Indian Institute of Technology Indore

Department of Computer Science and Engineering

Machine Learning Project

CS 403/603

Document Text Recognition

Authors:

Amit Raj(2204101010)
Drishti Sharma(2204101002)

Under the guidance of :

Dr. Puneet Gupta
Assistant Professor
Department of Computer Science and Engineering
IIT Indore

Date
18-11-2022

Acknowledgement

We would like to express our deepest appreciation to all those who provided us the possibility to complete this report. A special gratitude I give to the project guide, Dr. Puneet upta, whose contribution in stimulating suggestions and encouragement, helped us to coordinate our project. Furthermore, we would also like to acknowledge with much appreciation the crucial role of our TA Mr. Anup Kumar Gupta, who helped us to look for all the required and necessary materials to complete the study.

Amit Raj(2204101010)
Drishti Sharma(2204101002)

Contents

List of Figures

1 Introduction

In this project, we are going to achieve below given objectives-

- Creating a dataset of images that contain texts to be recognized.
- Using python libraries and functions for detection and recognition.
- Saving the recognized text into .txt files.

2 Problem statement

Detect text from the provided document and save the recognized text into local storage in .txt format.

3 Theory

3.1 What is Optical Character Recognition (OCR)?

Optical character recognition is another name for text recognition (OCR). Data is extracted and reused from scanned documents, camera photos, and image-only PDFs by an OCR program. The original material can be accessed and edited by using OCR software, which isolates letters on the image, turns them into words, and then turns the words into sentences. Additionally, it does away with the requirement for human data entry. There are two parts of OCR.

1. **Text Detection-** The initial step in the process is text detection, which identifies the textual elements present in the image. For the second stage of OCR, it is crucial to localize the text within the image.
2. **Text Recognization-** In the second section, writing is taken out of the picture. These methods can be combined to extract text from any image.

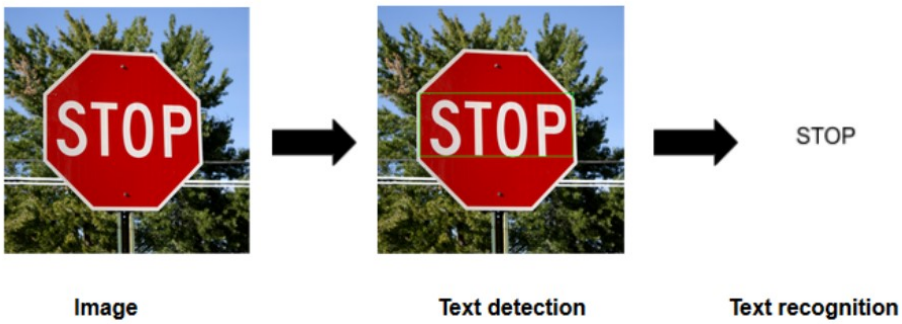

Figure 1: Flow of OCR
https://doi.org/10.48550/arXiv.1003.5893

4 Dataset

The dataset is self-created for fulfilling the vary purpose of this project. The collection of the images either belongs to the interior of the IIT Indore campus or taken from the Internet. All the images that we have used in this project are in .jpg format.

5 Methodology

There are a variety of tools and services available which are easy to use and make this task a no-brainer.

5.1 ABBYY FineReader

ABBYY FineReader or ABBYY FineReader PDF supports editing and working on PDF documents with ease is called ABBYY FineReader or ABBYY FineReader PDF. This productivity programme, created by ABBYY, features a comprehensive set of PDF capabilities that let you browse, edit, share, protect, retrieve, scan, and collaborate on PDF files without switching between programmes. It's renowned for its optical character recognition (OCR) feature, which can precisely read text.

5.2 Google Cloud Vision

The Google Cloud Vision API enables developers to understand the content of an image by encapsulating powerful machine learning models in an easy-to-use REST API. It quickly divides photographs into thousands of categories (such as "sailboat," "lion," and "Eiffel Tower"), locates and reads printed words within images, and recognizes specific items and persons within pictures. Through picture sentiment analysis, you can add information to your image database, control objectionable content, or enable new marketing scenarios. analyze the photos that were uploaded for the request or integrate your Google Cloud Store image storage.

5.3 Amazon Textract

Amazon Textract is a machine learning (ML) service that uses scanned documents to automatically extract text, handwriting, and data. To recognize, comprehend, and extract information from forms and tables, optical character recognition (OCR) is used in a more sophisticated way. These days, a lot of businesses either manually extract data from scanned documents like PDFs, pictures, tables, and forms or use basic OCR software that needs to be manually configured (which often must be updated when the form changes). Textract uses ML to read and process any form of a document, accurately extracting text, handwriting, tables, and other data without requiring manual labor to replace these time-consuming and expensive operations.

5.4 Tesseract OCR

It is an open-source optical character recognition (OCR) platform. OCR converts documents into new searchable text files, PDFs, or the majority of other widely used formats after extracting text from images and documents without a text layer. Tesseract is highly adaptable and works with most languages, including vertical text and multilingual documents.
In this project, we have used Tesseract-OCR for optimal character recognization.

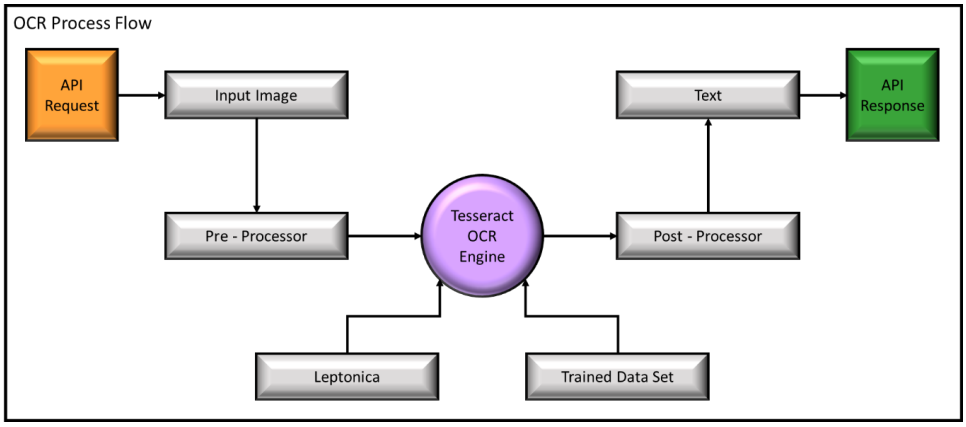

Figure 2: Working of Tesseract OCR
https://doi.org/10.48550/arXiv.1003.5893

Tesserect-OCR gives 78% overall accuracy on the document.
Tesseract-OCR gives 70% accuracy on the scanned image.
Tesseract-OCR gives 88% accuracy on the scanned document.

6 Limitation

Tesseract works best when there is a clean segmentation of the foreground text from the background. In practice, it can be extremely challenging to guarantee these types of setups. There are a variety of reasons you might not get good-quality output from Tesseract, like:

- Doesn't do well with images affected by artifacts including partial occlusion, distorted perspective, and complex background.
- It is not capable of recognizing handwriting.
- It may find gibberish and report this as OCR output.
- If a document contains languages outside of those given in the -l LANG arguments, results may be poor.
- It is not always good at analyzing the natural reading order of documents. For example, it may fail to recognize that a document contains two columns, and may try to join text across columns. Poor-quality scans may produce poor-quality OCR.
- It does not expose information about what font family text belongs to.

7 Future work

7.1 Improve the performance of Tesseract on Noisy image:

1. **Converting image to grayscale**
   A grayscale image is one in which the value of each pixel is a single sample representing only an amount of light. Grayscale images can then be the result of measuring the intensity of light at each pixel. Grayscaling the image improves the performance of Tesseract on it.
2. **Image Binarization**
   Image binarization is the process of taking an image and converting it to black-and-white, it then reduces the information contained within the grayscale version of the image from 256 shades of gray to 2: black and white, a binary image. This method works as follows-
   (a) Converts the image to grayscale
   (b) Loop through the image's pixels
   (c) Compares the value of the pixel to the threshold: If the pixel is less than the threshold, it changes its value to 0(black), and to 255(white) if not.

   Now we run our function with different values of threshold and see how it affects the performance of Tesseract.
3. **Resizing the image**
   Detecting text in a large or tiny image can be sometimes difficult. In Python PIL has a special function to helps resize our images with a multitude of options and filters.

7.2 Adding insight to recognition

OCR is moving away from just seeing and matching. Using deep learning, we can first recognize scanned text, then makes meaning of it. The competitive edge will be given to the software that provides the most powerful information extraction and highest-quality insights.

7.3 Handwritten textual recognition using OCR

Using Tesseract-OCR with the iJIT system[1], it is possible to identify unique identification tag associated with the user. Tesseract OCR engine is customized to perform user specific training on labeled handwriting samples of both isolated and free-flow texts, written using lower case Roman script. The performance is evaluated on both the categories of document pages for observation of segmentation and character recognition accuracies.

8 References

1. Rakshit, S., Basu, S., Ikeda, H. (2010). Recognition of Handwritten Textual Annotations using Tesseract Open Source OCR Engine for information Just In Time (iJIT). arXiv. https://doi.org/10.48550/arXiv.1003.5893

2. https://www.analyticsvidhya.com/blog/2020/05/build-your-own-ocr-google-tesseract-opencv/

3. https://towardsdatascience.com/the-future-of-text-recognition-is-artificial-intelligence-ead2a1c65471

4. https://www.klearstack.com/5-best-ocr-software/: :text=Tesseract