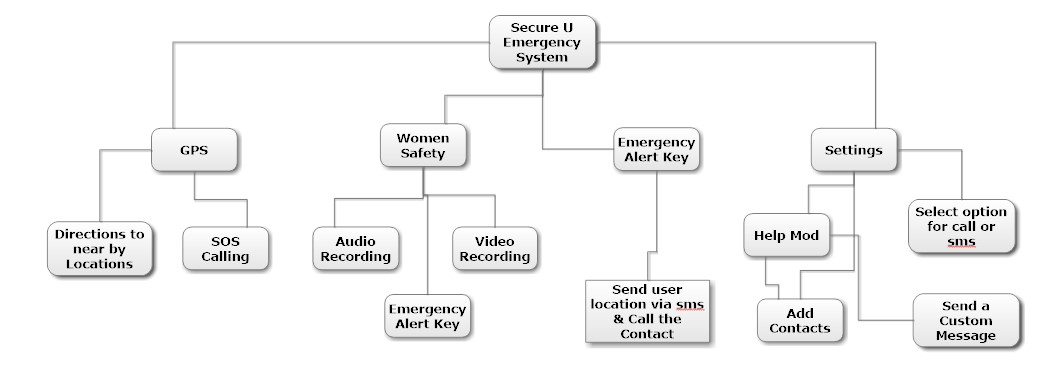
In designing the software following principles are followed:

* **Modularity and partitioning**: Software is designed such that, each system should consist of hierarchy of modules and serve to partition into separate function.
* **Coupling:** Modules should have little dependence on other modules of a system.
* **Cohesion:** Modules should carry out in a single processing function.
* **Shared use:** Avoid duplication by allowing a single module be called by other that need the function it provides.

**4.2 ARCHITECTURE:**

Architecture diagram is a [diagram](http://en.wikipedia.org/wiki/Diagram) of a [system,](http://en.wikipedia.org/wiki/System) in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.

A SMS user for who the application looks like an user interface actually consists of a database called as SQLite that comes along with Android SDK and need no other installation. This is the database that is used to store and retrieve information. This is an application that is developed in java and hence all its features apply here as well such as platform independence, data hiding.



**FIGURE 4.1: SYSTEM FLOW DIAGRAM**

**4.3 UNIFIED MODELING LANGUAGE (UML):**

The unified modeling is a standard language for specifying, visualizing, constructing and documenting the system and its components is a graphical language which provides a vocabulary and set of semantics and rules. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. It is used to understand, design, configure and control information about the systems.

Depending on the development culture, some of these artifacts are treated more or less formally than others. Such artifacts are not only the deliverables of a project; they are also critical in controlling, measuring, and communicating about a system during its development and after its deployment.

The UML addresses the documentation of a system's architecture and all of its details. The UML also provides a language for expressing requirements and for tests. Finally, the UML provides a language for modeling the activities of project planning and release management.

**4.3.1 BUILDING BLOCKS OF UML:**

The vocabulary of the UML encompasses three kinds of building blocks:

* Things.
* Relationships.
* Diagrams.

**4.3.1.1 Things in the UML:**

Things are the abstractions that are first-class citizens in a model; relationships tie these things together; diagrams group interesting collections of things.

There are four kinds of things in the UML:

* Structural things.
* Behavioral things.
* Grouping things.
* Annotational things.
* **Structural things** are the nouns of UML models. The structural things used in the project design are:
* First, a **class** is a description of a set of objects that share the same attributes, operations, relationships and semantics.

|  |
| --- |
| Window |
| origin size |
| open() close() move() display() |

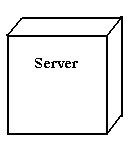
**Fig: Classes**

* Second, a **use case** is a description of set of sequence of actions that a system performs that yields an observable result of value to particular actor.



**Fig: Use Cases**

* Third, a node is a physical element that exists at runtime and represents a computational resource, generally having at least some memory and often processing capability.



**Fig: Nodes**

* **Behavioral things** are the dynamic parts of UML models. The behavioral thing used is:
* Interaction: An interaction is a behavior that comprises a set of messages exchanged among a set of objects within a particular context to accomplish a specific purpose. An interaction involves a number of other elements, including messages, action sequences (the behavior invoked by a message, and links (the connection between objects).



**Fig: Messages**

**4.3.1.2 Relationships in the UML:**

There are four kinds of relationships in the UML:

* Dependency.
* Association.
* Generalization.
* Realization.

* A **dependency** is a semantic relationship between two things in which a change to one thing may affect the semantics of the other thing (the dependent thing).



**Fig: Dependencies**

* An **association** is a structural relationship that describes a set links, a link being a connection among objects. Aggregation is a special kind of association, representing a structural relationship between a whole and its parts.



**Fig: Association**

* A **generalization** is a specialization/ generalization relationship in which objects of thespecialized element (the child) are substitutable for objects of the generalized element (the parent).



**Fig: Generalization**

* A **realization** is a semantic relationship between classifiers, where in one classifier specifies a contract that another classifier guarantees to carry out.

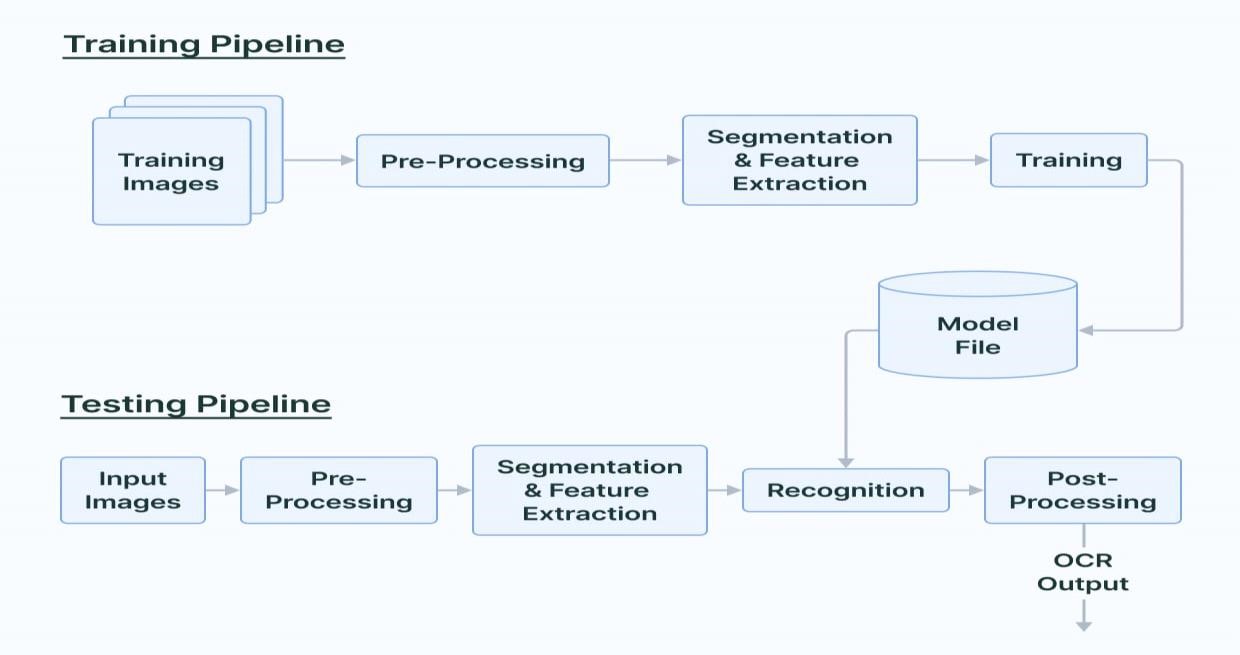


**Fig: Realization**

**4.3.2 UML DIAGRAMS:**

* + - 1. **DATA FLOW DIAGRAM:**

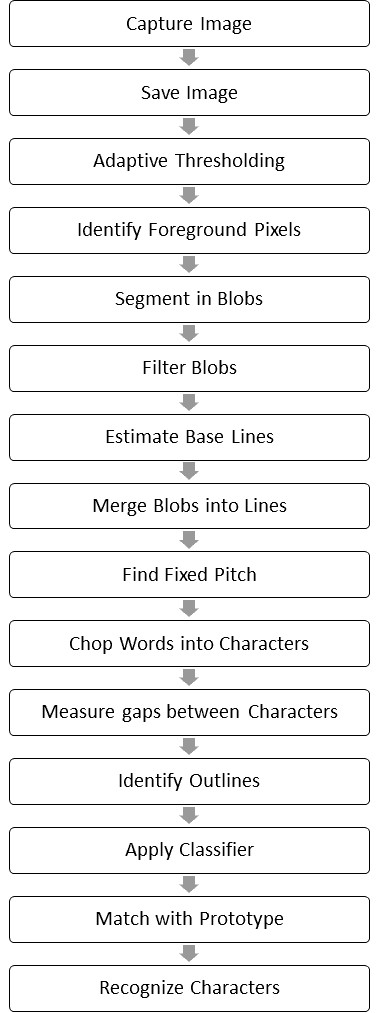
Data flow describes the information transferring between different parts of the systems. The arrow symbol is the symbol of data flow. A relatable name should be given to the flow to determine the information which is being moved. Data flow also represents material along with information that is being moved. Material shifts are modeled in systems that are not merely informative. A given flow should only transfer a single type of information. The direction of flow is represented by the arrow which can also be bidirectional. Data Flow diagrams are very popular because they help us to visualize the major steps and data involved in software-system processes.



**FIG 4.2 DATAFLOW DIAGRAM**

* + - 1. **FLOWCHART DIAGRAM:**

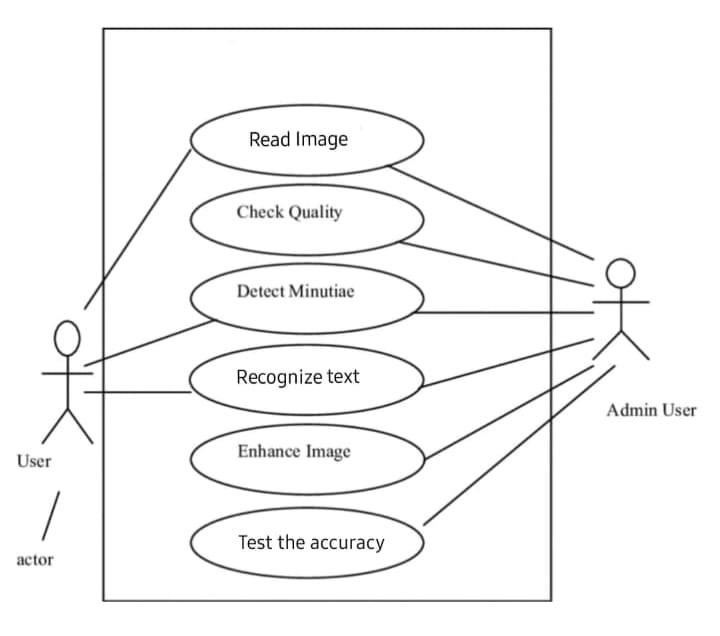
Flowchart is a graphical representation of an algorithm. Programmers often use it as a program-planning tool to solve a problem. It makes use of symbols which are connected among them to indicate the flow of information and processing. Flow charts are a better way of communicating the logic of the system. Flowcharts act as a guide for blueprint during program designed. Flowcharts help in debugging process.



**FIG 4.3 SEQUENCE DIAGRAM**

* + - 1. **USE CASE DIAGRAM:**

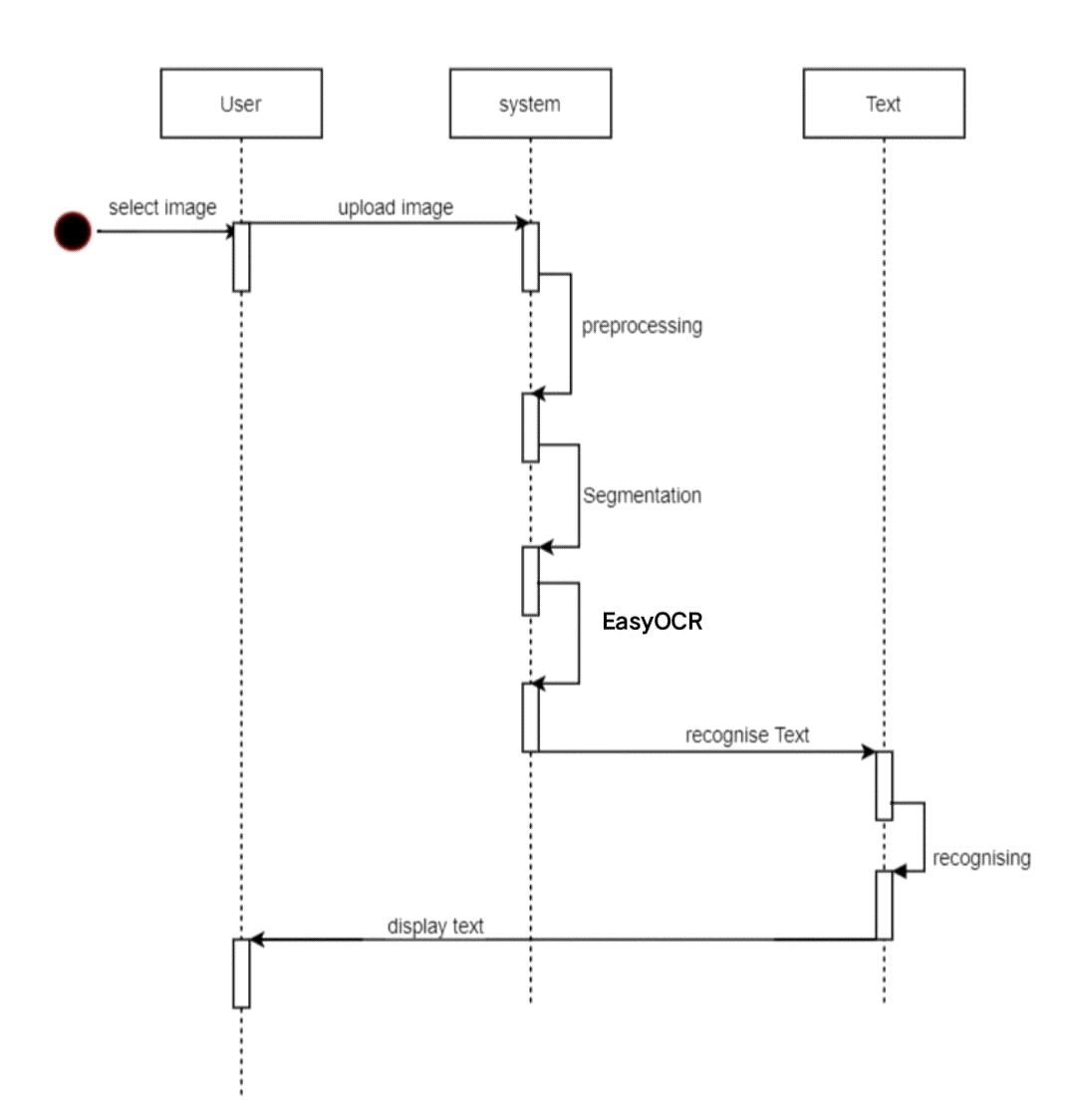
In the Unified Modeling Language (UML), a use case diagram can summarize the details of your system's users (also known as actors) and their interactions with the system. To build one, you'll use a set of specialized symbols and connectors. UML use case diagrams are ideal for representing the goals of system-user interactions, defining and organizing functional requirements in a system, specifying the context and requirements of a system, modeling the basic flow of events in a use case.



**FIG 4.4 USE CASE DIAGRAM**

* + - 1. **SEQUENCE DIAGRAM:**

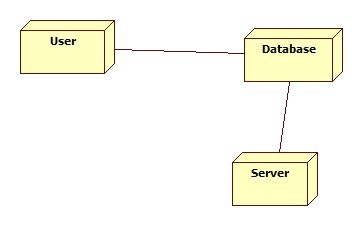
A sequence diagram or system sequence diagram (SSD) shows [process](https://en.wikipedia.org/wiki/Process_(computing)) interactions arranged in time sequence in the field of [software engineering.](https://en.wikipedia.org/wiki/Software_engineering) It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. Sequence diagrams are typically associated with use case realizations in the [architectural view model](https://en.wikipedia.org/wiki/4%2B1_architectural_view_model) of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.



**FIG 4.5 SEQUENCE DIAGRAM**

**4.3.2.5 DEPLOYMENT DIAGRAM:**

**Deployment diagrams** are used to visualize the topology of the physical components of a system where the software components are deployed. So deployment diagrams are used to describe the static deployment view of a system. Deployment diagrams consist of nodes and their relationships.

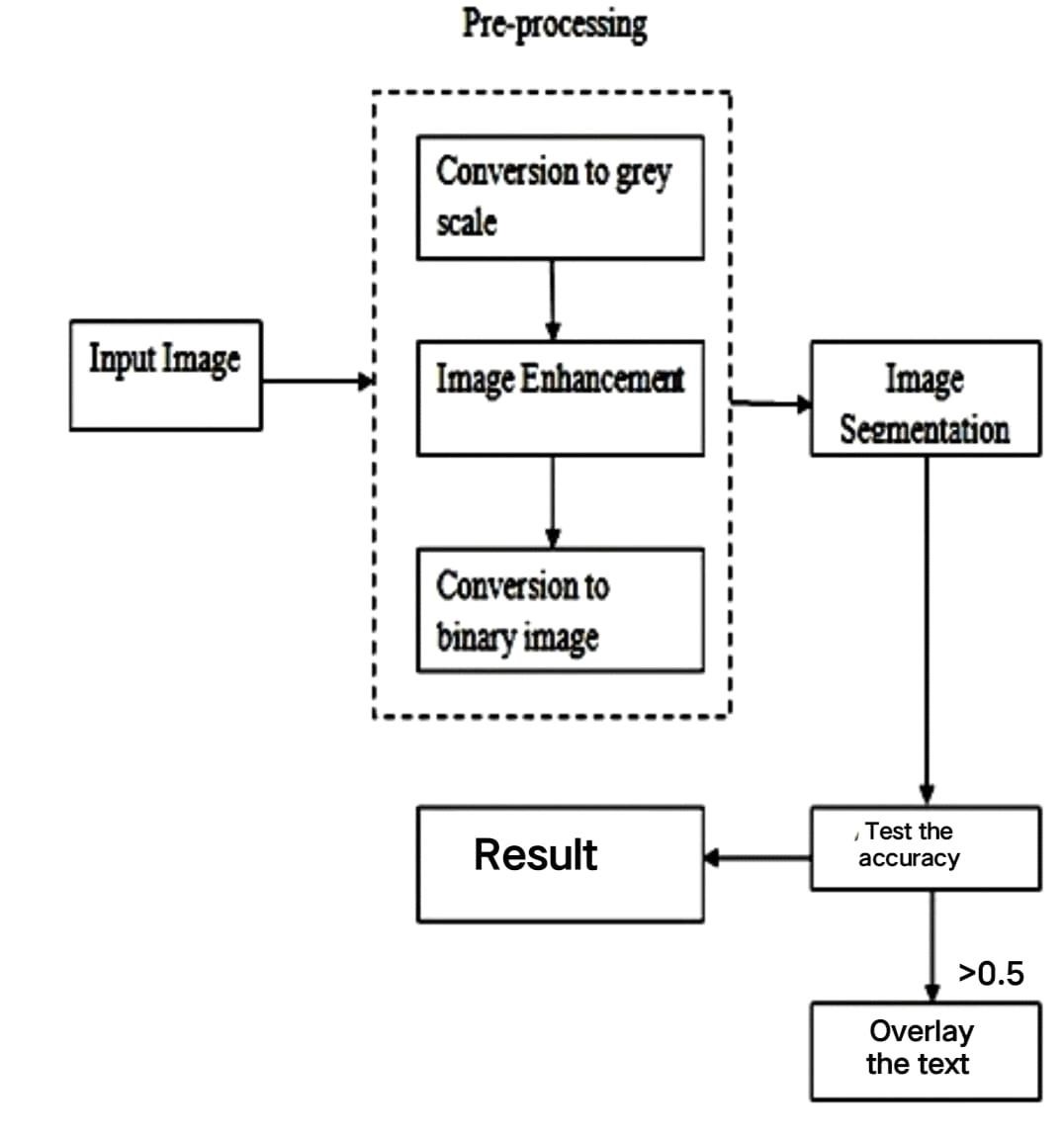


**FIG 4.6 DEPLOYMENT DIAGRAM**

**4.3.2.6 BLOCK DIAGRAM:**

A **block diagram** is a drawing illustration of a system whose major parts or components are represented by blocks. These blocks are joined by lines to display the relationship between subsequent blocks.

We use block diagrams to visualize the functional view of a system. It uses blocks connected with lines to represent components of a system. With a block diagram, you can easily illustrate the essential parts of a software design or engineering system and depict the data flow in a process flow chart. You can learn everything about a block diagram, including its uses and types. Check out some of its most used examples and learn how to create your block diagram.



**FIG 4.7 BLOCK DIAGRAM**

### CHAPTER 5 INPUT/OUTPUT DESIGN

**5.1 INPUT DESIGN:**

Input Design plays a vital role in the life cycle of software development, it requires very careful attention of developers. The input design is to feed data to the application as accurate as possible. So inputs are supposed to be designed effectively so that the errors occurring while feeding are minimized. According to Software Engineering Concepts, the input forms or screens are designed to provide to have a validation control over the input limit, range and other related validations. Input design is the process of converting the user created input into a computer-based format. The goal of the input design is to make the data entry logical and free from errors.

Validations are required for each data entered. Whenever a user enters an erroneous data, error message is displayed and the user can move on to the subsequent pages after completing all the entries in the current page.

**5.2 OUTPUT DESIGN:**

The Output from the computer is required to mainly create an efficient method of communication within the company primarily among the project leader and his team members, in other words, the administrator and the clients. The output of VPN is the system which allows the project leader to manage his clients in terms of creating new clients and assigning new projects to them, maintaining a record of the project validity and providing folder level access to each client on the user side depending on the projects allotted to him. After completion of a project, a new project may be assigned to the client.

User authentication procedures are maintained at the initial stages itself.

**CHAPTER 6**

### IMPLEMENTATION

**Step 1: Install and import modules**

!pip install opencv-python

!pip install matplotlib

!pip install numpy

!pip install torch==1.7.1+cpu torchvision==0.8.2+cpu torchaudio===0.7.2 f<https://download.pytorch.org/whl/torch_stable.html>

!pip install easyocr

The OpenCV module is for computer vision-related operations in Python. Specifically, we will use it to overlay images with respective recognized texts later. We need the Matplotlib module to display images. And we will use the Numpy module to convert images into arrays.

PyTorch is a prerequisite for the EasyOCR module. Its installation varies according to OS and GPU driver requirements. You can get the installation commands at the PyTorch homepage. Copy and execute the respective command as shown in Figure if you operate on Windows.

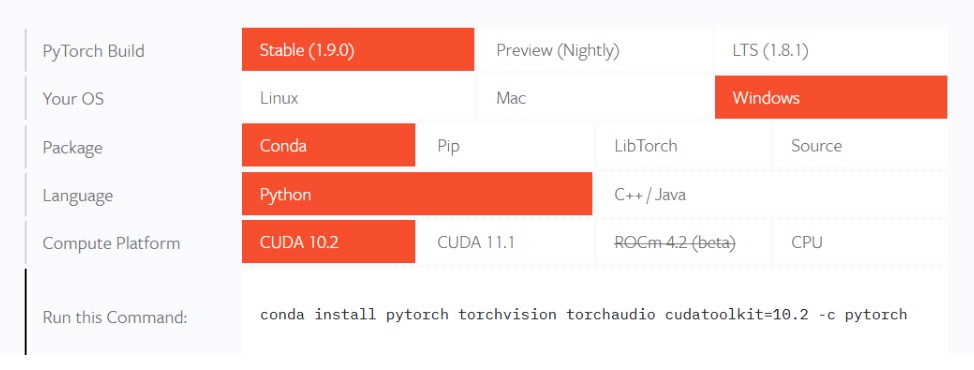


FIG 6.1 Installation Command for the PyTorch

install the EasyOCR module — the tool we need for extracting text from images.

import cv2

import numpy as np import easyocr import matplotlib.pyplot as plt

%matplotlib inline

The %matplotlib inline magic command is exclusive for Jupyter notebooks. It is not required in a Python script. It sets the backend of the Matplotlib module to display figures inline and not on a separate window.

**Image loading and text recognition using EasyOCR** from google.colab import drive drive.mount('/content/drive')

Copy the path of the images im\_1\_path = '/content/drive/MyDrive/p1.jpg' im\_2\_path = '/content/drive/MyDrive/p2.jpg' im\_3\_path = '/content/drive/MyDrive/p3.jpg' im\_4\_path = '/content/drive/MyDrive/p4.jpg'

def recognize\_text(img\_path):

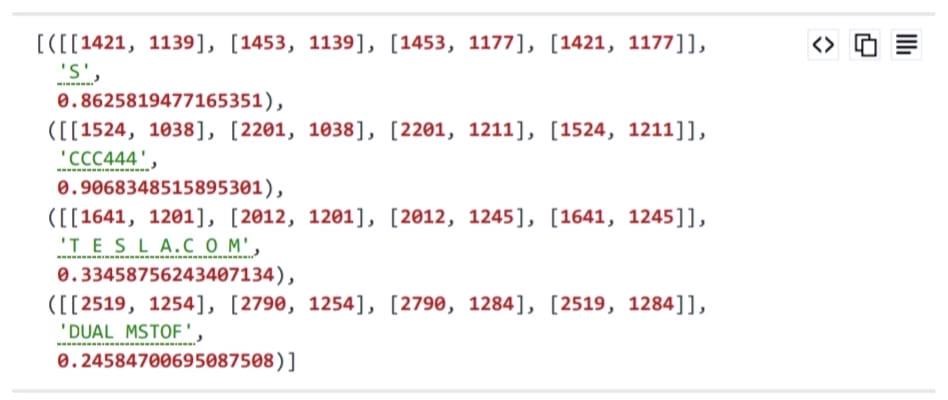
'''loads an image and recognizes text.'''

reader = easyocr.Reader(['en']) return reader.readtext(img\_path)

The recognize\_text() function initializes an OCR reader to a variable named reader. It takes a list of languages as a parameter. We want to only recognize English text, thus the ‘en’ in the list. The readtext method reads an image given its stored directory. The returned OCR result is passed as the output of the recognize\_text() function.

result = recognize\_text(im\_1\_path) result

It it is going to take more time to execute EasyOCR on a CPU instead of a GPU. The im\_1\_path image took around ten seconds to be executed by recognize\_text().



img\_1 = cv2.imread(im\_1\_path) img\_1 = cv2.cvtColor(img\_1, cv2.COLOR\_BGR2RGB) plt.imshow(img\_1)



Overlay recognized text on image using OpenCV def overlay\_ocr\_text(img\_path, save\_name):

'''loads an image, recognizes text, and overlays the text on the image.'''

# loads image img = cv2.imread(img\_path) img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

dpi = 80 fig\_width, fig\_height = int(img.shape[0]/dpi), int(img.shape[1]/dpi) plt.figure() f, axarr = plt.subplots(1,2, figsize=(fig\_width, fig\_height)) axarr[0].imshow(img)

The OpenCV module to load an image as a Numpy array and correct its color channels. The array is assigned to the variable img. Display two images — the original image, and the original image with recognized texts. The subplots method of Matplotlib is used to display more than one figure at a time. The imshow method of the axarr[0] variable displays the original image.

# recognize text result = recognize\_text(img\_path)

# if OCR prob is over 0.5, overlay bounding box and text for (bbox, text, prob) in result: if prob >= 0.5: # display print(f'Detected text: {text} (Probability: {prob:.2f})')

# get top-left and bottom-right bbox vertices

(top\_left, top\_right, bottom\_right, bottom\_left) = bbox top\_left = (int(top\_left[0]), int(top\_left[1]))

bottom\_right = (int(bottom\_right[0]), int(bottom\_right[1]))

# create a rectangle for bbox display cv2.rectangle(img=img, pt1=top\_left, pt2=bottom\_right, color=(255, 0, 0), thicknes s=10)

# put recognized text

cv2.putText(img=img, text=text, org=(top\_left[0], top\_left[1] -

10), fontFace=cv2.FONT\_HERSHEY\_SIMPLEX, fontScale=1, color=(255, 0, 0), thickne ss=8)

The recognize\_text() function returns the OCR output and assigns it to the result variable. A for loop is created to go through each text element contained in the variable. Recognized text elements are displayed only if their OCR confidence levels are higher than 0.5 (prob >= 0.5). Then, the top left and bottom right vertices of each bounding box are obtained. They are converted into tuples of integer values (as required by OpenCV). The rectangle method creates a green bounding box for each detected text element. The putText method displays recognized text above its respective bounding box. As all these are done in a for loop, the operation repeats for every recognized text in the result variable.

# show and save image axarr[1].imshow(img) plt.savefig(f'./output/{save\_name}\_overlay.jpg', bbox\_inches='tight')

the overlay\_ocr\_text() function displays every created text and bounding box. The imshow method of the axarr[1] variable displays the final image. As both the left and right images are in the same subplot, they are displayed as one final image. The savefig method stores the final image to a defined local directory.

overlay\_ocr\_text(im\_1\_path, '1\_carplate') overlay\_ocr\_text(im\_2\_path, '2\_handwriting') overlay\_ocr\_text(im\_3\_path, '3\_digits') overlay\_ocr\_text(im\_4\_path, '4\_invoice') overlay\_ocr\_text(im\_5\_path, '5\_roadsign')

**CHAPTER 7**

### TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

**7.1 TESTING OBJECTIVES:**

* To ensure that during operation the system will perform as per specification.
* To make sure that system meets the user requirements during operation
* To make sure that during the operation, incorrect input, processing and output will be detected
* To see that when correct inputs are fed to the system the outputs are correct
* To verify that the controls incorporated in the same system as intended
* Testing is a process of executing a program with the intent of finding an error
* A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

**7.2 TESTING METHODOLOGIES:**

* White box testing.
* Black box testing.
* Unit testing.
* Integration testing.
* User acceptance testing.
* Output testing.
* Validation testing.
* System testing.
* **White Box Testing:**

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu.

After which the control exits the current menu.

* **Black Box Testing:**

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

* **Unit Testing:**

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module’s control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing

* **Integration Testing:**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

**The following are the types of Integration Testing:**

* **Top Down Integration:**

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module.

* **Bottom Up Integration:**

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.

* **User acceptance Testing:**

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

* **Output Testing:**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

* **Validation Testing:**

Validation testing is generally performed on the following fields:

* **Text Field:**

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables.

Incorrect entry always flashes and error message.

* **Numeric Field:**

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error messages. The individual modules are checked for accuracy and what it has to perform.

* **Preparation of Test Data:**

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

* **Using Live Test Data:**

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

* **Using Artificial Test Data:**

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications.

* 1. **USER TRAINING:**

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

* 1. **MAINTAINENCE:**

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user’s requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

* 1. **TESTING STRATEGY:**

A strategy for system testing integrates system test cases and design techniques into a well planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding.

* + 1. **SYSTEM TESTING:**

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

* + 1. **UNIT TESTING:**

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module. In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.

### CHAPTER 8 OUTPUT SCREENS

overlay\_ocr\_text(im\_2\_path, '2\_handwriting')

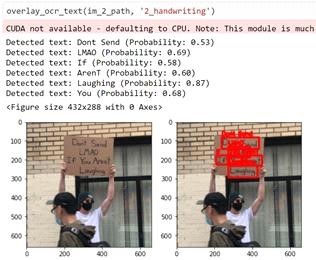


FIG 8.1 Hand writing

overlay\_ocr\_text(im\_2\_path, '1\_carplate')



FIG 8.2 Car number plate

overlay\_ocr\_text(im\_3\_path, '3\_digits')

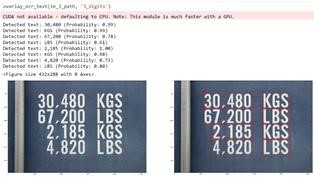


FIG 8.3 Digits

overlay\_ocr\_text(im\_5\_path, '5\_roadsign')



FIG 8.4 Roadsign

**CHAPTER 9**

### CONCLUSION AND FUTURE SCOPES

**CONCLUSION**

This project explains how to extract text elements from images using EasyOCR. It also shows how to overlay recognized text on images using OpenCV. A simple text-to-speech is also introduced as an extended application for OCR output. EasyOCR is the simplest way of applying Optical Character Recognition and it's the most accurate by far. For most tested images and extended over a wide range of languages. It is easy to use and requires only a few lines of code to implement, and has proper accuracy. Automatic text detection and extraction from an image is an important research branch of content-based information retrieval and text based image indexing. Some of the applications fields of text extraction are mobile robot navigation, vehicle license detection and recognition, object identification, document retrieving, page segmentation etc.

**FUTURE SCOPE**

In future we will extract the text from videos and reshaping the scanning documents and character recognition as much as any other sector. We will try to improve the accuracy near to 99% in typewritten texts.

**Cursive Character:**

There is heavy demand for an EasyOCR system which recognizes handwritten cursive scripts. This avoids keyboard typing and font coding for the images. This method helps in detecting handwritten characters with a precision of about 90%.

**Speech to text converter through EasyOCR:**

Speech recognition is one of the most important application today. The recognized speech could be recorded and output of text could be generated.

### BIBLIOGRAPHY

[https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocrhands-on-guide/](https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocr-hands-on-guide/) <https://nanonets.com/blog/deep-learning-ocr/><https://serengetitech.com/tech/deep-learning-ocr-text-detection-and-text-recognition/>[https://www.etutorialspoint.com/index.php/381-text-extraction-from-image-using-opencvand-ocr-python](https://www.etutorialspoint.com/index.php/381-text-extraction-from-image-using-opencv-and-ocr-python) [https://analyticsindiamag.com/hands-on-tutorial-on-easyocr-for-scene-text-detection-inimages/](https://analyticsindiamag.com/hands-on-tutorial-on-easyocr-for-scene-text-detection-in-images/) [https://blog.finxter.com/how-to-extract-text-from-images-ocr-in-python-using-opencv-andeasyocr/](https://blog.finxter.com/how-to-extract-text-from-images-ocr-in-python-using-opencv-and-easyocr/) [https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-characterrecognition/](https://pyimagesearch.com/2020/09/14/getting-started-with-easyocr-for-optical-character-recognition/) <https://mlhive.com/2022/04/optical-character-recognition-using-easyocr-in-python><https://pianalytix.com/using-easyocr-library-for-text-extraction/>[https://www.opcito.com/blogs/extracting-text-from-images-with-tesseract-ocr-opencv-andpython](https://www.opcito.com/blogs/extracting-text-from-images-with-tesseract-ocr-opencv-and-python)