University of Mumbai

Text Recognition in Natural Images

Submitted at the end of semester VI in partial fulfillment of requirements For the degree of

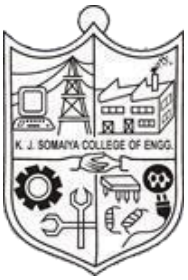
Bachelor of Technology

by

Roll No: 1713026 Jaini Gala
Roll No: 1713033 Jil Kothari
Roll No: 1713035 Fenil Madlani
Roll No: 1713041 Manav Parekh

Guide

Ms. JAYASHREE M. KUNDARGI



Department of Electronics and Telecommunication Engineering
K. J. Somaiya College of Engineering, Mumbai-77 (Autonomous College Affiliated to University of Mumbai)

Batch 2017 -2021

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate

This is to certify that the dissertation report entitled **Text Recognition in Natural Images** submitted at the end of semester VI of TY B. Tech is a bonafide record for partial fulfillment of requirements for the degree of Bachelors in Technology in

Electronics and Telecommunication Engineering of University of Mumbai

_____	_____
Guide/Examiner1	Expert/Examiner2

Date:13/06/2020

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

Certificate of Approval of Examiners

We certify that this dissertation report entitled **Implementation of Text Recognition in Natural Images** is bonafide record of project work done by Jaini Gala, Jil Kothari, Fenil Madlani and Manav Parekh during semester VI.

This project work is submitted at the end of semester VI in partial fulfillment of requirements for the degree of Bachelors in Technology in Electronics and Telecommunication Engineering of University of Mumbai.

_____

Internal Examiners

_____

External/Internal Examiners

Date:13/06/2020

Place: Mumbai-77

K. J. Somaiya College of Engineering, Mumbai-77

(Autonomous College Affiliated to University of Mumbai)

DECLARATION

We declare that this written report submission represents the work done based on our and / or others' ideas with adequately cited and referenced the original source. We also declare that we have adhered to all principles of intellectual property, academic honesty and integrity as we have not misinterpreted or fabricated or falsified any idea/data/fact/source/original work/ matter in my submission.

We understand that any violation of the above will be cause for disciplinary action by the college and may evoke the penal action from the sources which have not been properly cited or from whom proper permission is not sought.

Jaini Gala **1713026** **Roll No.**	**Jil Kothari** **1713033** **Roll No.**
Fenil Madlani **1713035** **Roll No.**	**Manav Parekh** **1713041** **Roll No.**

Date:13/06/2020

Place: Mumbai-77

Abstract

Text detection is the method of locating areas in a picture wherever, text is present. Text detection and classification in natural pictures is very important for several computer vision applications like optical character recognition, distinguish between human and machine inputs and spam removal. Currently the challenge in text identifying is to detect the text in natural pictures due to many factors like, low-quality image, unclear words, typical font, image having a lot of color stroke than the background color, blurred pictures due to some natural problems like rain, sunny, snow, etc. The main aim of this work is to identify and classify the text in natural pictures. Here system detects the text and finds the connected regions, chain them together in their relative position. Uses a text classification engine (OCR) to filter chains with low classification confidence scores

***Key words*: Text detection, Optical Character Recognition (OCR), Tesseract, Segmentation, Processing, Deep Learning**

Contents

List of Figures

Chapter 1

Introduction

This chapter presents a brief idea about this project where the aim is to provide a quick understanding of the various concepts used in the recognition of text in natural images. It also gives a succinct account about the motivation behind this project and its scope.

1.1 Background

Text recognition in images is an active research area which attempts to develop a computer application with the ability to automatically read the text from images. Humans can understand the contents of an image simply by looking. We perceive the text on the image as text and can read it but computers don't work in the same way. They need something more concrete, organized in a way that they can understand. This is where Optical Character Recognition (OCR) kicks in and it can prove to be a very useful technique. While it's not always perfect, it's very convenient and it enables people to do their jobs faster and more efficiently.

1.2 Motivation

Nowadays, there is a huge demand of storing the information available on digital documents such as images in the form of text for later use. However, this is not quite as simple to execute due to the distortions in the font characteristics of the characters and the poor quality of the image. Also, the text is usually printed against shaded or textured backgrounds or is embedded in the image which often interferes with the results. This is why we decided to try our best and implement a model with a relatively high accuracy of recognizing text from the images which can then be applied to many realtime applications and make the job effortless. This is what motivated us.

1.3 Scope of the project

We have successfully implemented the code and achieved desired results with an accuracy of more than 80%. The examples given in this report have a 100% accuracy rate.

1.4 Brief description of project undertaken :

Text recognition is an interesting and highly researched field of computer vision. It encourages the applications of automation and hence reducing human efforts. The accuracy of OCR predominantly depends on image preprocessing and quality of the image. Moreover, factors such as font size, geometric distortions and illumination effects are critical in recognizing the required text accurately. This project aims at implementing a model which recognizes the text from the images with the highest possible accuracy by using the concepts of Deep Learning and Optical Character Recognition.

Chapter 2

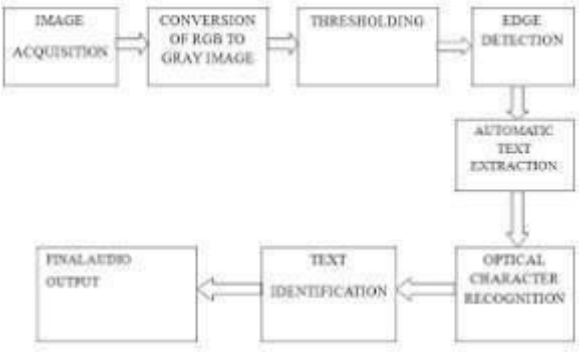
Literature Survey

Various Methods of OCR –

1) Hybrid method - Hybrid method is used for text classification. This approach detects and recognizes texts in CAPTCHA images. The strength of CAPTCHA can be checked. This method efficiently detects and recognizes the text with a low false positive.

Example of Hybrid method of OCR

2) Edge based method - This method is also known as image processing technique, which finds boundaries of the images or any other objects within the images. It works by detecting discontinuities in brightness. This approach is also used for image segmentation and data extraction in areas such as image processing, machine vision and computer vision.



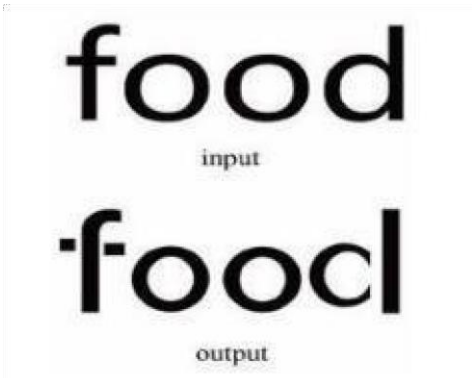
Flow chart of Edge based method

3) Color based method - Color based approach is used for clustering. It consists of two phases: text detection phase and text extraction phase. In text detection phase, two features are considered – homogeneous color and sharp edges, and color based clustering is used to decompose color edge map of image to several edge maps, which makes text detection more accurate. In text extraction phase, the difference between the text and background in image is considered.

4) Texture based method - It is another approach used for detecting texts in images. This approach uses Support Vector Machine (SVM) to analyze the textural properties of texts. This method also uses continuously adaptive mean shift algorithm (CAMSHIFT) that results in texture analysis. It combines both SVM and CAMSHIFT to provide robust and efficient text detection.

5) Corner based method - Corner based method is used for text extraction method. It has three stages - a) Computing corner response in multiscale space and thresholding it to get the candidate region of text; b)verifying candidate region by combining color and size range features and; c) locating the text line using bounding box. It is two- dimensional feature point which has high curvature in region boundary.

6) Stroke based method - This approach is used to detect and recognize text from the video. It uses text confidence using an edge orientation variance and opposite edge pair feature. The components are extracted and grouped into text lines based on text confidence maps. It can detect multilingual texts in video with high accuracy.



Example of stroke based method

LITERATURE REVIEW:

Text detection from images is useful in many real world applications. The data that is stored in text is huge and there is need to store this data in such a manner that can be searched easily whenever required. Elimination of the use of paper is one of the steps to progress towards a world of electronics. Also, data that can be converted to audio form is a way to ease the lives of visually impaired people.

In [2007] Ray Smith published an overview of Tesseract OCR Engine. It stated that Tesseract started as a PhD project sponsored by HP in 1984. In 1987, a second person was assigned to help build it better. In 1988, HPLabs joint with Scanner Division project. In 1990, the scanner product was cancelled and four years ahead HPLabs project was cancelled too. From the year 1995 till the year 2005, Tesseract was in its dark ages. But in the year 2005, it was open sourced by HP. In 2006, Google took over it. In 2008, Tesseract expanded to support six languages. By the year 2016, it was developed further to makes use of LSTM for the purpose of OCR.

In [2015] Pratik MadhukarManwatkar and Dr. Kavita R. Singh published a technical review on text recognition from images. It emphasizes on the growing demand of OCR applications as it necessary in today's world to store information digitally so that it can be edited whenever required. This information can later be searched easily as it is in digital format. The system takes image as an input, processes on the image and the output is in the form of textual data. In [2016] Akhilesh A. Panchal, ShrugalVarde and M.S. Panse proposed a character detection and recognition system for visually impaired people. They focused on the need of people that are visually impaired as it is difficult for them to read text data. This system can be used to extract text data from shop boards or direction boards and convey this information to the user in audio form. The main challenges are the different fonts of the texts on the natural scene images.

In [2017] Nada Farhani, NaimTerbeh and MounirZrigui published a paper that stated the conversion of different modalities. Human beings have different modalities such as gesture, sound, touch and images. It is vital that they can convert the information between these modalities. The paper focuses on conversion of image to text and also on text-tospeech so that the user can hear the information whenever required. Azmi Can Özgen, Mandana Fasounaki and Hazım Kemal Ekenel published a paper that stated how text data can be extracted from both natural scene images and computer generated images. They make use of Maximally Stable Extermal Regions for the purpose of text detection and

recognition. This method eliminated the non-text part of the image so that OCR can be done more efficiantly.

In [2018] Sandeep Musale and VikramGhiye proposed a system that is a smart reader for visually impaired people. Using this system, they can convert the text information to audio format. This system has an audio interface that the people with visual problems can use easily. It uses a combination of OTSU and Canny algorithms for the purpose for character recognition. Christian Reul, UweSpringmann, Christoph Wick and Frank Puppe proposed a method to reduce the errors generated during OCR process. It includes cross fold training and voting to recognize the words more accurately. As LSTM is introduced now, it is easier to recognize words of old printed books, handwritten words, blurry or uneven words with high accuracy. A combination of ground truths and confidence values are used in this method for optimal recognition of the characters.

Chapter 3

Project design

This chapter presents the main objective of this project and brief description of the tools used in the designing of this project.

3.1 Introduction

OCR = Optical Character Recognition. In other words, OCR systems transform a two-dimensional image of text, that could contain machine printed or handwritten text from its image representation into machine-readable text. OCR as a process generally consists of several sub-processes to perform as accurately as possible. The subprocesses are:

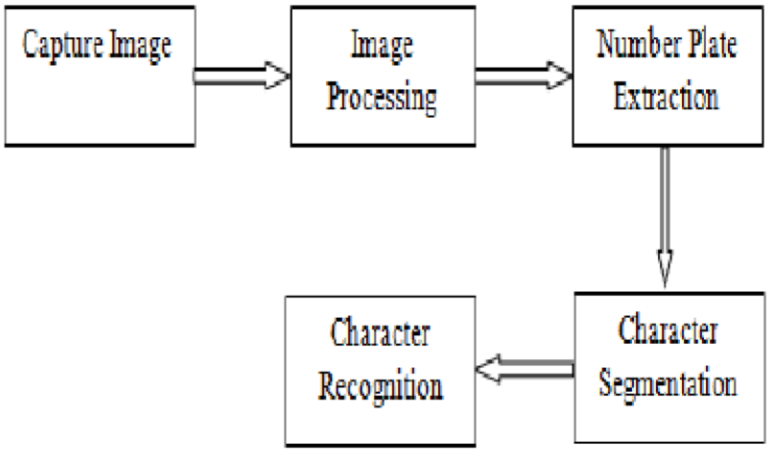
- Preprocessing of the Image
- Text Localization
- Character Segmentation
- Character Recognition
- Post Processing

.

3.2 Problem Statement

Aim of this project is to detect the text that is given in the image, recognize the text and print the same

3.3 Block diagram / system diagram



3.4 Objectives

We propose to implement the text detection algorithm , and classify the detected text using the OCR algorithm Tesseract . We will try to make these steps robust to different imaging conditions, and will explore various processing steps throughout the pipeline to optimize the performance of the text detection.

3.5 Text recognition

Tesseract — is an optical character recognition engine with open-source code, this is the most popular and qualitative OCR-library.

OCR uses artificial intelligence for text search and its recognition on images.

Tesseract is finding templates in pixels, letters, words and sentences. It uses two-step approach that calls adaptive recognition. It requires one data stage for character recognition, then the second stage to fulfil any letters, it wasn't insured in, by letters that can match the word or sentence contex

Chapter 4

Implementation and experimentation

This chapter presents the steps in implementing a classifier, the fuctions of the software used along with its parameters and the complete code of the project with respective results.

4.1 Steps in implementing a classifier for text categorizaion

ALGORITHM:

- OCR:

Optical character recognition or optical character reader (OCR) is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo

- Text Detection:

Text detection techniques required to detect the text in the image and create and bounding box around the portion of the image having text. Standard objection detection techniques will also work here.

The aim of this Project is to be able to recognise text from an image file using the Tesseract Library in the Python Programming Language.
Tesseract is an Open Source library for Optical Character recognition (OCR). We will be using print the recognized text given an input image of any of the following formats: jpeg, png, gif.

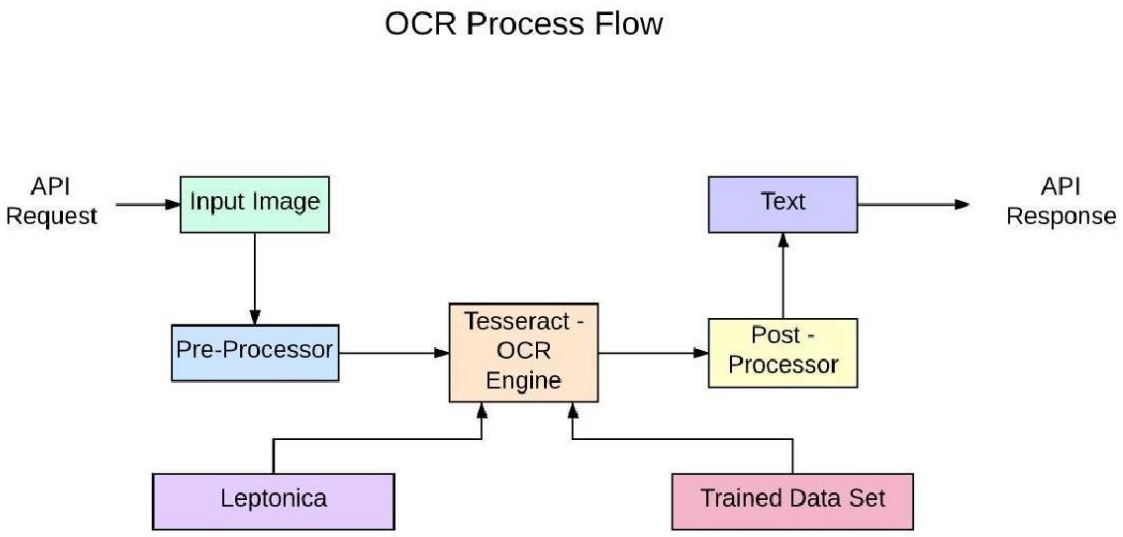
The subprocesses are:

- Preprocessing of the Image

- Text Localization

- Character Segmentation
- Character Recognition

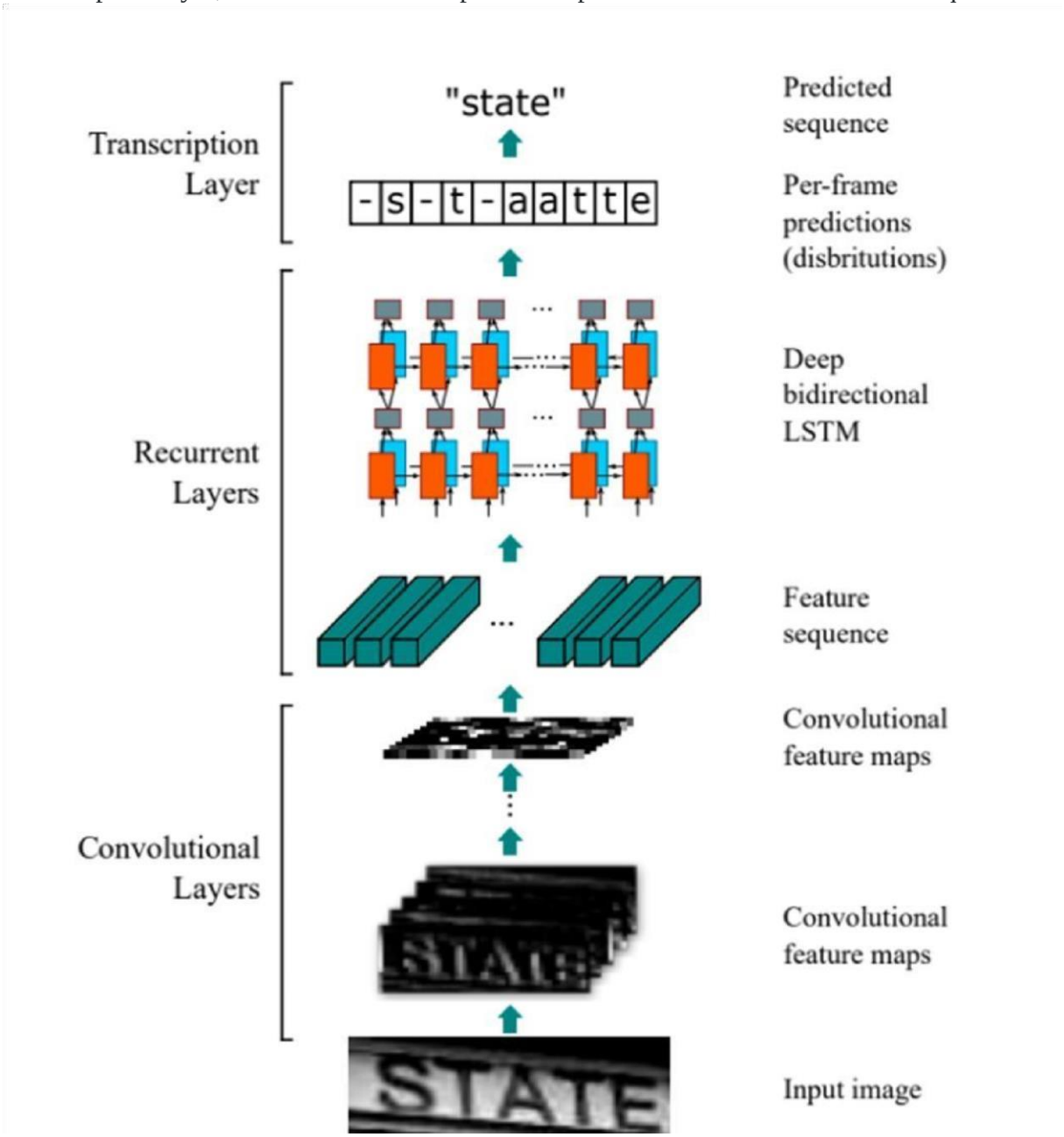
- Post Processing



The architecture consists of three parts:

1) convolutional layers, which extract a feature sequence from the input image

2) recurrent layers, which predict a label distribution for each frame
3) transcription layer, which translates the per-frame predictions into the final label sequence.



4.2 Software Implementation

Functions Used:

- get_tesseract_version - Returns the Tesseract version installed in the system.
- image_to_string - Returns the result of a Tesseract OCR run on the image to string
- image_to_boxes - Returns result containing recognized characters and their box boundaries
- image_to_data - Returns result containing box boundaries, confidences, and other information.

Requires Tesseract 3.05+. For more information, please check the Tesseract TSV documentation

- image_to_osd - Returns result containing information about orientation and script detection.
- run_and_get_output - Returns the raw output from Tesseract OCR. Gives a bit more control over the parameters that are sent to tesseract.

Parameters :

image_to_data(image, lang=None, config='', nice=0, output_type=Output.STRING, timeout=0, pandas_config=None)

- image Object or String - PIL Image/NumPy array or file path of the image to be processed by Tesseract. If you pass object instead of file path, pytesseract will implicitly convert the image to RGB mode.

- lang String - Tesseract language code string. Defaults to eng if not specified! Example for multiple languages:

lang='eng+fra'

- config String - Any additional custom configuration flags that are not available via the pytesseract function. For example: config='--psm 6'

- nice Integer - modifies the processor priority for the Tesseract run. Not supported on Windows. Nice adjusts the niceness of unix-like processes.

- output_type Class attribute - specifies the type of the output, defaults to string. For the full list of all supported types, please check the definition of pytesseract.Output class.

4.3 Program :

This section of code downloads testing image from url.

```
import requests

print ('Starting to Download!')

url = 'http://pun.me/pages/funny-quote-about-life.jpg' r= requests.get(url)

filename = '2.jpg'

with open(filename, 'wb') as out_file:
out_file.write(r.content) print("Download complete!")
```

This section of code performs OCR

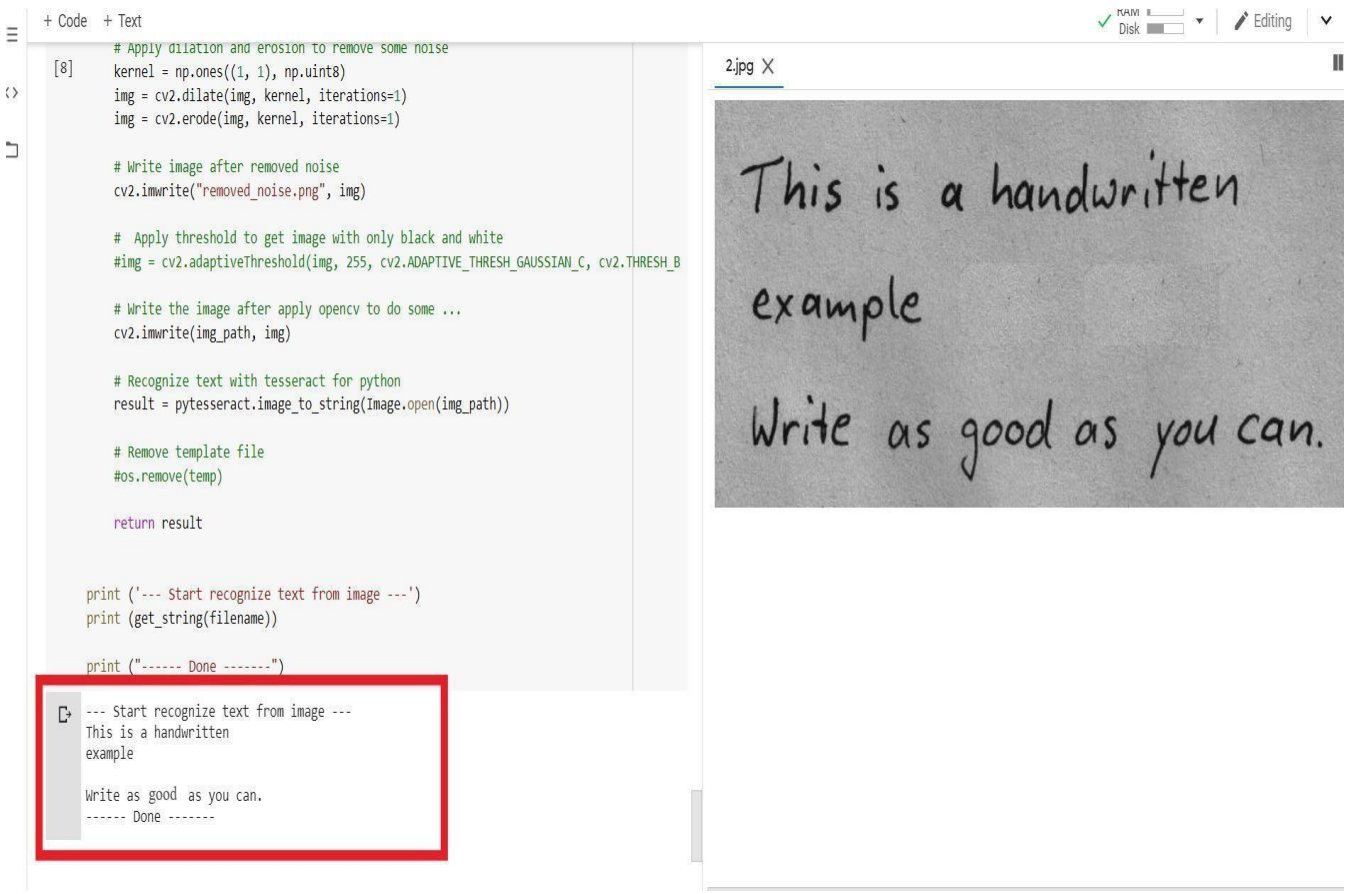
```
import cv2 import
numpy as np import
pytesseract
from PIL import Image

# Path of working folder on Disk

def get_string(img_path):

    # Read image with opencv
    img = cv2.imread(img_path)

    # Convert to gray
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
```

```
    # Apply dilation and erosion to remove some noise
kernel = np.ones((1, 1), np.uint8)
img = cv2.dilate(img, kernel, iterations=1)
img = cv2.erode(img, kernel, iterations=1)

    # Write image after removed noise
    cv2.imwrite("removed_noise.png", img)

#  Apply threshold to get image with only black and white        #
Write the image after apply opencv to do some ...
cv2.imwrite(img_path, img)

 # Recognize text with tesseract for python
    result = pytesseract.image_to_string(Image.open(img_path))

    # Remove template file return result

print ('--- Start recognize text from image ---') print
(get_string(filename))

print ("------ Done------- ")
```

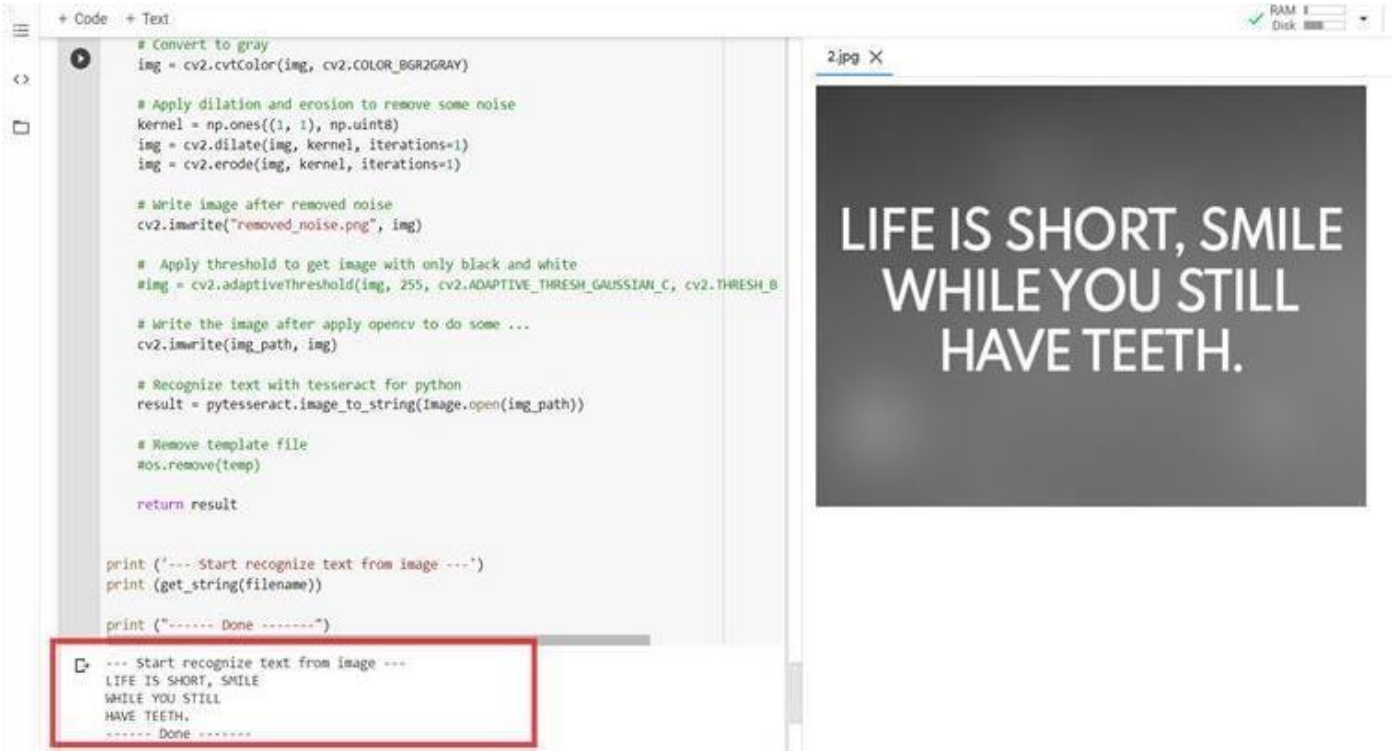
4.4 Results and discussion

The desired output here was to able to detect the location of the text in the image and highlighted the region enclosed with a red box is the output of code. Some samples have been tried below

TEST 1:



TEST 2:



```
+ Code  + Text

    # Convert to gray
    img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Apply dilation and erosion to remove some noise
    kernel = np.ones((1, 1), np.uint8)
    img = cv2.dilate(img, kernel, iterations=1)
    img = cv2.erode(img, kernel, iterations=1)

    # Write image after removed noise
    cv2.imwrite("removed_noise.png", img)

    #  Apply threshold to get image with only black and white
    #img = cv2.adaptiveThreshold(img, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_B

    # Write the image after apply opencv to do some ...
    cv2.imwrite(img_path, img)

    # Recognize text with tesseract for python
    result = pytesseract.image_to_string(Image.open(img_path))

    # Remove template file
    #os.remove(temp)

    return result


print ('--- Start recognize text from image ---')
print (get_string(filename))

print ("------ Done -------")

--- Start recognize text from image ---
LIFE IS SHORT, SMILE
WHILE YOU STILL
HAVE TEETH.
------ Done -------
```

2.jpg

Chapter 5

Conclusions and scope for further work

This chapter presents the final conclusion and the scope of text recognition for the further work.

5.1 Conclusions

Thus we successfully implemented the OCR algorithm and obtained desired results with good accuracy of more than 80% . In the above examples the accuracy was 100%. The system is stable and robust all the system parameters remain the same throughout all the experiments.

5.2 Scope for further work

The future scope of OCR includes :

1. Font Independent OCR :

An Optical Character Recognition system could be developed by considering the multiple font style in use. Our approach is very much useful for the font independent case. Because, for font or character size, it finds the string and the strings are parsed to recognize the character. Once character is identified, the corresponding character could be ejected through an efficient editor.

2. Cursive Characters OCR

There is heavy demand for an OCR system which recognizes cursive scripts and manuscripts like Palm Leaves. This actually avoids keyboard typing and font encoding too.

3. Language Converter through OCR:

OCR has been developed for two languages with font encoding, spell checker and grammatical

sentence check, then a converter could be implemented to convert sentences from one language to another through a transliteration and translation scheme.

4.Speech recognition from OCR:

The most required application today is Speech recognition. The recognized Printed or Handwritten character could be recorded and through a voice synthesizer speech output could be generated. This would help the blind to send and receive information.

References

1. Ray Smith, "An overview of the tesseract OCR engine," 2005.

2. Pratik MadhukarManwatkar, Dr. Kavita R. Singh, "A technical review on text recognition from images," IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO), 2015.

3. Akhilesh A. Panchal, ShrugalVarde, M.S. Panse, "Character detection and recognition system for visually impaired people," IEEE International Conference On Recent Trends In Electronics Information Communication Technology, May 20-21, 2016.

4. Nada Farhani, NaimTerbeh, MounirZrigui, "Image to text conversion: state of the art and extended work," IEEE/ACS 14th International Conference on Computer Systems and Applications, 2017.

5. Azmi Can Özgen, MandanaFasounaki, Hazım Kemal Ekenel, "Text detection in natural and computergenerated images," 2017.

6. Christian Reul, UweSpringmann, Christoph Wick, Frank Puppe, "Improving OCR accuracy on early printed books by utilizing cross fold training and voting," 13th IAPR International Workshop on Document Analysis Systems, 2018.

Acknowledgement

We would firstly like to thank Ms. Jayshree Kundargi Ma'am for her invaluable support and guidance throughout the project.

We would also like to thank the Department of Electronics & Telecommunications and K.J Somaiya College of Engineering for giving us the opportunity to do this project.

Last but not least, we would like to thank our family and friends who contributed by supporting us morally throughout the project.