

Guided Project Report

Text Detection

Name: Shruti Verma
Course: AI and ML
(Batch 4)
Duration: 10 months

Problem Statement: Implement a text detection and extraction model using OpenCV and OCR

Prerequisites

What things you need to install the software and how to install them:

Python 3.8 or higher versions This setup requires that your machine has latest version of python. The following url <https://www.python.org/downloads/> can be referred to download python. Once you have python downloaded and installed, you will need to setup PATH variables (if you want to run python program directly, detail instructions are below in how to run software section). To do that check this: <https://www.pythoncentral.io/add-python-to-path-python-is-not-recognized-as-an-internal-or-external-command/>. Setting up PATH variable is optional as you can also run program without it and more instruction are given below on this topic.

Second and easier option is to download anaconda and use its anaconda prompt to run the commands. To install anaconda check this url <https://www.anaconda.com/download/> You will also need to download and install below 3 packages after you install either python or anaconda from the steps above Sklearn (scikit-learn) numpy scipy if you have chosen to install python 3.8 then run below commands in command prompt/terminal to install these packages `pip install -U scikit-learn` `pip install numpy` `pip install scipy` if you have chosen to install anaconda then run below commands in anaconda prompt to install these packages `conda install -c scikit-learn` `conda install -c anaconda numpy` `conda install -c anaconda scipy` . Install the pytesseract using the command, `pip install pytesseract` and open CV using `pip install opencv-python`.

Video Link

<https://drive.google.com/drive/folders/1nKjHKic6ZATLwmXNwcyb1Guxb1LNOrrO>

Dataset used

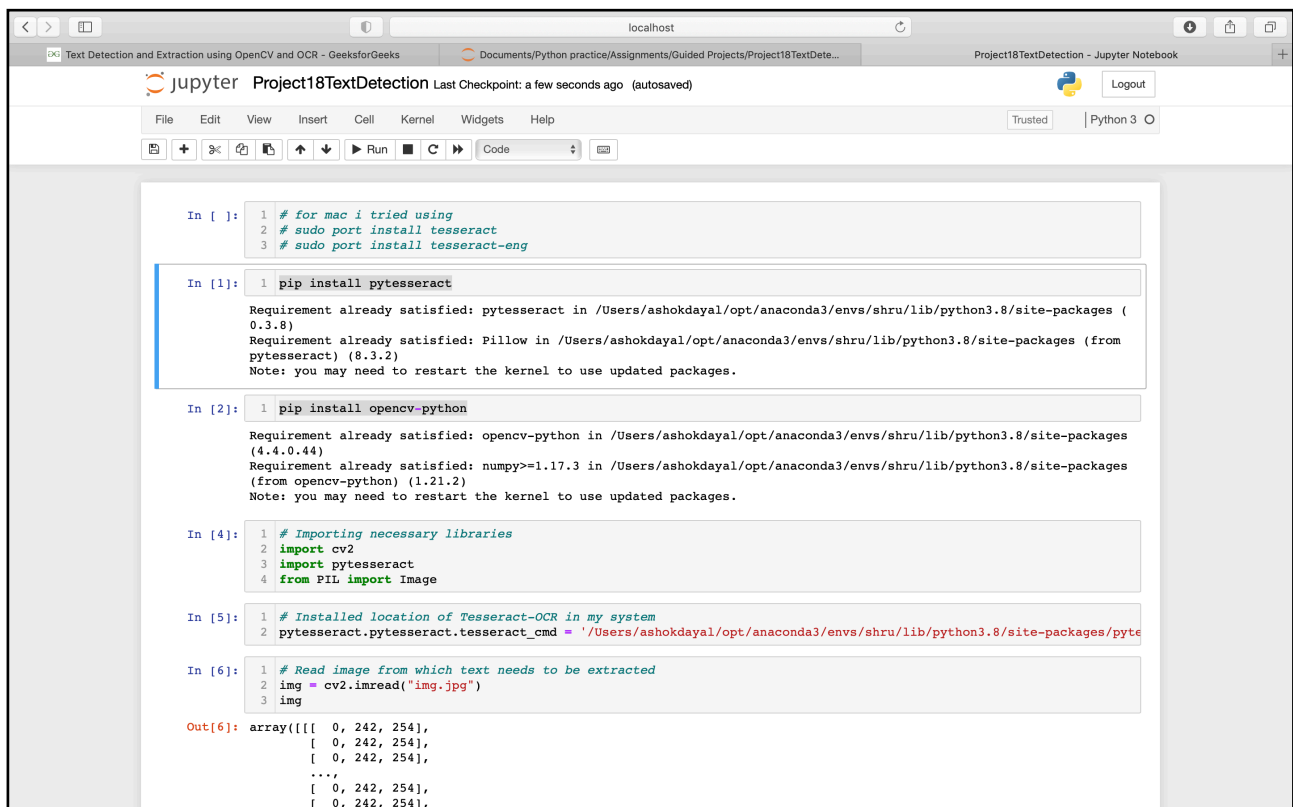
Sample image with text - img.jpeg

Method used for detection

- Image Preprocessing
- Contours representing the text areas
- Applying OCR using python-tesseract

Importing the libraries and capturing images:

Installing python-tesseract and openCV



The screenshot shows a Jupyter Notebook titled "Project18TextDetection" with a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar. The notebook contains several code cells:

```
In [ ]: 1 # for mac i tried using
        2 # sudo port install tesseract
        3 # sudo port install tesseract-eng

In [1]: 1 pip install pytesseract

Requirement already satisfied: pytesseract in /Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages (0.3.8)
Requirement already satisfied: Pillow in /Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages (from pytesseract) (8.3.2)
Note: you may need to restart the kernel to use updated packages.

In [2]: 1 pip install opencv-python

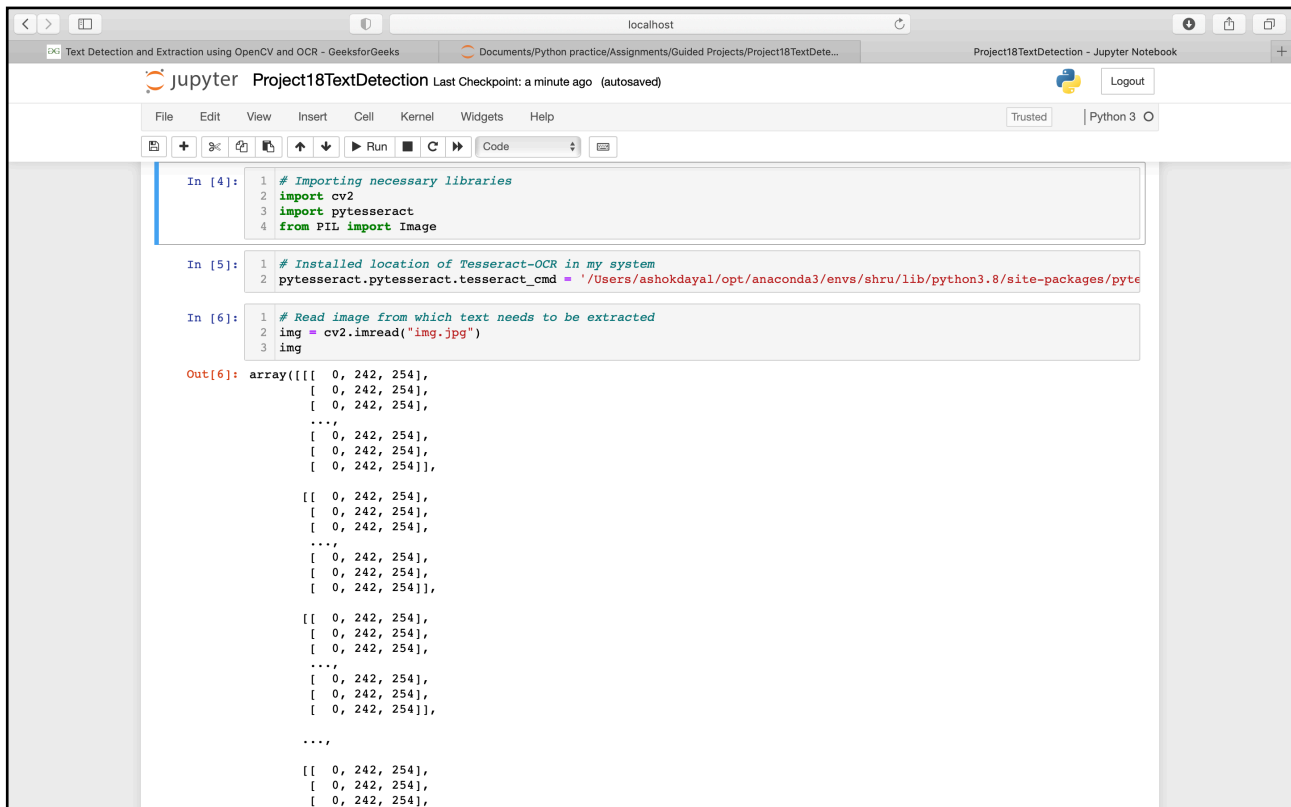
Requirement already satisfied: opencv-python in /Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages (4.4.0.44)
Requirement already satisfied: numpy>=1.17.3 in /Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages (from opencv-python) (1.21.2)
Note: you may need to restart the kernel to use updated packages.

In [4]: 1 # Importing necessary libraries
        2 import cv2
        3 import pytesseract
        4 from PIL import Image

In [5]: 1 # Installed location of Tesseract-OCR in my system
        2 pytesseract.pytesseract.tesseract_cmd = '/Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages/pyte

In [6]: 1 # Read image from which text needs to be extracted
        2 img = cv2.imread("img.jpg")
        3 img

Out[6]: array([[ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254],
               ...,
               [ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254],
```



```
In [4]: 1 # Importing necessary libraries
2 import cv2
3 import pytesseract
4 from PIL import Image

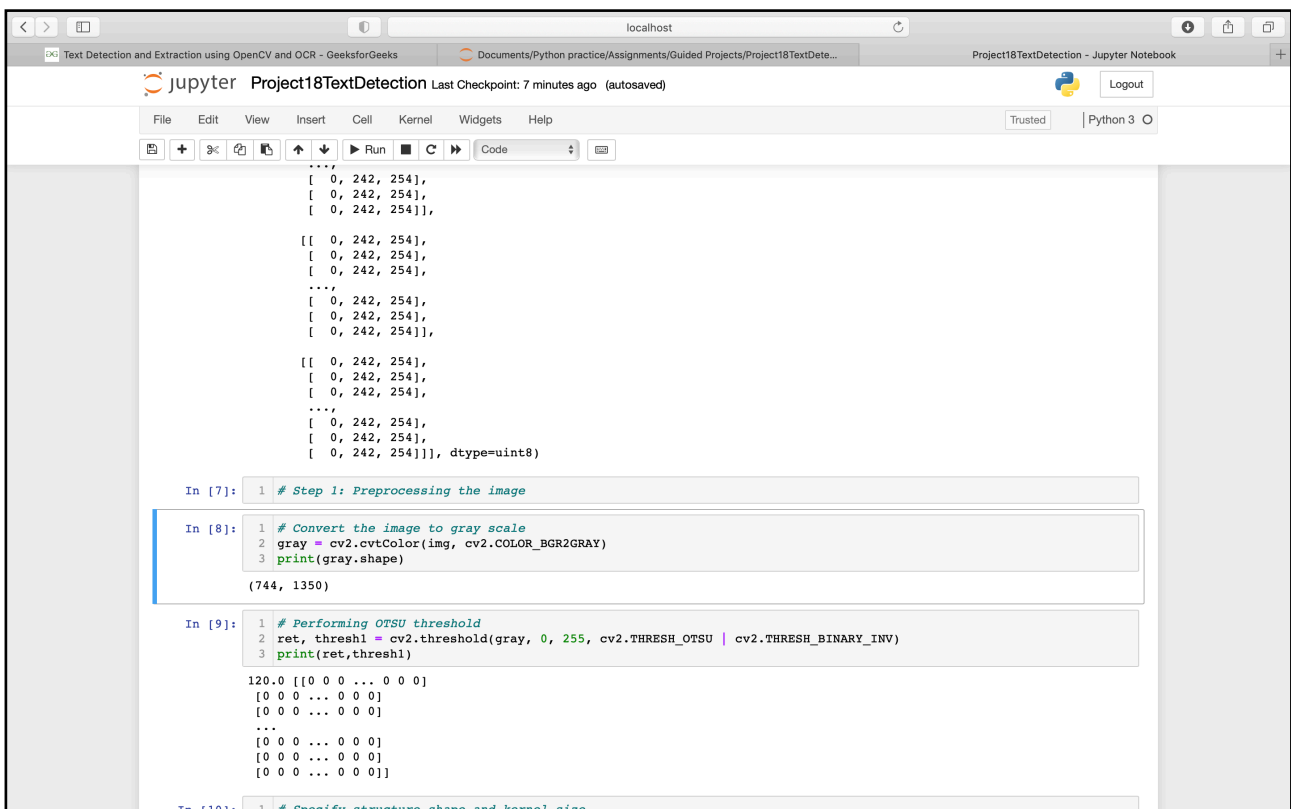
In [5]: 1 # Installed location of Tesseract-OCR in my system
2 pytesseract.pytesseract.tesseract_cmd = '/Users/ashokdayal/opt/anaconda3/envs/shru/lib/python3.8/site-packages/pyte

In [6]: 1 # Read image from which text needs to be extracted
2 img = cv2.imread("img.jpg")
3 img

Out[6]: array([[ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254],
               ...,
               [ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254]],
              ...,
              [[ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254],
               ...,
               [ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254]],
              ...,
              [[ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254],
               ...,
               [ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254]],
              ...,
              [[ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254],
               ...,
               [ 0, 242, 254],
               [ 0, 242, 254],
               [ 0, 242, 254]]]
```

Importing necessary libraries and reading the image 'img'

Preprocessing the image



```
[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254]],
[[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254],
...,
[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254]],
...,
[[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254],
...,
[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254]],
...,
[[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254]], dtype=uint8)

In [7]: 1 # Step 1: Preprocessing the image

In [8]: 1 # Convert the image to gray scale
2 gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
3 print(gray.shape)

(744, 1350)

In [9]: 1 # Performing OTSU threshold
2 ret, thresh1 = cv2.threshold(gray, 0, 255, cv2.THRESH_OTSU | cv2.THRESH_BINARY_INV)
3 print(ret, thresh1)

120.0 [[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
...
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]]

In [10]: 1 # Specify structure shape and kernel size.
```

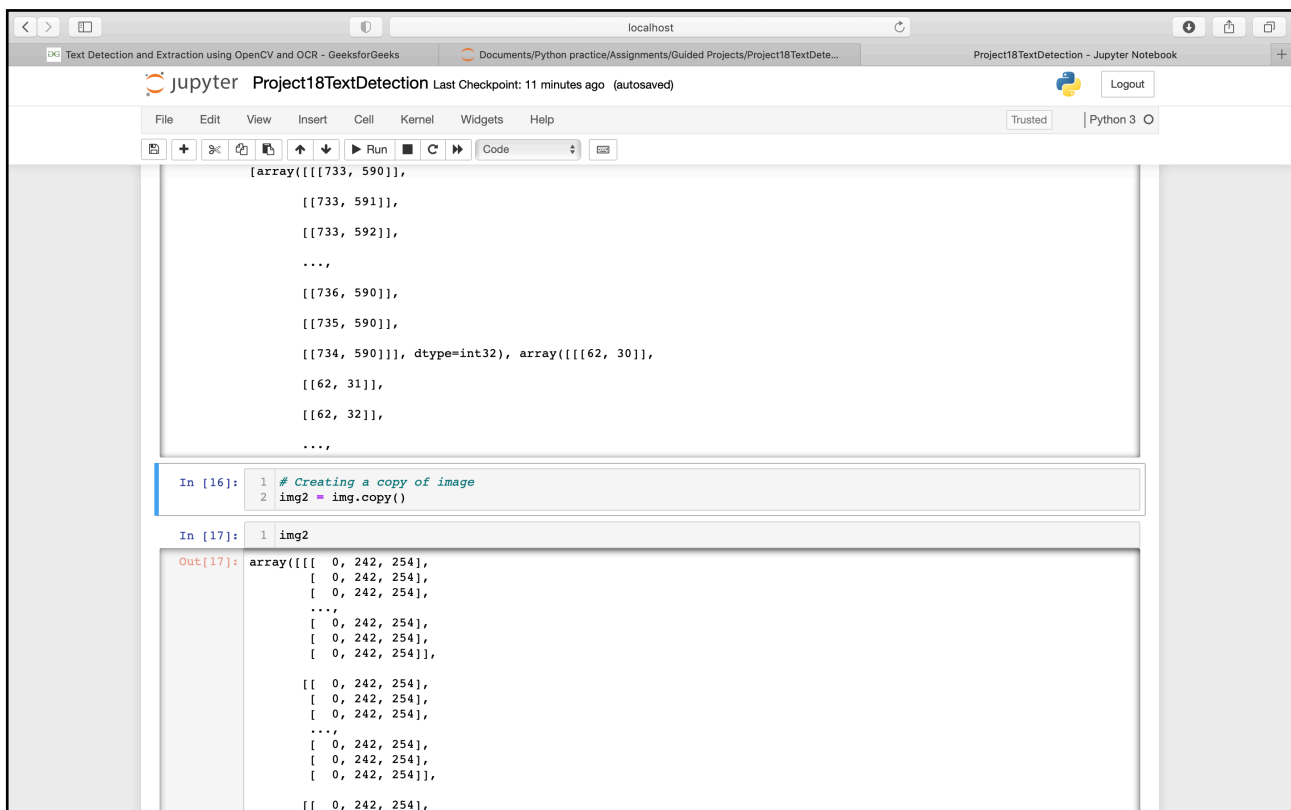
Rectangle kernel specifications

[illegible]

Applying dilation and finding contours

[illegible]

Copy the image



The screenshot shows a Jupyter Notebook titled "Project18TextDetection" with a "Last Checkpoint: 11 minutes ago (autosaved)". The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook content is as follows:

```
[array([[733, 590]],

       [[733, 591]],

       [[733, 592]],

       ...,

       [[736, 590]],

       [[735, 590]],

       [[734, 590]], dtype=int32), array([[62, 30]],

       [[62, 31]],

       [[62, 32]],

       ...,

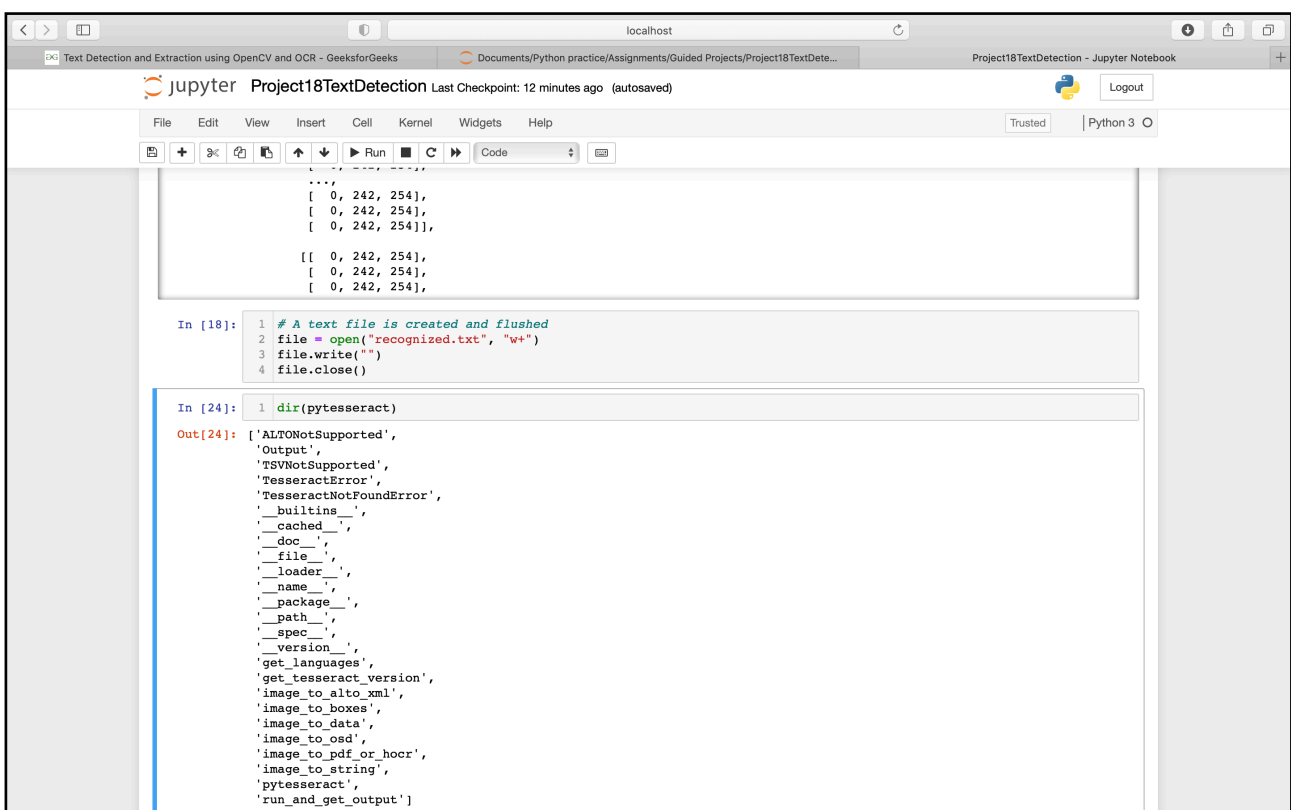
In [16]: 1 # Creating a copy of image
         2 img2 = img.copy()

In [17]: 1 img2
Out[17]: array([[ 0, 242, 254],
                 [ 0, 242, 254],
                 [ 0, 242, 254],
                 ...,
                 [ 0, 242, 254],
                 [ 0, 242, 254],
                 [ 0, 242, 254]],

                [[ 0, 242, 254],
                 [ 0, 242, 254],
                 [ 0, 242, 254],
                 ...,
                 [ 0, 242, 254],
                 [ 0, 242, 254],
                 [ 0, 242, 254]],

                [[ 0, 242, 254],
```

Creating an empty text file and using image to string from pytesseract



The screenshot shows the same Jupyter Notebook interface, but with additional code cells. The notebook title is "Project18TextDetection" and the last checkpoint is "12 minutes ago (autosaved)". The content is as follows:

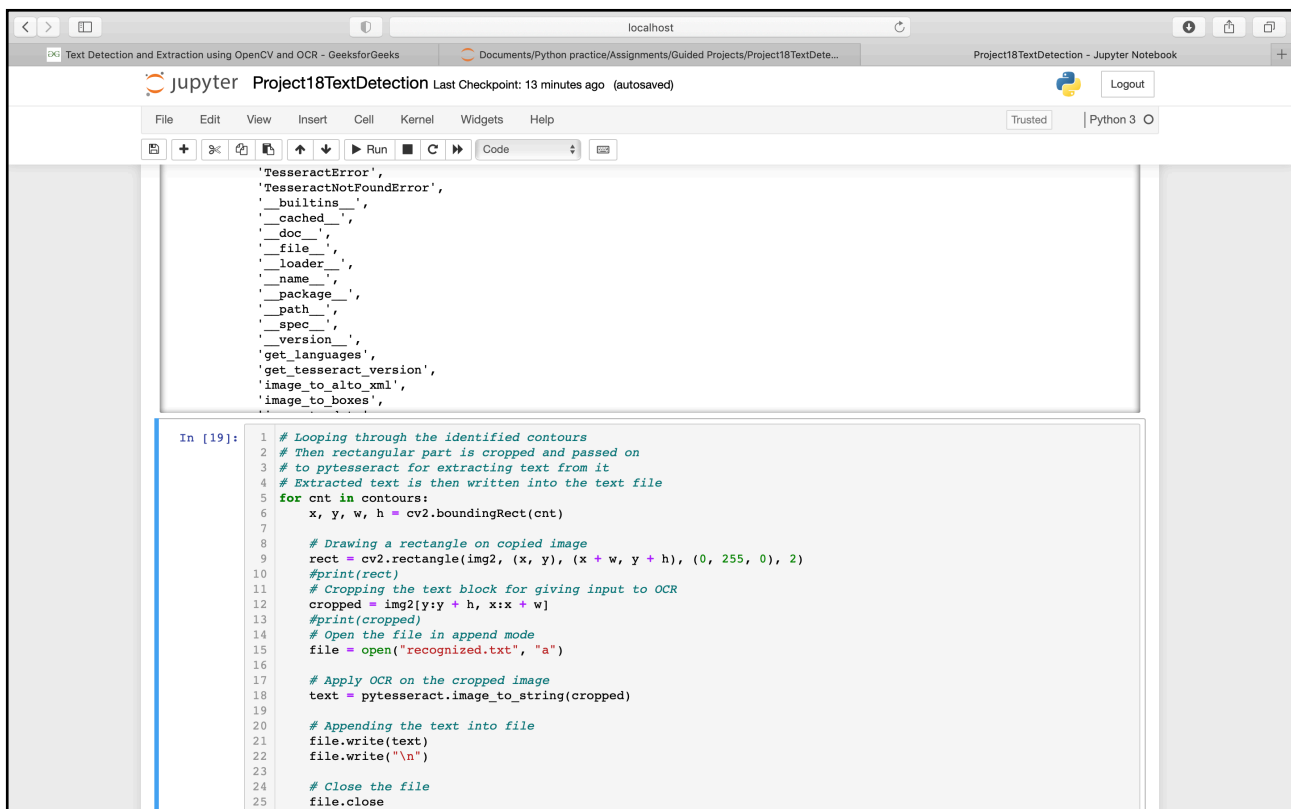
```
...
[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254]],

[[ 0, 242, 254],
[ 0, 242, 254],
[ 0, 242, 254],

In [18]: 1 # A text file is created and flushed
         2 file = open("recognized.txt", "w+")
         3 file.write("")
         4 file.close()

In [24]: 1 dir(pytestesseract)
Out[24]: ['ALTONotSupported',
          'Output',
          'TSVNotSupported',
          'TesseractError',
          'TesseractNotFoundError',
          '_builtins_',
          '_cached_',
          '_doc_',
          '_file_',
          '_loader_',
          '_name_',
          '_package_',
          '_path_',
          '_spec_',
          '_version_',
          'get_languages',
          'get_tesseract_version',
          'image_to_alto_xml',
          'image_to_boxes',
          'image_to_data',
          'image_to_osd',
          'image_to_pdf_or_hocr',
          'image_to_string',
          'pytestesseract',
          'run_and_get_output']
```

Applying OCR on the cropped text and appending in the text file and closing file



The screenshot shows a Jupyter Notebook titled 'Project18TextDetection' running on a localhost. The notebook has a menu bar with File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. Below the menu bar is a toolbar with icons for file operations, running, and other functions. The notebook content is divided into two main sections. The top section contains a list of attributes for the 'pytesseract' module, including 'TesseractError', 'TesseractNotFoundError', 'builtins', 'cached', 'doc', 'file', 'loader', 'name', 'package', 'path', 'spec', 'version', 'get_languages', 'get_tesseract_version', 'image_to_alto_xml', and 'image_to_boxes'. The bottom section, labeled 'In [19]:', contains a Python script that processes contours from an image, crops the text, applies OCR using pytesseract, and appends the result to a file named 'recognized.txt'.

```
'TesseractError',
'TesseractNotFoundError',
'builtins',
'cached',
'doc',
'file',
'loader',
'name',
'package',
'path',
'spec',
'version',
'get_languages',
'get_tesseract_version',
'image_to_alto_xml',
'image_to_boxes',
...
```

```
In [19]: 1 # Looping through the identified contours
2 # Then rectangular part is cropped and passed on
3 # to pytesseract for extracting text from it
4 # Extracted text is then written into the text file
5 for cnt in contours:
6     x, y, w, h = cv2.boundingRect(cnt)
7
8     # Drawing a rectangle on copied image
9     rect = cv2.rectangle(img2, (x, y), (x + w, y + h), (0, 255, 0), 2)
10    #print(rect)
11    # Cropping the text block for giving input to OCR
12    cropped = img2[y:y + h, x:x + w]
13    #print(cropped)
14    # Open the file in append mode
15    file = open("recognized.txt", "a")
16
17    # Apply OCR on the cropped image
18    text = pytesseract.image_to_string(cropped)
19
20    # Appending the text into file
21    file.write(text)
22    file.write("\n")
23
24    # Close the file
25    file.close
```