December 22, 2020

Prakash

Saveetha egineering college

Chennai

Documentation for OCR - Editor

Author : prakash

# Contents

[Contents 1](#_Toc59527051)

[1. Problem statement: 2](#_Toc59527052)

[2. My Solution: 2](#_Toc59527053)

[3. Requirements: 2](#_Toc59527054)

[4. Documenatation 2](#_Toc59527055)

[4.1 Constants 2](#_Toc59527056)

[4.2 Variables 3](#_Toc59527057)

[4.3 Functions 3](#_Toc59527058)

[4.3.1 Remove\_control\_characters : 3](#_Toc59527059)

[4.3.2 Get\_file 3](#_Toc59527060)

[4.3.3 Export 4](#_Toc59527061)

[4.3.4 Export\_as 5](#_Toc59527062)

[4.3.5 Save 5](#_Toc59527063)

[4.3.6 Save\_as 5](#_Toc59527064)

[4.3.7 Open\_edit 6](#_Toc59527065)

[4.3.8 Change\_image 6](#_Toc59527066)

[4.3.9 Refresh 7](#_Toc59527067)

[4.3.10 Open\_file 8](#_Toc59527068)

[4.4 Threads 8](#_Toc59527069)

[4.5 UI Elements 8](#_Toc59527070)

[4.5.1 Buttons 8](#_Toc59527071)

[4.5.2 Check boxes 9](#_Toc59527072)

[4.5.3 Sliders 9](#_Toc59527073)

[4.5.4 Text box 9](#_Toc59527074)

[4.6 Execution 9](#_Toc59527075)

[6. Code Usage 12](#_Toc59527076)

[7. Contact details 12](#_Toc59527077)

# Problem statement:

To optically recognize a handwritten image or typed text and save the detected text to an external file.

# My Solution:

I have used open CV in python for the backend code and Tkinter for the front end. Text can be recognized using tesseract module but at times, some images require preprocessing to be recognized properly.

So, the idea I came with is to design a desktop app that might help with the preprocessing and also the OCR part.

The application must be able to edit and tinker certain values of the image, like B/W, threshold limit of the image, median blur

# Requirements:

Pillow (PIL)

Pytesseract

Opencv-python

Numpy

Docx2pdf

Python-docx

# Documenatation

## Constants

Main.py

Filename (str) : the default image in the editor. It can be found in the root folder of the application

Constants.py

BG (str) : hex color code for background

FG (str) : hex color code for foreground

TEXT (str) : hex color code for Text

SECONDARY (str) : hex color code for extra color

## Variables

Black\_and\_white (IntVar) : to store the user choice, whether the image should be colourised or in black and white.

Img (PIL Image) : to save the default image in the executable file

Save\_path (str): current path of the image to be saved

Export\_path (str): current path of the text to be exported

Output (str): recognized text is stored

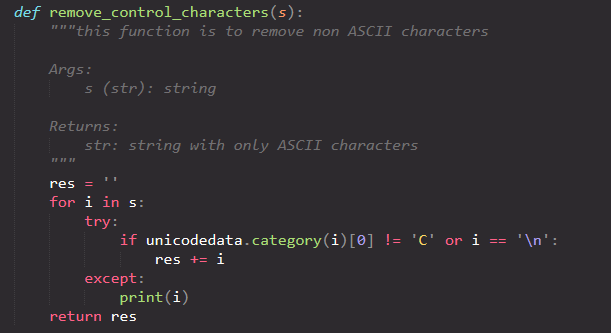
Text (str): dummy variable to save the copy of ouput

## Functions

### Remove\_control\_characters :

Args : S (str)

Returns : str : string with only ASCII characters

Usage : remove\_control\_characters(s : str) -> str : 

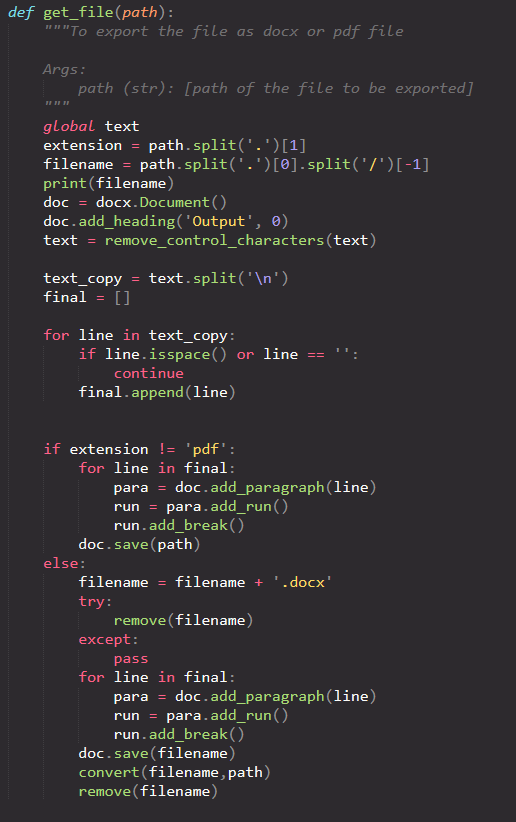
### Get\_file

Args: path (str): [path of the file to be exported]

returns: None

Function : To export the file as docx or pdf

Usage : get\_file(path):



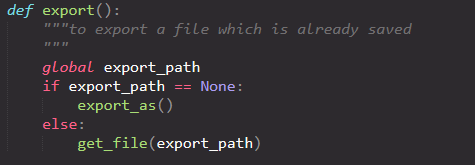
### Export

Args : None

Returns : None

Function : wrapper function to call get\_file()

Usage : export():



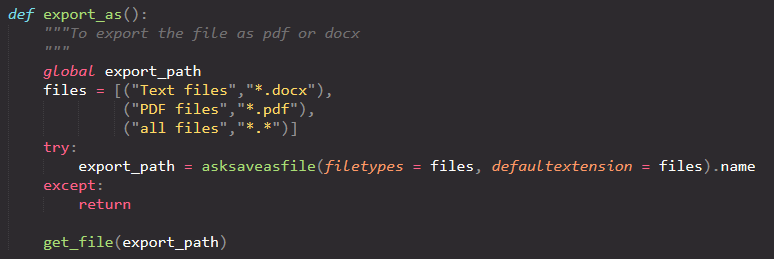
### Export\_as

Args : None

Returns : None

Function : wrapper function to call get\_file()

Usage : export\_as()



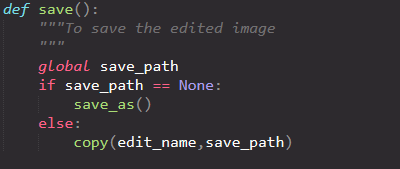
### Save

Args : None

Returns : None

Function : wrapper function to call save\_as with an alias

Usage : save()



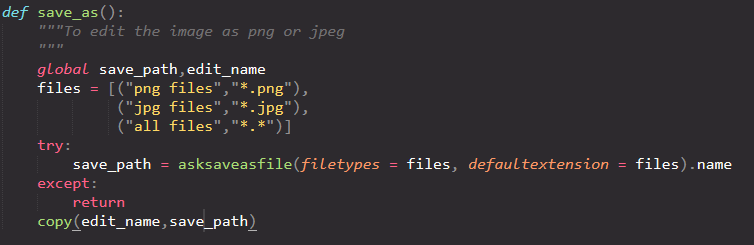
### Save\_as

Args : None

Returns : None

Function : to save the edited image using file dialog box

Usage : save\_as()



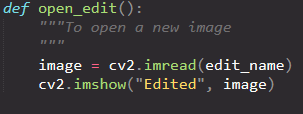
### Open\_edit

Args : None

Returns : None

Function : to show the preview of the edited image

Usage : open\_edit()



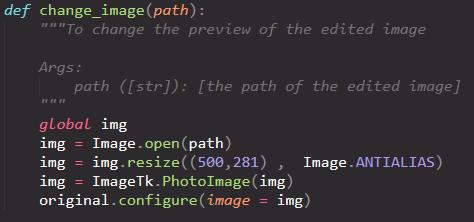
### Change\_image

Args : None

Returns : None

Function : to change the image in the tkinter application

Usage : change\_image()

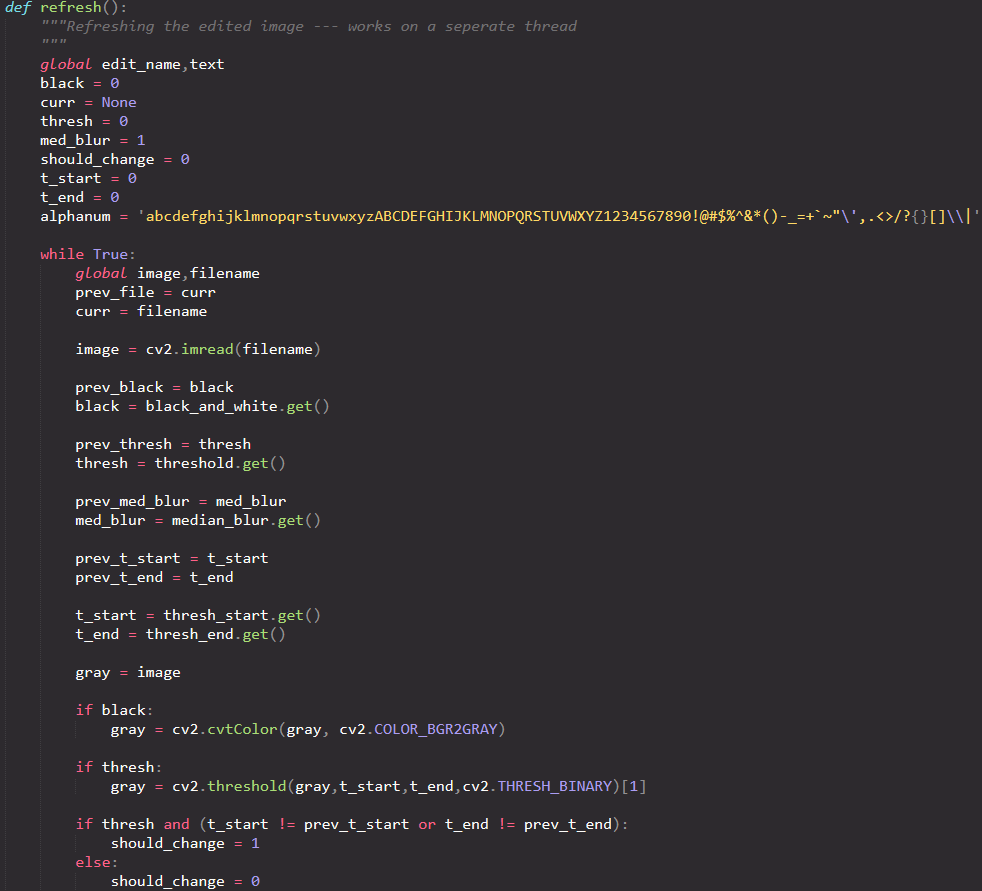


### Refresh

Args : None

Returns : None

Function : to refresh all the edits happened to the image



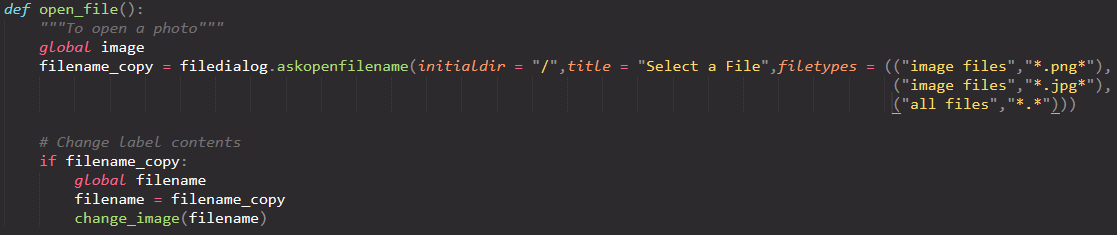


### Open\_file

Args : None

Returns : None

Function : Opens a dialog box to choose an image.wrapper for change\_image() function



## Threads

Main thread : to take care of all the UI widgets and to capture user input

Secondary thread : To take care of the refreshing of the image when edited

## UI Elements

### Buttons

Open edited : to open the preview of the edited image.

### Check boxes

Black and white : To make the image black and white

Threshold : to turn on the threshold for the image

### Sliders

Threshold start : to set the start of the threshold value

Threshold end : to set the end of threshold value

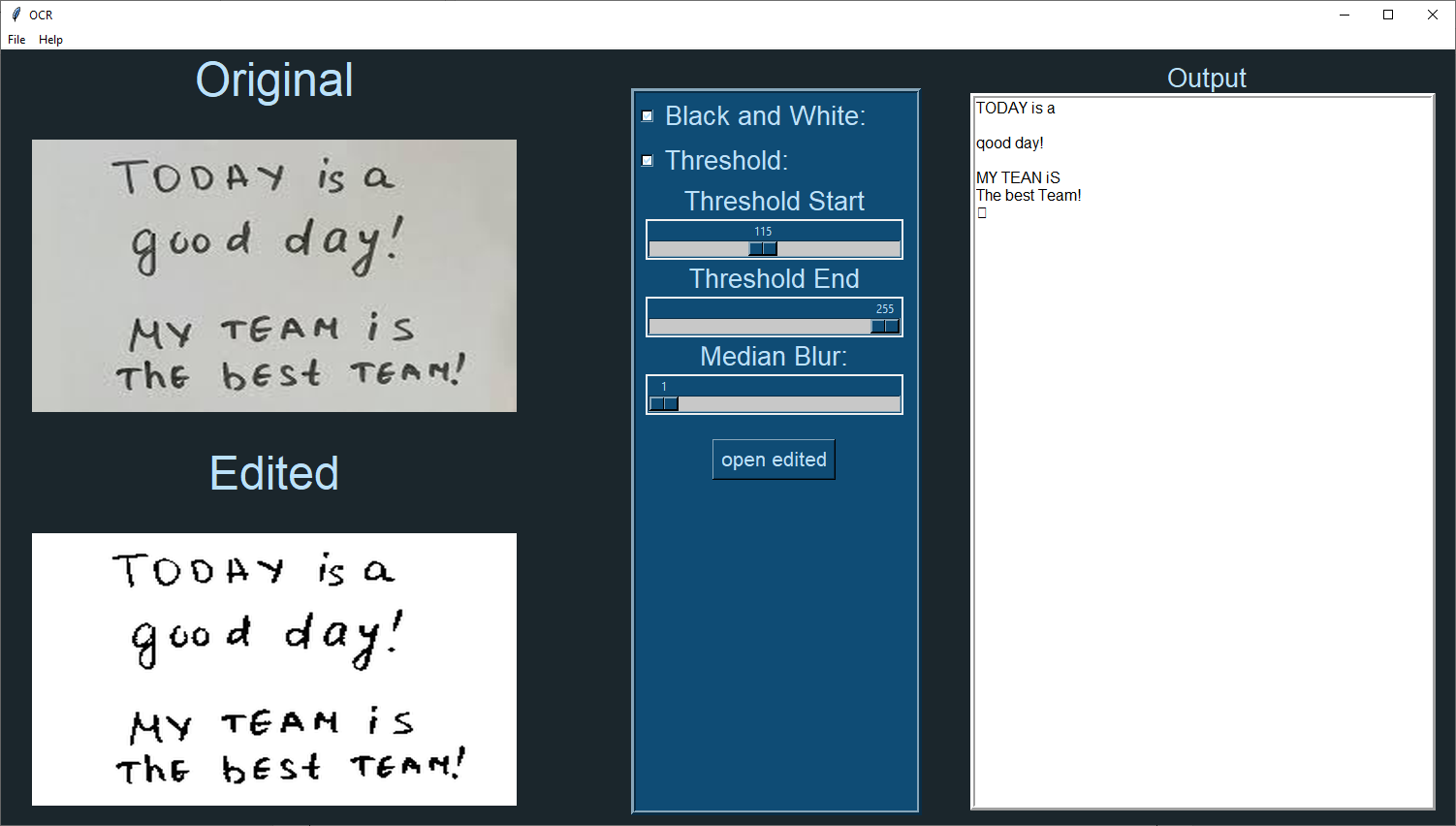
Medan blur : To set the amount of median blur

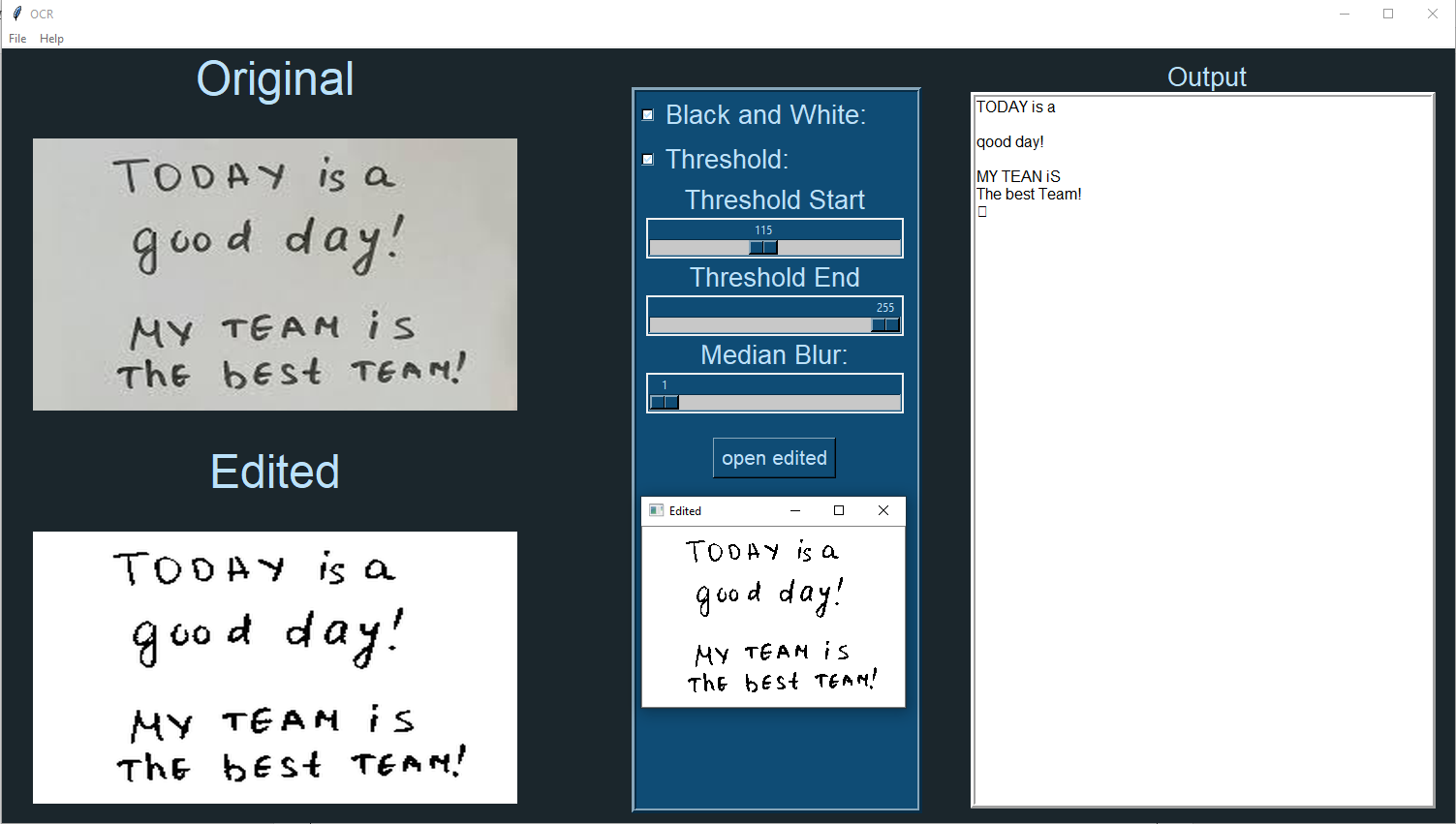
### Text box

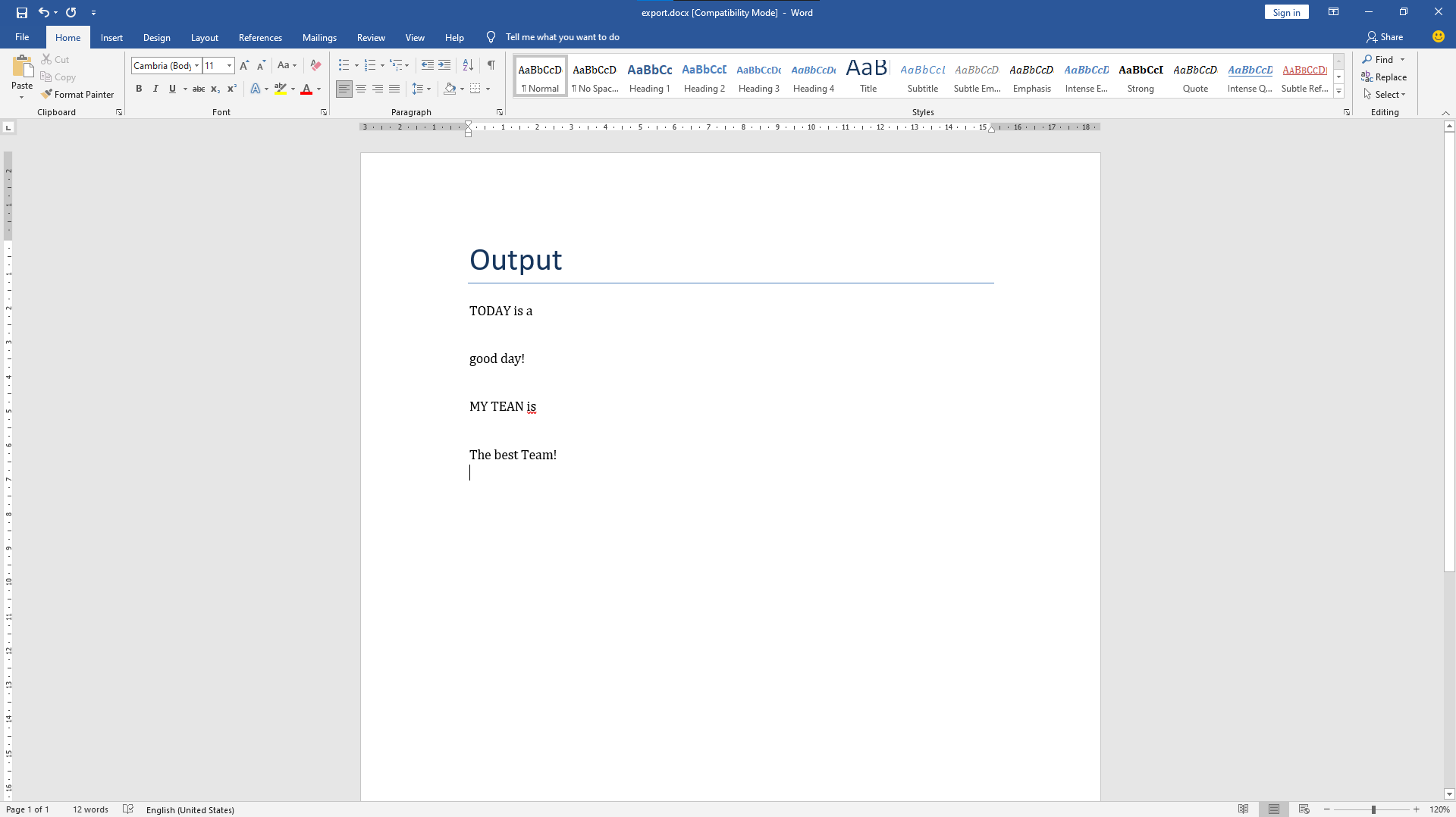
Output : To display the output of the recognized text.

## Execution

#### 







References

Tkinter :

[Tkinter documentation](https://docs.python.org/3/library/tk.html)

Pytesseract :

[Pytesseract documentation](https://pytesseract.readthedocs.io/en/latest/)

[Youtube video for pytesseract](https://www.youtube.com/watch?v=4DrCIVS5U3Y)

Open CV :

[Open CV docs](https://docs.opencv.org/master/d6/d00/tutorial_py_root.html)

[Youtube playlist on open CV](https://www.youtube.com/watch?v=Yrg96JQVT-M&list=PLuNAkIFfopwlMipNag-FWPFKQ1umLRO_-)

# Code Usage

# Contact details

Name : Prakash kumar R

College : Saveetha Engineering College

Phone : 7338942498

Email : prakash218kumar@gmail.com