**BEST ENLIST PYTHON INTERNSHIP**

**Report on Internship Project**

**Optical Character Recognition (OCR)**

**By**

**NAVEEN KARTHIK C**

**INTRODUCTION**

Optical character recognition (OCR) method has been used in converting printed text into editable text. OCR is very useful and popular method in various applications. Accuracy of OCR can be dependent on text preprocessing and segmentation algorithms. Sometimes it is difficult to retrieve text from the image because of different size, style, orientation, complex background of image etc. We begin this paper with an introduction of Optical Character Recognition (OCR) method, History of Open Source OCR tool Tesseract, architecture of it and experiment result of OCR performed by Tesseract on different kinds images are discussed. We conclude this paper by comparative study of this tool with other commercial OCR tool Transym OCR by considering vehicle number plate as input. From vehicle number plate we tried to extract vehicle number by using Tesseract and Transym and compared these tools based on various parameters.

Python is widely used for analyzing the data but the data need not be in the required format always. In such cases, we convert that format (like PDF or JPG etc.) to the text format, in order to analyze the data in better way. Python offers many libraries to do this task.

There are several ways of doing this, including using libraries like PyPDF2 in Python.

Firstly, we need to convert the pages of the PDF to images and then, use OCR (Optical Character Recognition) to read the content from the image and store it in a text file.

**OBJECTIVE**

Converting hand written documents as scanned images or photos (in any format) to legible text document using AI extracting important and critical information into database.

**REQUIREMENTS**

* Jupyter Notebook
* Python Library (pytesseract)
* Tesseract Application

**JUPYTER NOTEBOOK**

JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones.

**PYTHON LIBRARY (pytesseract)**

Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images.Python-tesseract is a wrapper for Google’s tesseract OCR machine. It is also useful as a stand-alone invocation script to tesseract, as it can read all image types supported by the Pillow and Leptonica imaging libraries, including jpeg, png, gif, bmp, tiff, and others. Additionally, if used as a script, Python-tesseract will print the recognized text instead of writing it to a file.

**TESSERACT APPLICATION**

Tesseract is an optical character recognition engine for various operating systems. Tesseract is suitable for use as a backend and can be used for more complicated OCR tasks including layout analysis by using a frontend such as OCRopus.Tesseract's output will have very poor quality if the input images are not preprocessed to suit it: Images (especially screenshots) must be scaled up such that the text x-height is at least 20 pixels, any rotation or skew must be corrected or no text will be recognized, low-frequency changes in brightness must be high-pass filtered, or Tesseract's binarization stage will destroy much of the page, and dark borders must be manually removed, or they will be misinterpreted as characters.

**CODE**

import pytesseract

pytesseract.pytesseract.tesseract\_cmd = r'C:\Program Files\Tesseract- OCR\tesseract'

text=pytesseract.image\_to\_string(r"C:\Users\PRAKASH\Desktop\IMG2.jpg")

f = open("recognized.txt", "w")

f.write(text)

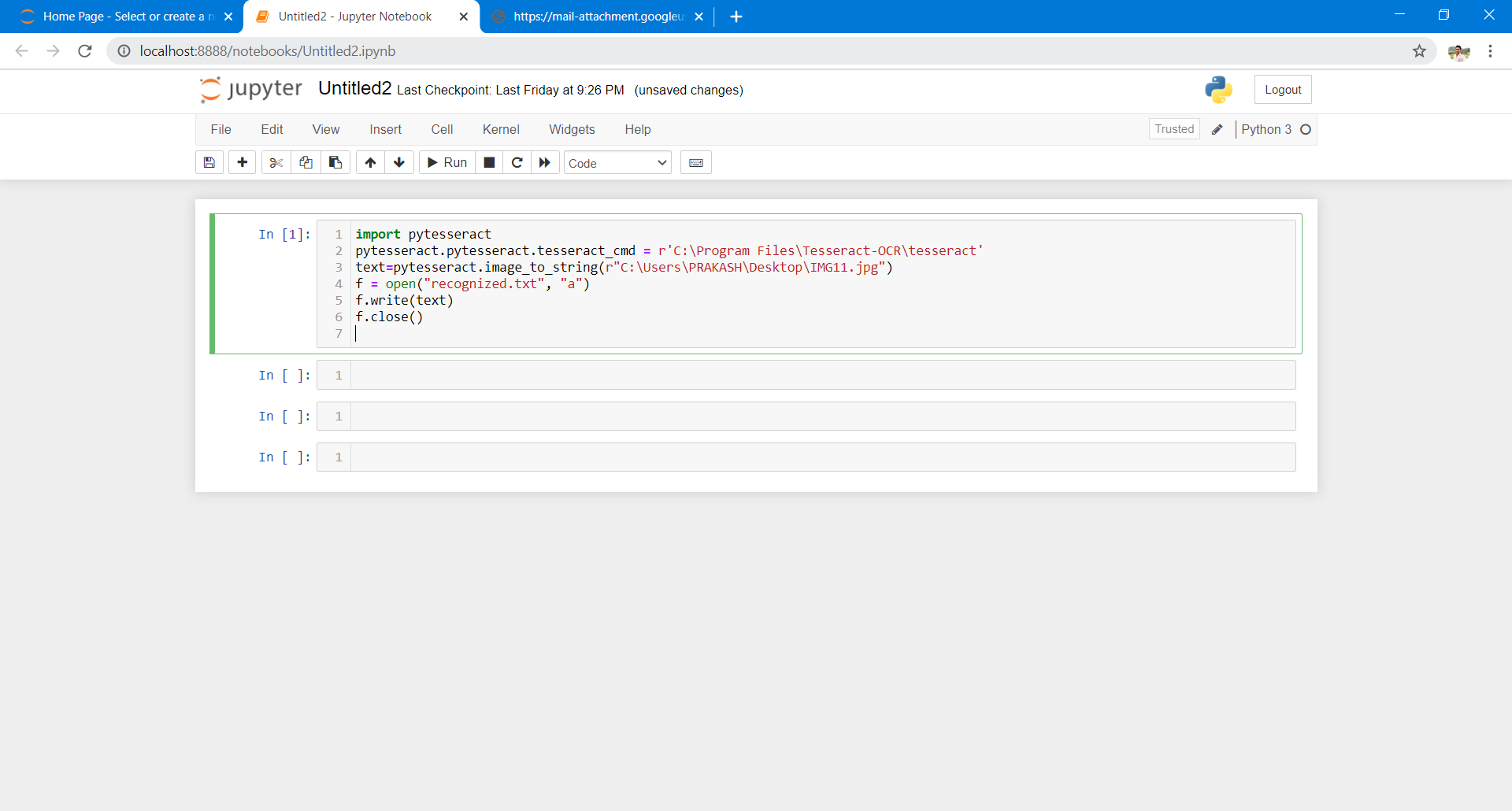
f.close()

**IMPLEMENTATION**

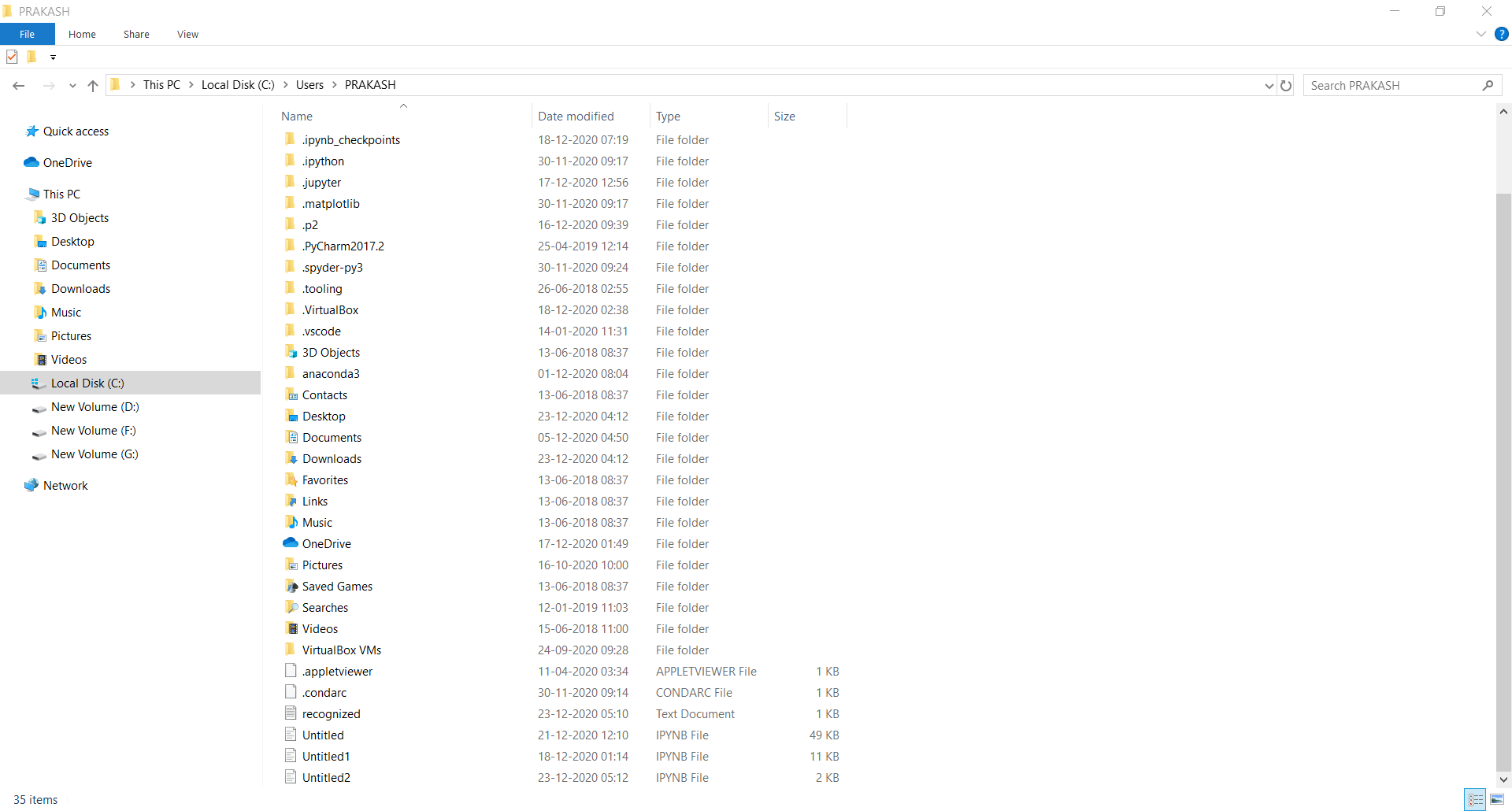
* **Loading an Image saved from the computer or download it using a browser and then loading the same. (Any Image with Text).**
* **Binarizing the Image (Converting Image to Binary).**
* **We will then Pass the Image through the OCR system.**
* Install the pytesseract package so that we can access Tesseract via the Python programming language.
* **Line 1** handle our imports. The Image class is required so that we can load our input image from disk in PIL format, a requirement when using pytesseract.
* **Line 2** specifies the path where the tesseract module is installed.
* **On Line 3,** using pytesseract.image\_to\_string, we convert the contents of the image into our desired string, text.
* Then write text in a text file “recognized.txt” and save it.
* Then in this line, I create a file with append mode and write the text extracted from the given image.
* The file is automatically saved.

**SCREENSHOTS**

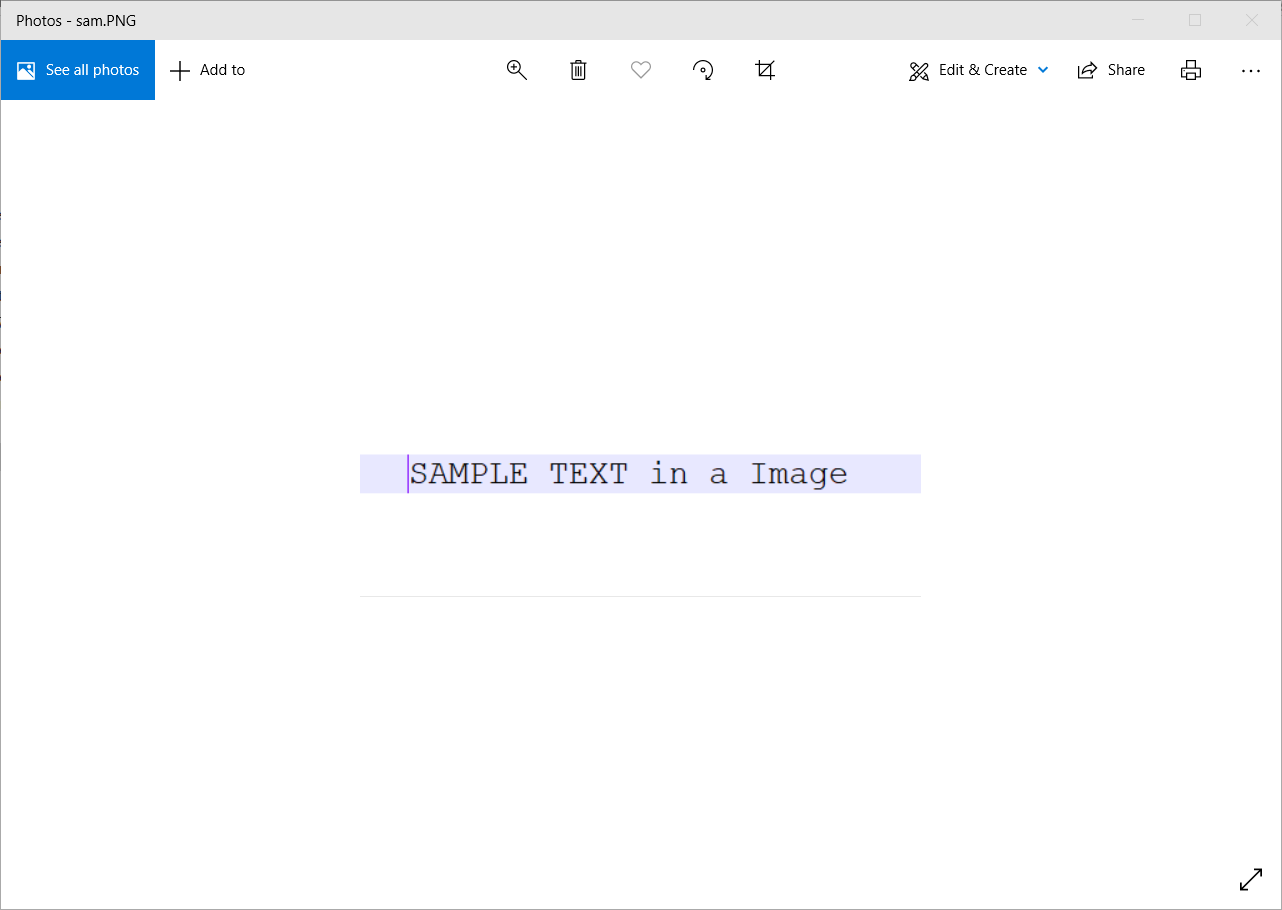
**Code:**

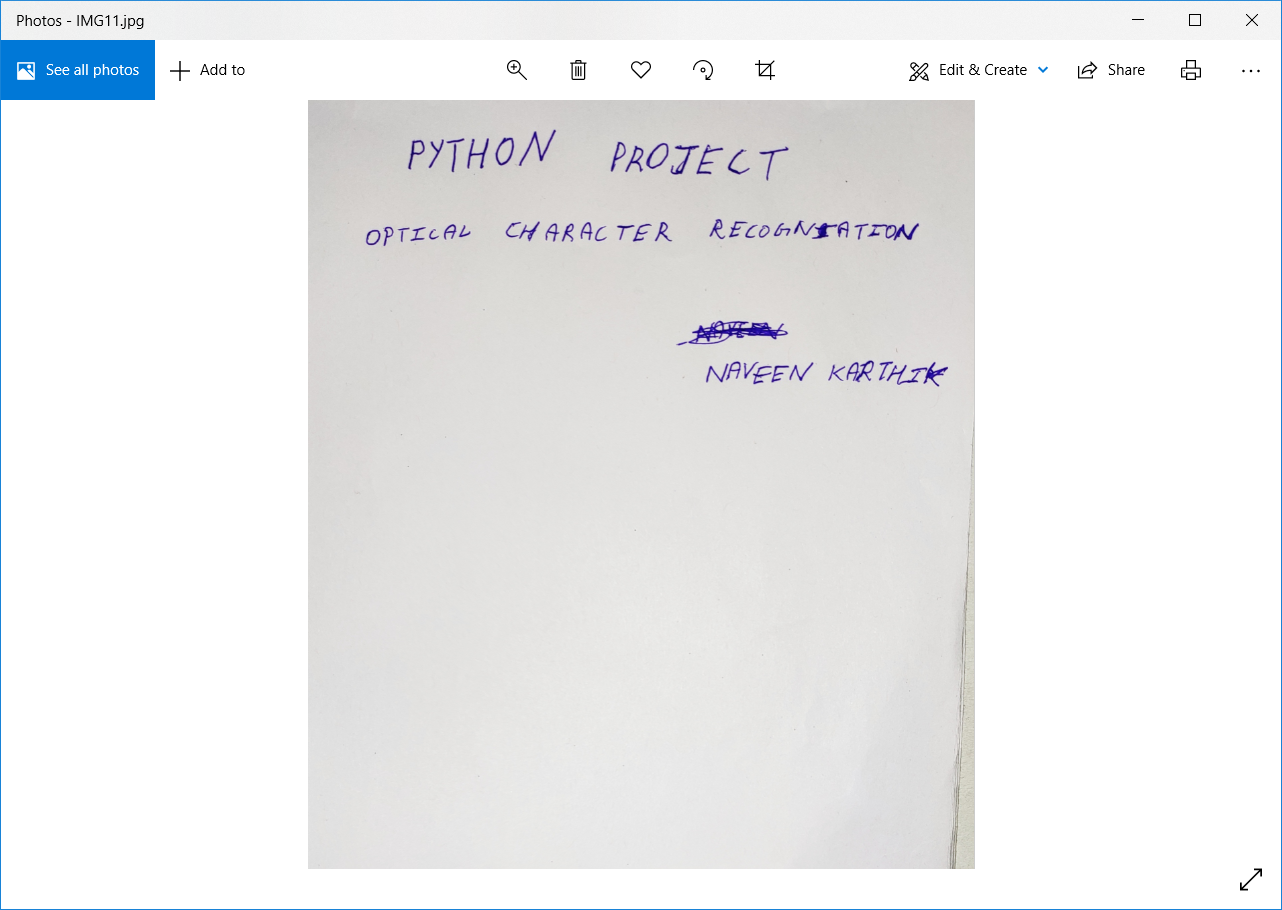


**Recognized output file:**

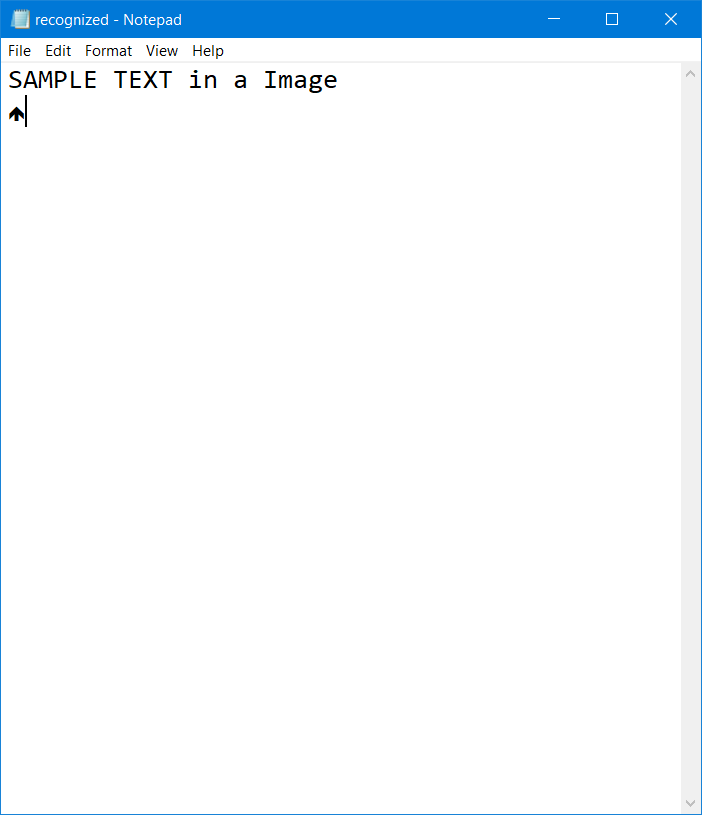


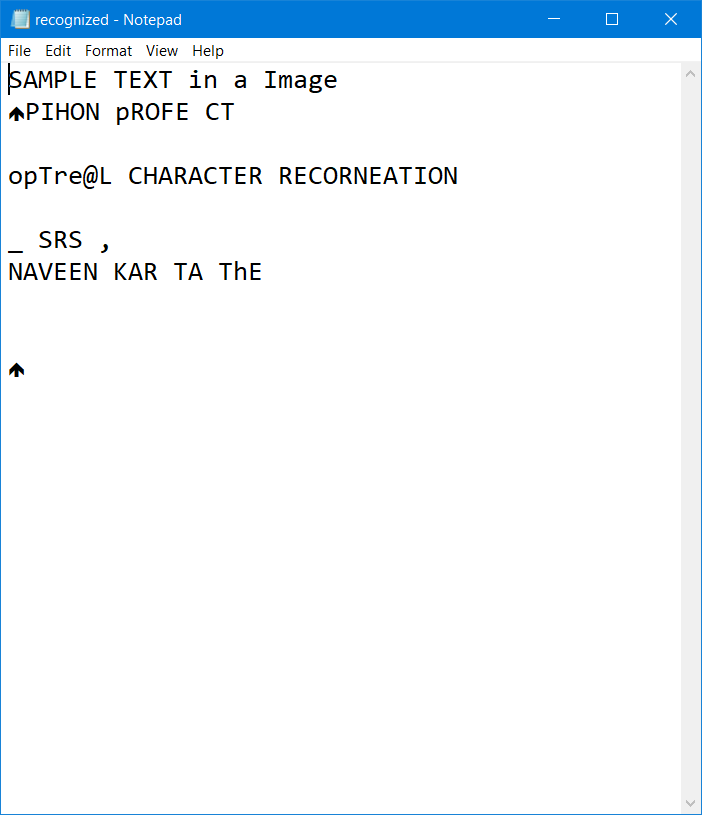
Sample Inputs:





**Output screenchots:**





**CONCLUSION**

OCR, while no longer a new technology, is still an active area of research in the computer vision literature *especially* when applying OCR to real-world, unconstrained images. Deep learning and Convolutional Neural Networks (CNNs) are certainly enabling us to obtain higher accuracy, but we are still a long way from seeing “near perfect” OCR systems. Furthermore, as OCR has many applications across many domains, some of the best algorithms used for OCR are commercial and require licensing to be used in our own projects.

**REFERENCES**

<https://www.google.com/amp/s/www.geeksforgeeks.org/python-reading-contents-of-pdf-using-ocr-optical-character-recognition/amp/>

<https://www.google.com/amp/s/www.geeksforgeeks.org/python-convert-image-to-text-and-then-to-speech/amp/>

<https://medium.com/@ageitgey/python-3-quick-tip-the-easy-way-to-deal-with-file-paths-on-windows-mac-and-linux-11a072b58d5f#:~:text=To%20use%20it%2C%20you%20just,for%20the%20current%20operating%20system>

<https://realpython.com/creating-modifying-pdf/>

<https://www.geeksforgeeks.org/reading-text-from-the-image-using-tesseract/>

<https://appliedmachinelearning.blog/2018/06/30/performing-ocr-by-running-parallel-instances-of-tesseract-4-0-python/>

<https://www.tensorscience.com/ocr/optical-character-recognition-ocr-with-python-and-tesseract-4-an-introduction>

<https://towardsdatascience.com/optical-character-recognition-ocr-with-less-than-12-lines-of-code-using-python-48404218cccb>