PONDICHERRY UNIVERSITY

Public university in Pondicherry

WEB ACCESSIBILITY – CSCE843

(Softcore)

TASK-5 (OCR Study)

DHIVYA. K
21376005
PG Scholar
Master of Technology
Department of Computer Science and Engineering
II – Year & III – Semester
Pondicherry University

Date of Submission: 18/09/2022

Submitted to

Dr.K.S.Kuppusamy
Associate Professor
Department of Computer Science and Engineering
Pondicherry University

AGENTA

- **OCR**
- **Accuracy of OCR**
- **OCR Characters Details**
- **OCR working Process**
- **Advantages of Optical character Reader (OCR)**
- **Disadvantages of Optical character Reader (OCR)**
- **CASE STUDY:**
    1. **Tesseract OCR**
        - **Tesseract Installation Steps**
        - **Converting image file into text file using Tesseract OCR Tool**
    2. **Google Lens**
        - **Advantages of Google Lens**
        - **Disadvantages of Google Lens**
        - **Google Lens works**
        - **Converting image file into text file using Google Lens**
    3. **Online OCR tool (www.newocr.com)**
        - **Features**
        - **Input file formats**
        - **Output file formats**
        - **Converting image file into text file using online OCR**
- **Document Similarity:**
    - ✓ **Document Similarity in Python Coding for Tesseract software**
        - ❖ **OUTPUT RESULT AND SCREENSHOT: (For Tesseract OCR file similarity)**
    - ✓ **Document Similarity in Python Coding for Google Lens software**
        - ❖ **OUTPUT RESULT AND SCREENSHOT: (For Google Lens file similarity)**
    - ✓ **Document Similarity in Python Coding for online newocr website**
        - ❖ **OUTPUT RESULT AND SCREENSHOT: (For online OCR file similarity)**
- **Comparative analysis of OCR tools**
- **Conclusion**

OCR

- Optical character Reader (OCR) helps in differentiating presence of characters by their shapes.
- OCR is employed as data input device to read numeric and alphanumeric characters from pre-printed documents produced by printers, type-writers etc.

Accuracy of OCR

- Level of accuracy to scan characters accurately is based on how clear writing is.
- Scanners are improved to be ready to read different styles and sizes of text also as neat handwriting.
- There is utmost accuracy in the text scanned with OCR but then also there is the need to check as some letters are often misread.
- OCR is employed to automatically recognize postcodes on letters at sorting offices.

OCR Characters Details

- In OCR, scanner is provided with character recognition software which converts bitmap images of characters to equivalent ASCII codes.
- First step in whole process is to create bitmap of image of document then with help of software OCR translates the array of grid points into ASCII text which pc can understand and process it as letters, numbers and special characters.

OCR working Process

- OCR software process bitmap of every character and compares it with set of characters which machine has been programmed to acknowledge to translate bitmaps into text, Whichever character pattern it matches or nearly matches, is taken into account to be character read.

- If a scanned character doesn't match with any of the already stored character patterns, it's rejected.

- Its main focus is to convert Image document into Text Document

Advantages of Optical character Reader (OCR)

- Information of OCR can be readable with high degree of accuracy. Flatbed scanners are very accurate and may produce reasonably top-quality images.
- Processing of OCR information is fast. Large quantities of text are often input quickly.
- A paper-based form is often becoming an electronic form which is straightforward to store or send by mail.
- It is cheaper than paying someone amount to manually enter great deal of text data. Moreover, it takes less time to convert within the electronic form.
- The latest software can re-create tables also as original layout.
- This process is much faster as compared to the manual typing the information into the system
- Advanced version can even Recreate tables, columns and even produce sites.

Disadvantages of Optical character Reader (OCR)

- OCR text works efficiently with the printed text only and not with handwritten text. Handwriting must be learnt by the pc.
- OCR systems are expensive.
- There is the need of lot of space required by the image produced.
- The quality of the image can be lost during this process.
- Quality of the ultimate image depends on quality of the first image.
- All the documents got to be checked over carefully then manually corrected.
- Not 100% accurate, there are likely to be some mistakes made during the method.
- Not worth doing for little amounts of text.

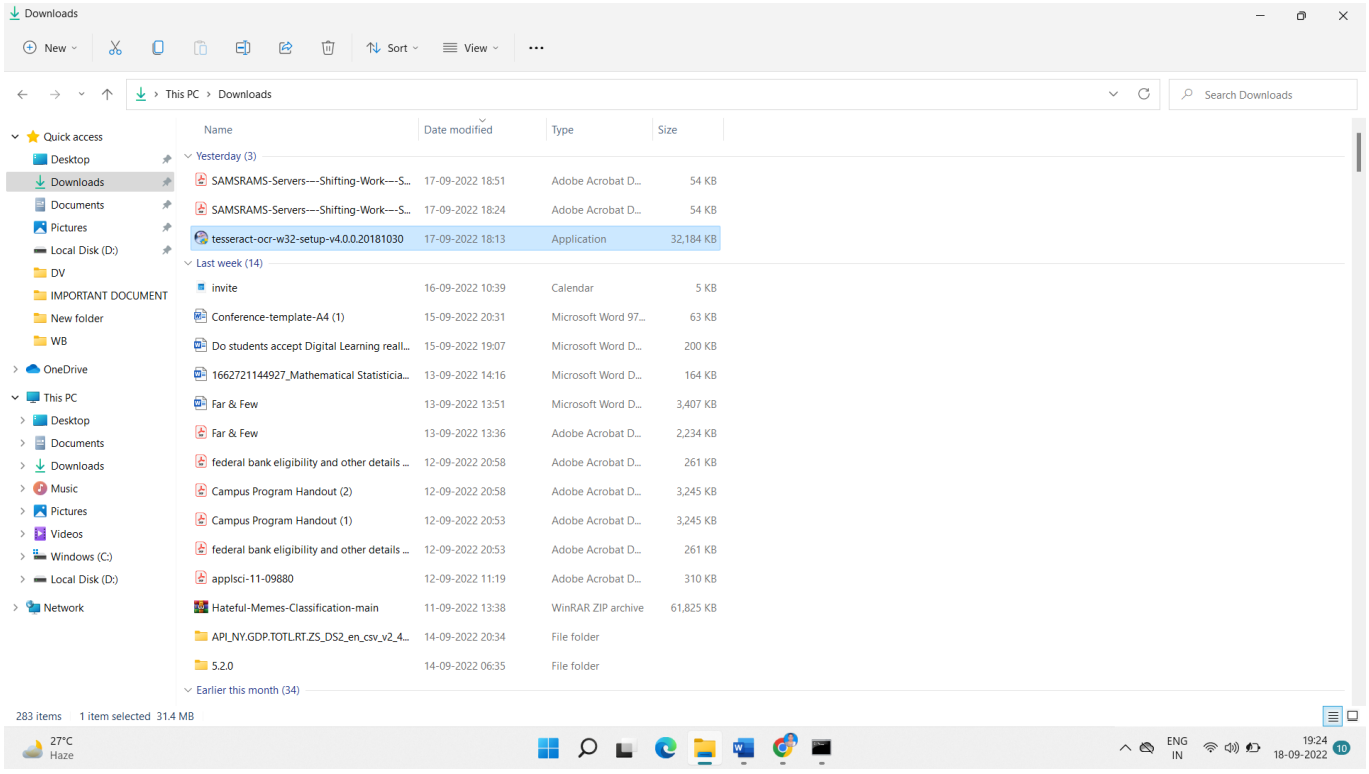
CASE STUDY: (3 CASE STUDY)

1. Tesseract OCR

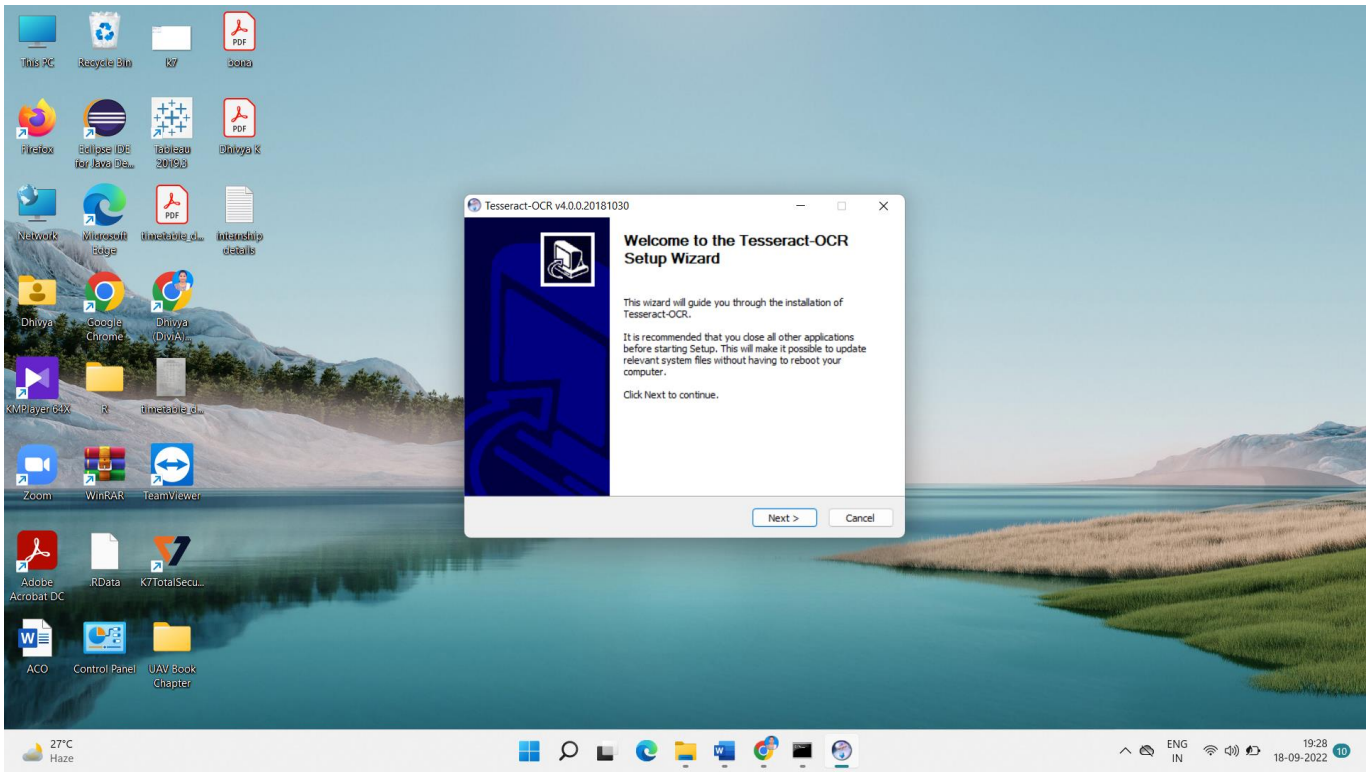
- Tesseract stand for optical character recognition engine.
- It is most popular open-source OCR tool
- It is released under Apache License
- It is originally developed by Hewlett-Packard as proprietary software in 1980 and released as open-source software in 2005
- Its development has been sponsored by Google since 2006

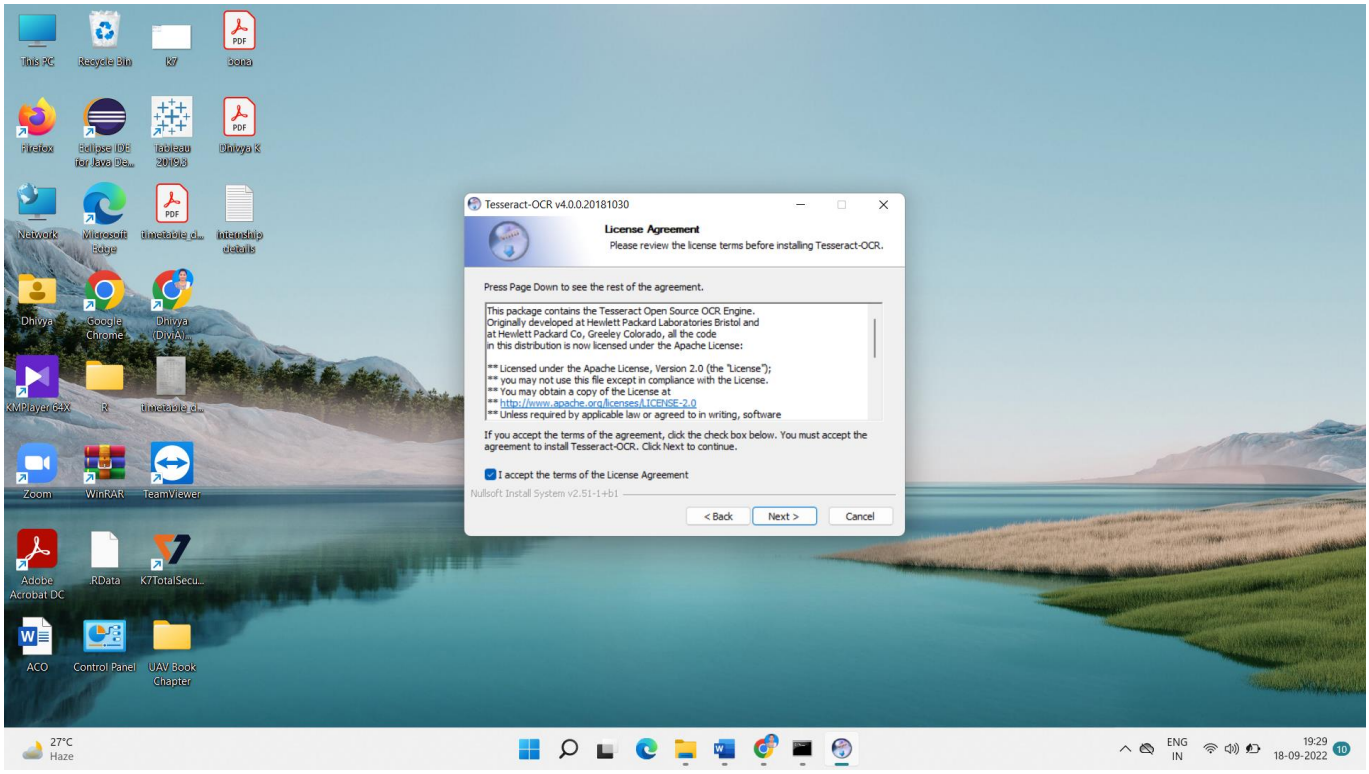
Tesseract Installation Steps

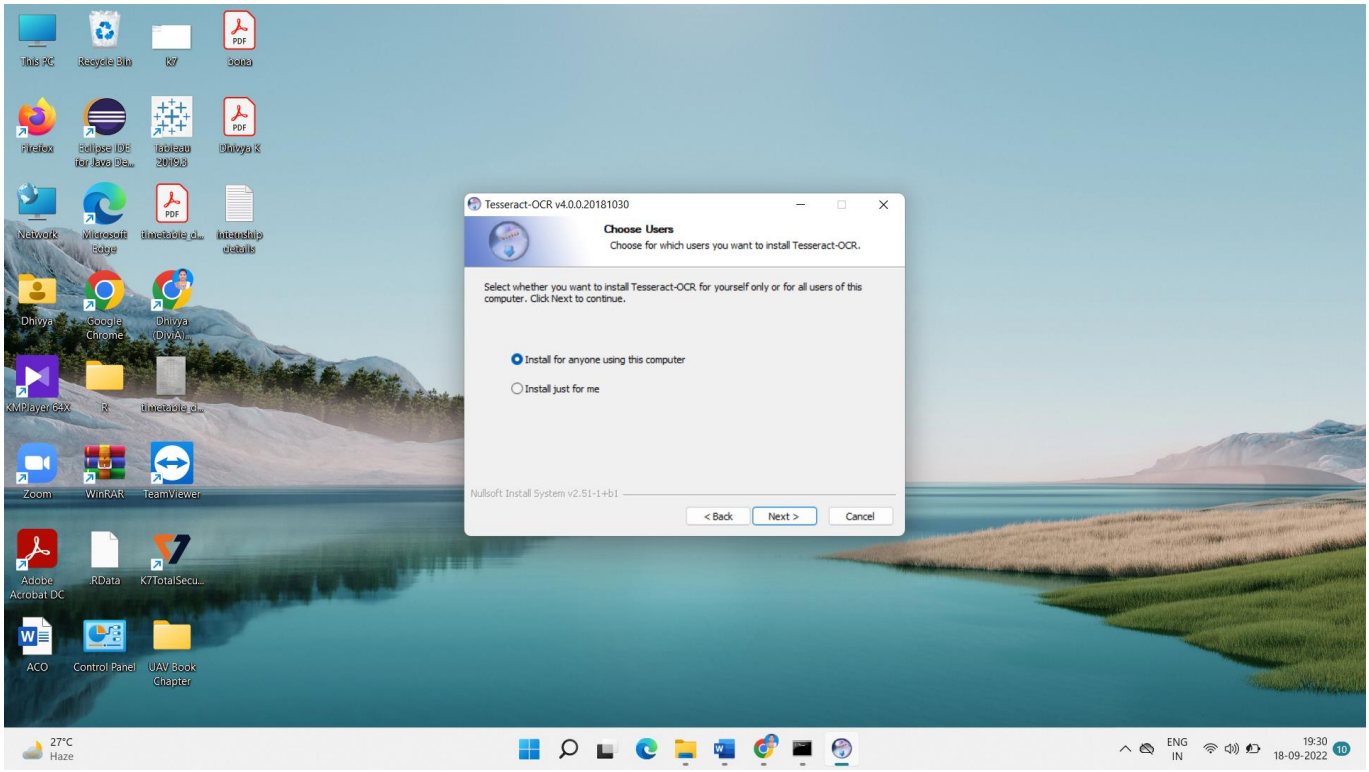
Step 1 – Download and install from the link tesseract-ocr-w64-setup-v4.0.0.20181030.exe (now in downloads)

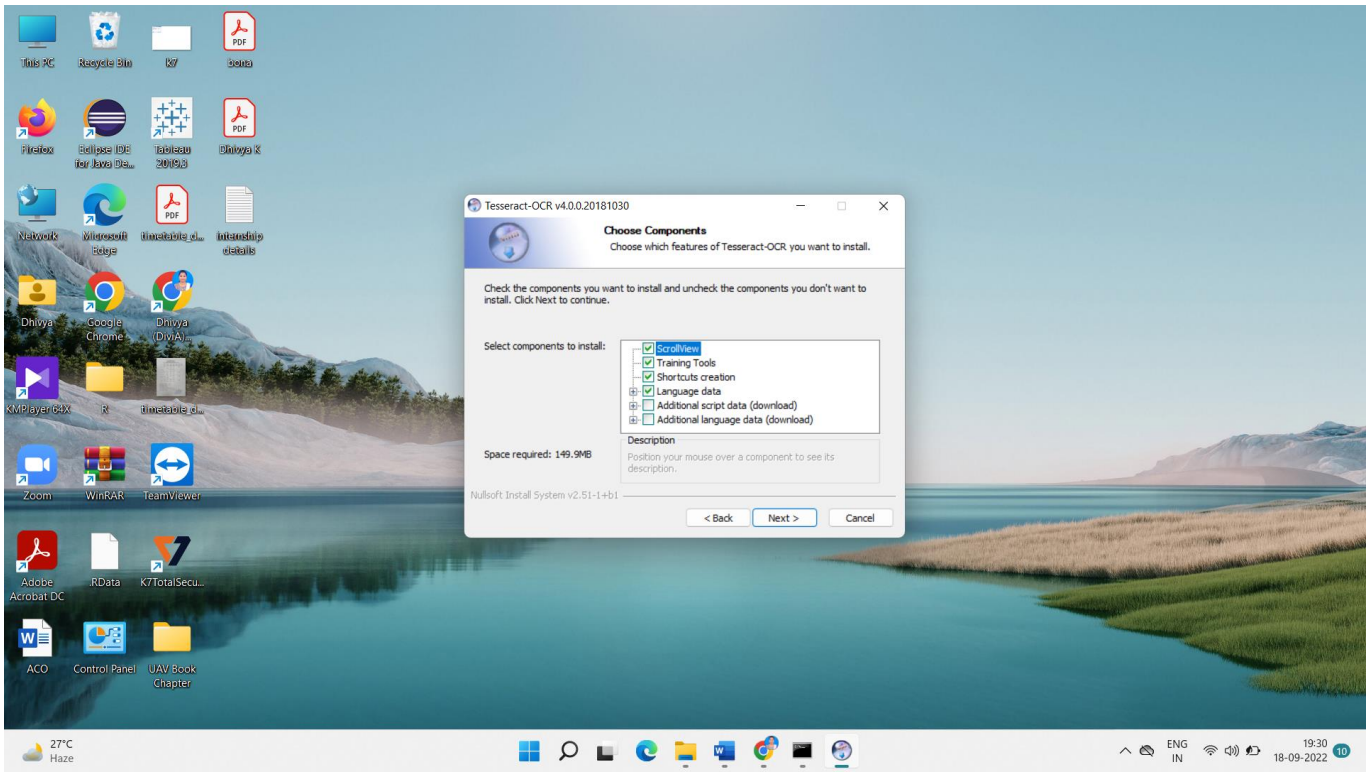


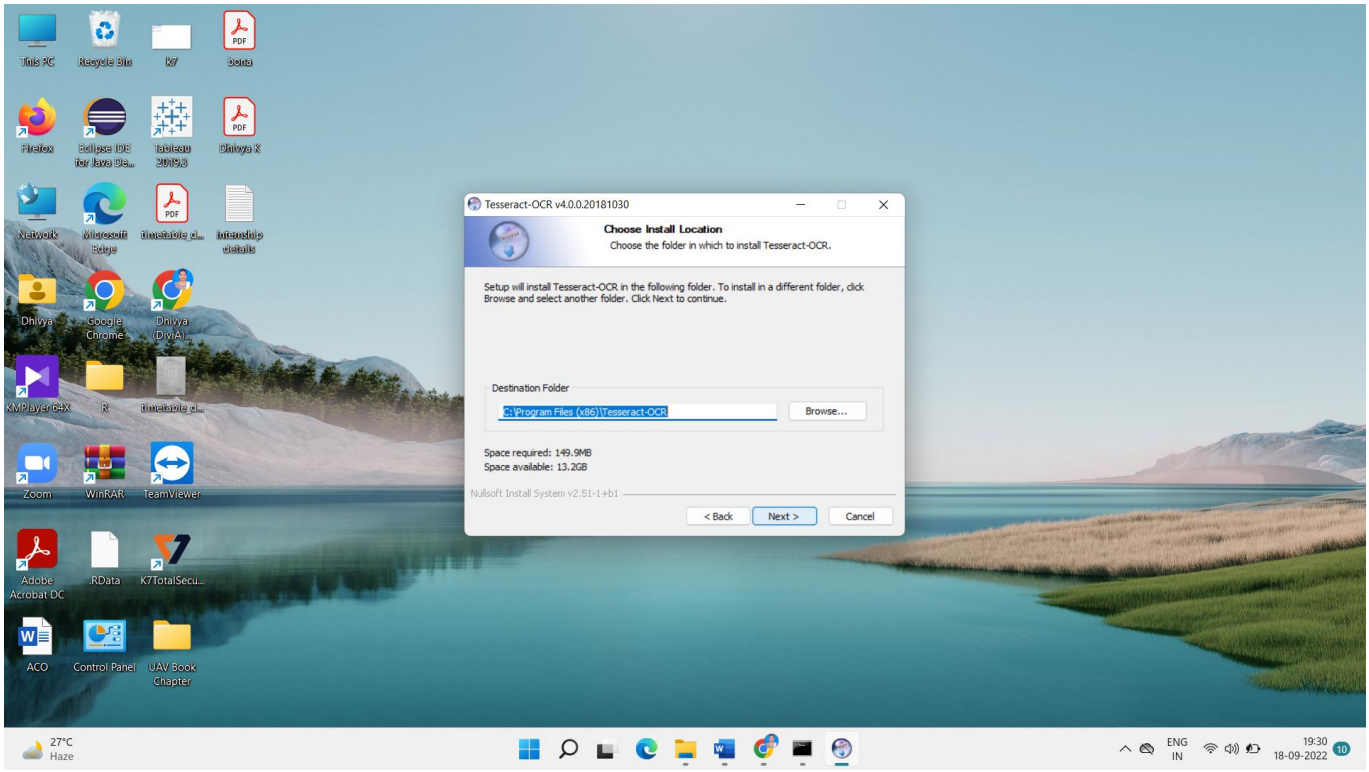
Step 2 – Double click and start installation process

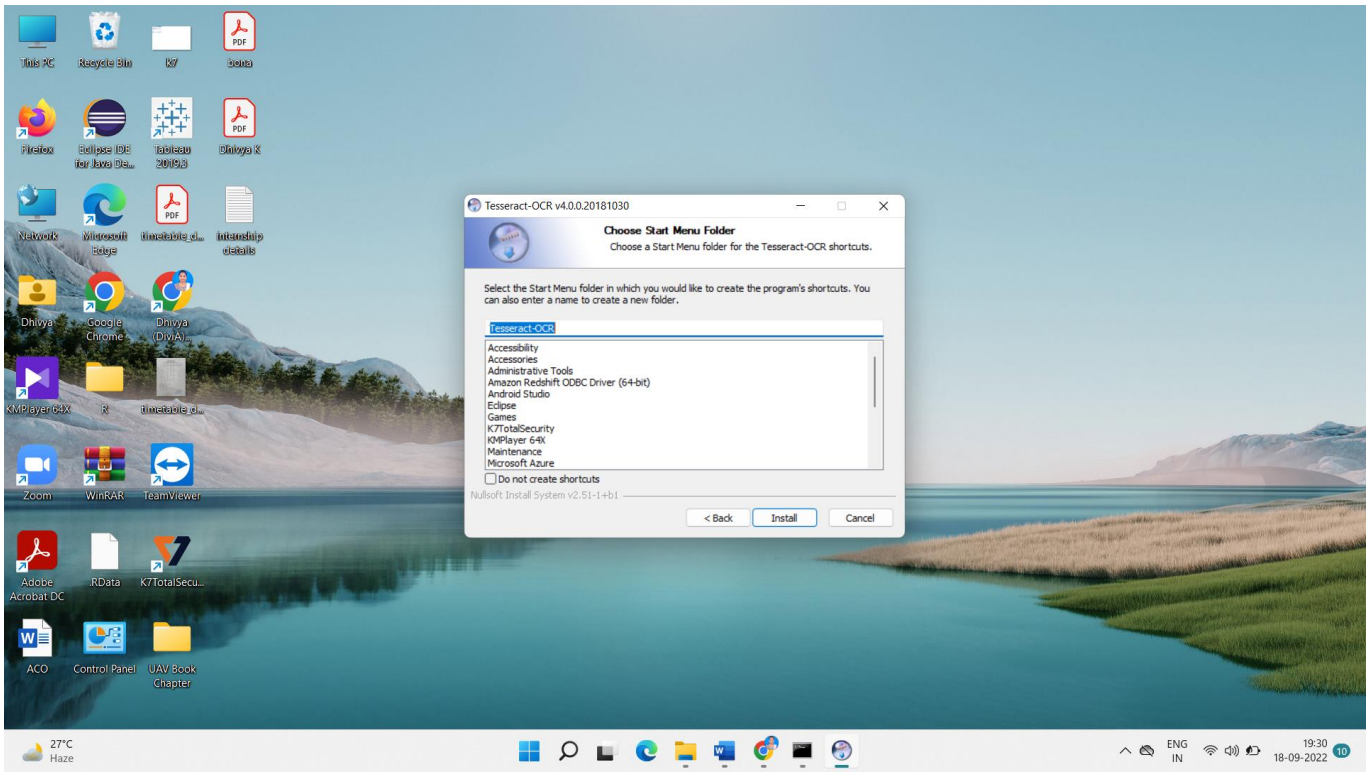


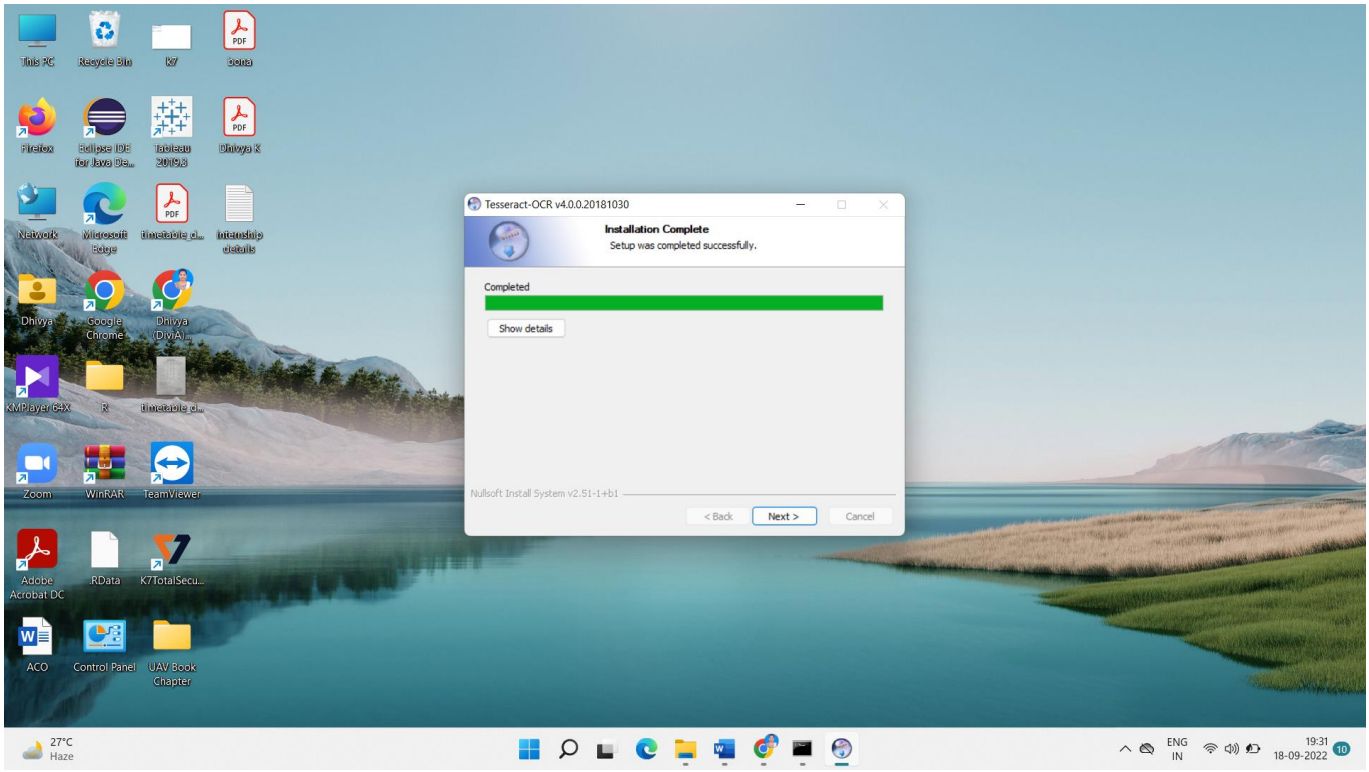


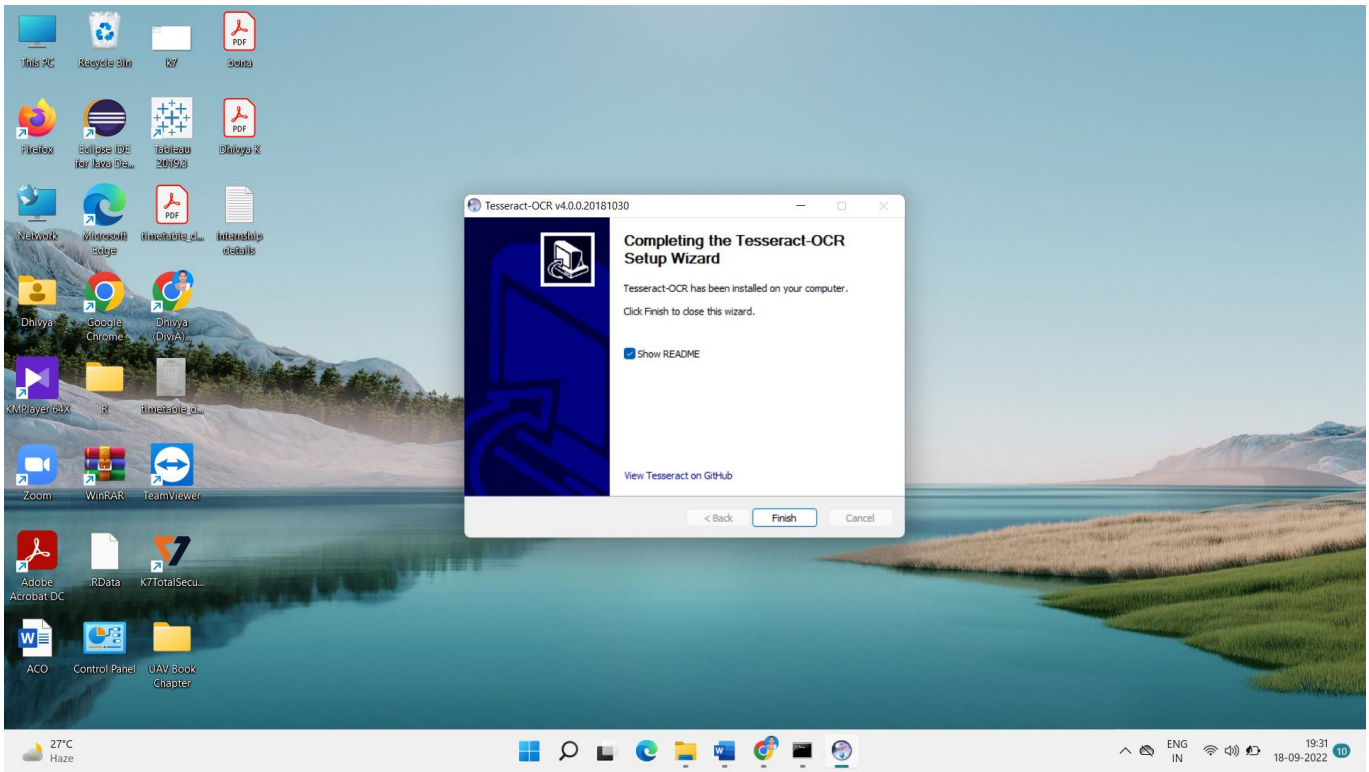

This PC
Recycle Bin
K7
Beuta
Firefox
Eclipse IDE for Java Da...
Tableau 2019.3
Dhivya K
Network
Microsoft Edge
timetable_d...
Internship details
Dhivya
Google Chrome
Dhivya (DivA)...
KMPlayer 64X
R
timetable_d...
Zoom
WinRAR
TeamViewer
Adobe Acrobat DC
.RData
K7TotalSecu...
ACO
Control Panel
UAV Book Chapter
Tesseract-OCR v4.0.0.20181030
Choose Users
Choose for which users you want to install Tesseract-OCR.
Select whether you want to install Tesseract-OCR for yourself only or for all users of this computer. Click Next to continue.
Install for anyone using this computer
Install just for me
Nullsoft Install System v2.51-1+b1
< Back
Next >
Cancel
27°C
Haze
ENG
IN
19:30
18-09-2022


This PC
Recycle Bin
K7
Beuta
Firefox
Eclipse IDE for Java Da...
Tableau 2019.3
Dhivya K
Network
Microsoft Edge
timetable_d...
Internship details
Dhivya
Google Chrome
Dhivya (DivA)...
KMPlayer 64X
R
timetable_d...
Zoom
WinRAR
TeamViewer
Adobe Acrobat DC
.RData
K7TotalSecu...
ACO
Control Panel
UAV Book Chapter
Tesseract-OCR v4.0.0.20181030
Choose Components
Choose which features of Tesseract-OCR you want to install.
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.
Select components to install:
ScrollView
Training Tools
Shortcuts creation
Language data
Additional script data (download)
Additional language data (download)
Description
Position your mouse over a component to see its description.
Space required: 149.9MB
Nullsoft Install System v2.51-1+b1
< Back
Next >
Cancel
27°C
Haze
ENG
IN
19:30
18-09-2022


Tesseract-OCR v4.0.0.20181030
Choose Install Location
Choose the folder in which to install Tesseract-OCR.
Setup will install Tesseract-OCR in the following folder. To install in a different folder, click Browse and select another folder. Click Next to continue.
Destination Folder
C:\Program Files (x86)\Tesseract-OCR
Browse...
Space required: 149.9MB
Space available: 13.2GB
Nullsoft Install System v2.51-1+b1
< Back
Next >
Cancel


Tesseract-OCR v4.0.0.20181030
Choose Start Menu Folder
Choose a Start Menu folder for the Tesseract-OCR shortcuts.
Select the Start Menu folder in which you would like to create the program's shortcuts. You can also enter a name to create a new folder.
Tesseract-OCR
Accessibility
Accessories
Administrative Tools
Amazon Redshift ODBC Driver (64-bit)
Android Studio
Eclipse
Games
K7TotalSecurity
KMPlayer 64X
Maintenance
Microsoft Azure
Do not create shortcuts
Nullsoft Install System v2.51-1+b1
< Back
Install
Cancel


Tesseract-OCR v4.0.0.20181030
Installation Complete
Setup was completed successfully.
Completed
Show details
Nullsoft Install System v2.51-1+b1
< Back
Next >
Cancel

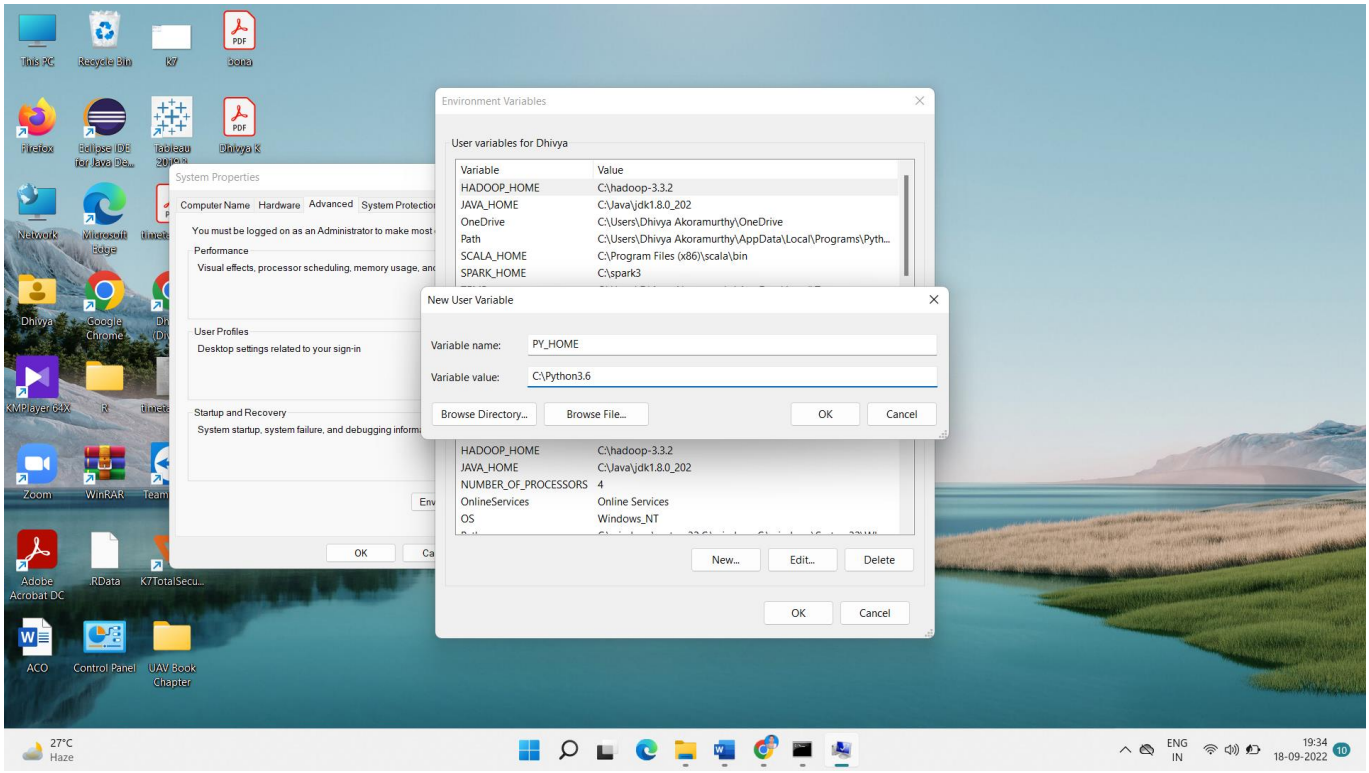

Tesseract-OCR v4.0.0.20181030
Completing the Tesseract-OCR Setup Wizard
Tesseract-OCR has been installed on your computer.
Click Finish to close this wizard.
Show README
View Tesseract on GitHub
< Back
Finish
Cancel

Step 3 - In the System Environment Variables, add Python installation path as shown below

In Windows Search Bar, type “system environment variables” –> click open –> Click Environment Variables button at bottom, Environment Variables tab will appear

Variable Name –PY_HOME

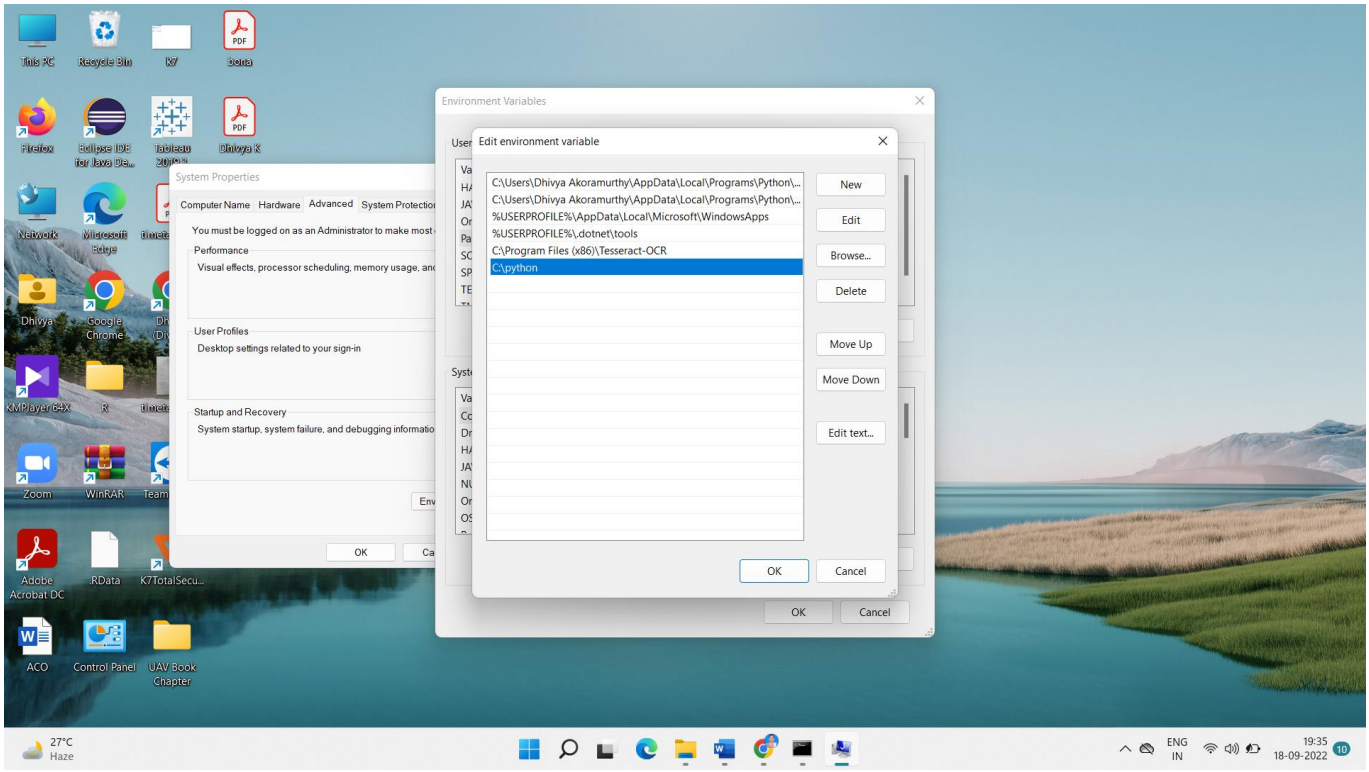
Variable Value –C:\Python3.6



Step 4 – Then in the System Environment Variables, add Python Path as shown below:

Variable Name –PYTHONPATH

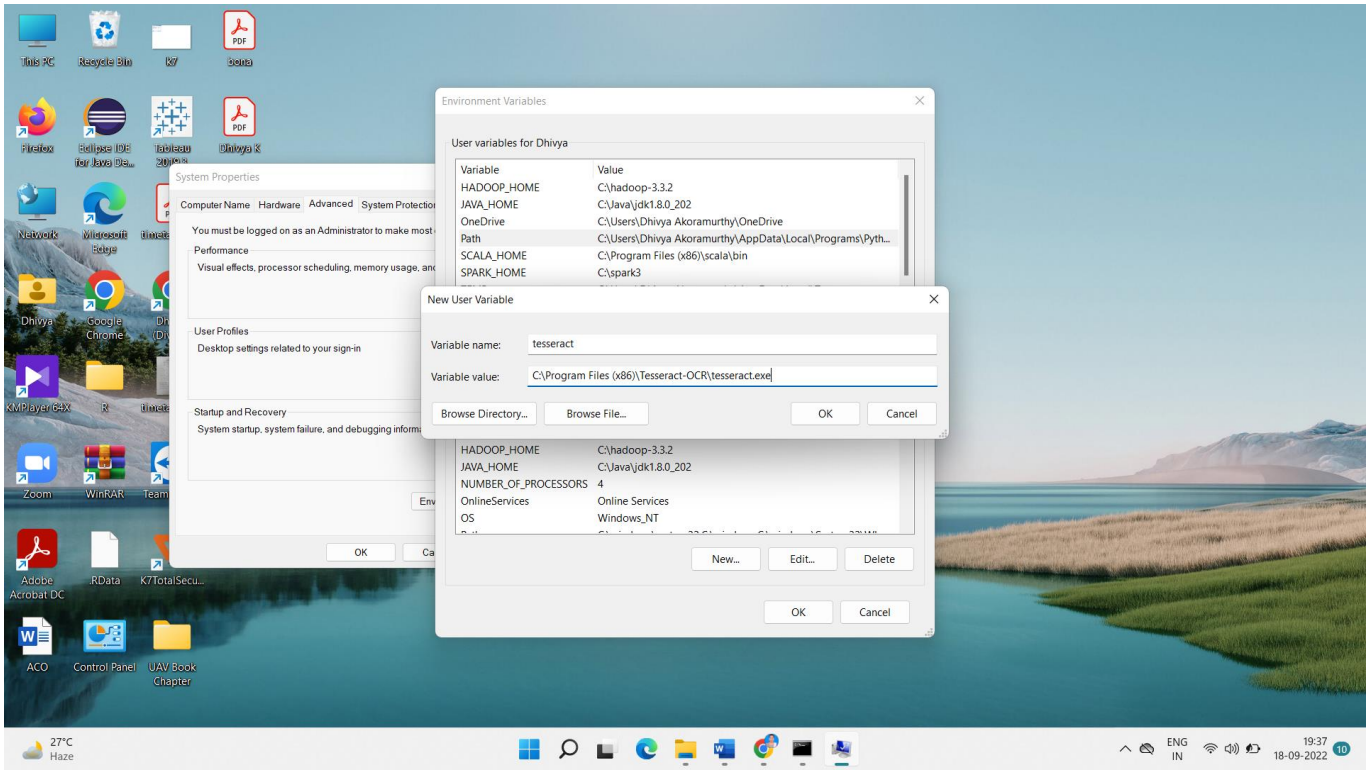
Variable Value –%PY_HOME%\Lib;%PY_HOME%\DLLs;%PY_HOME%\Lib\lib-tk;C:\another-library



Step 5 – Add another environment variable “tesseract”

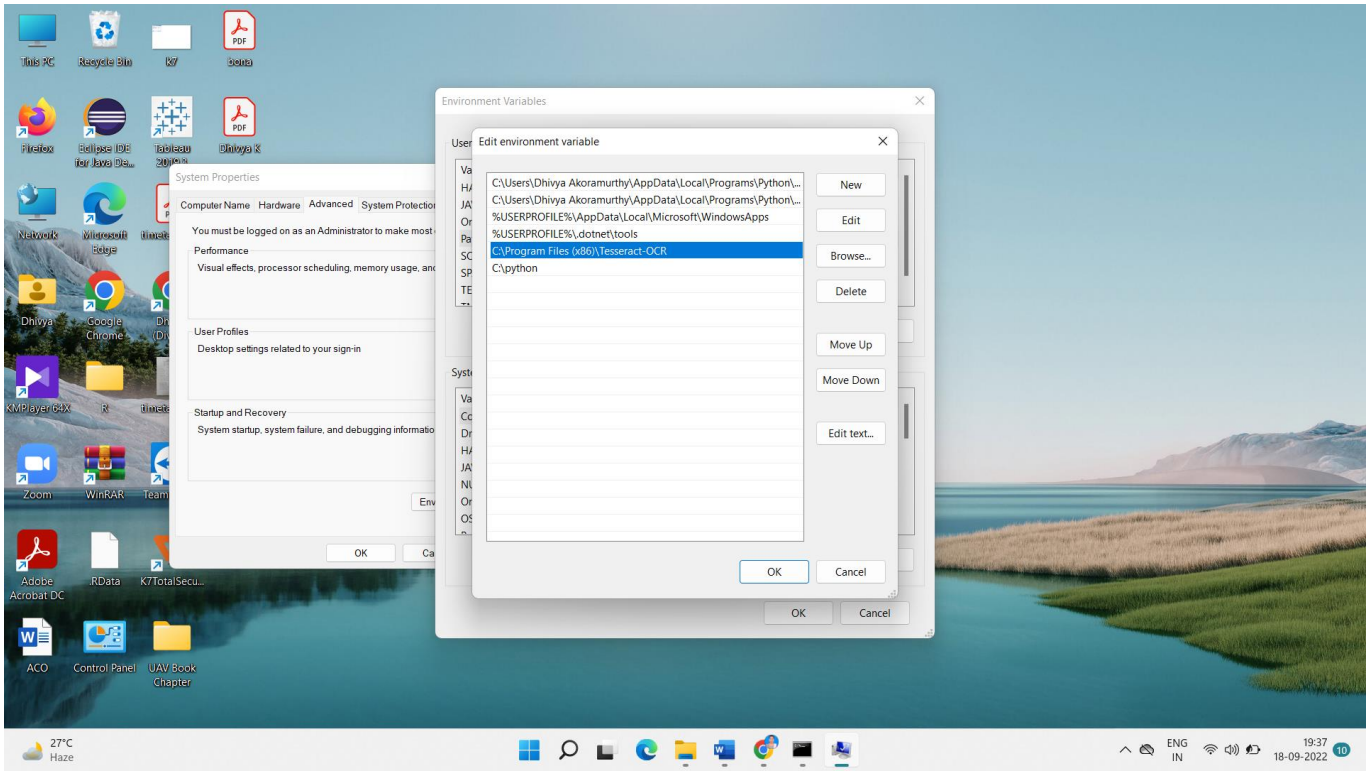
Variable Name –tesseract

Variable Value –C:\Program Files (x86)\Tesseract-OCR\tesseract.exe

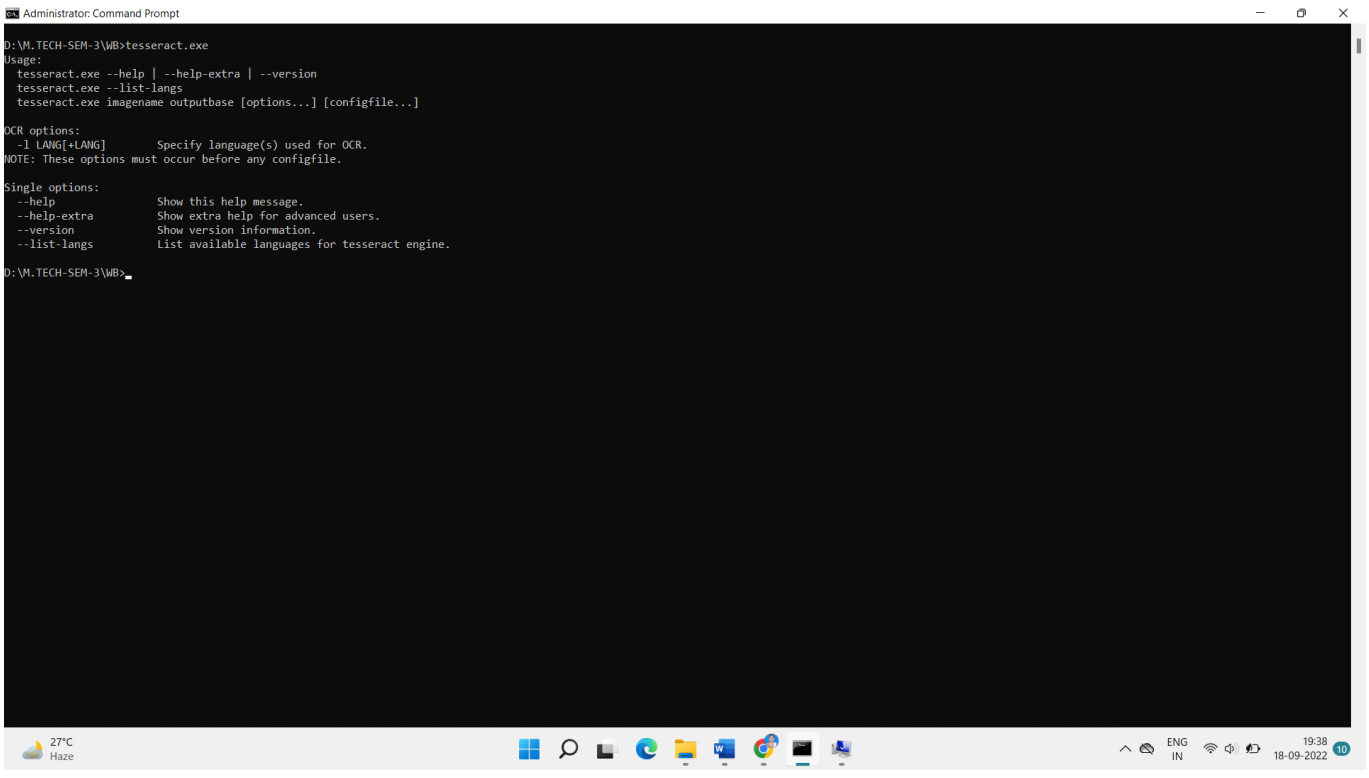


Step 6 – In the PATH environment variable add following path of installation of tesseract

Variable Value –C:\Program Files (x86)\Tesseract-OCR

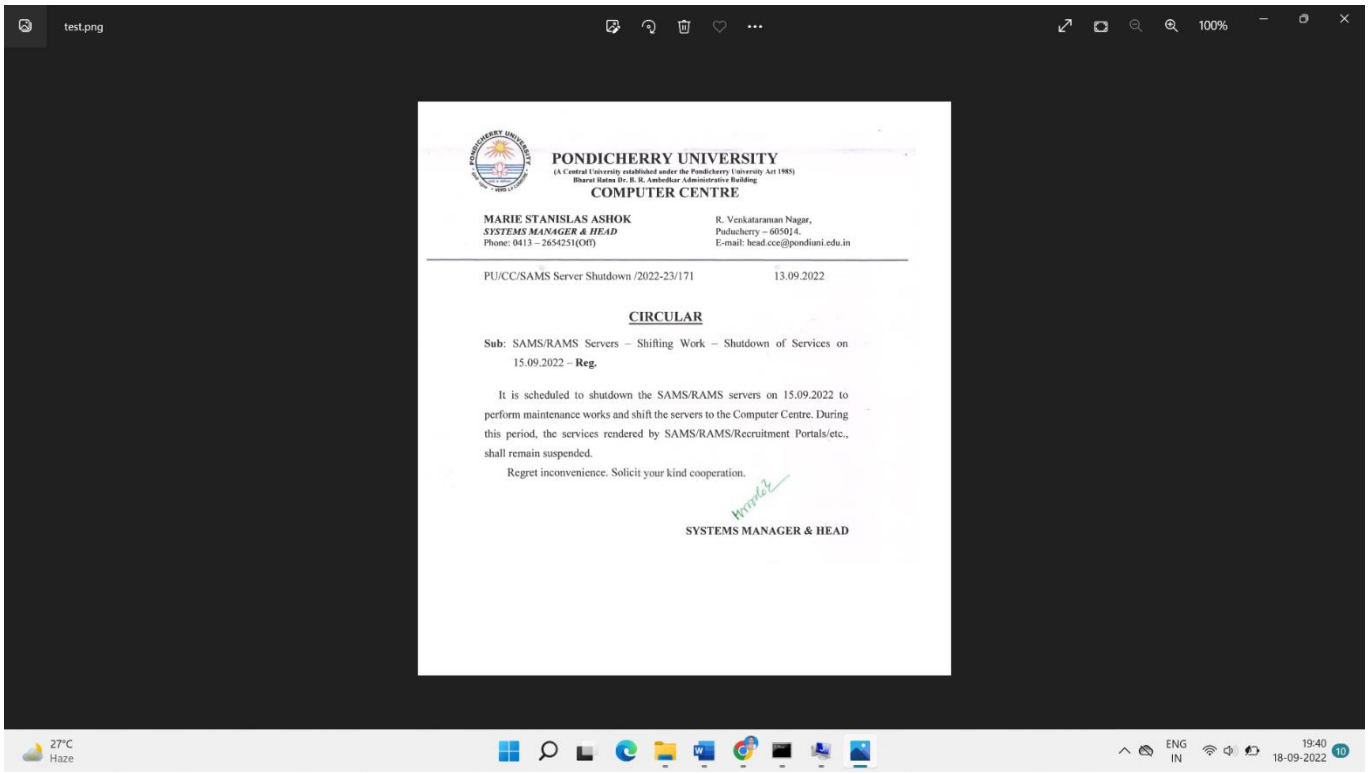


Step 7 – Successfully installed and checked in Command Prompt (CMD)



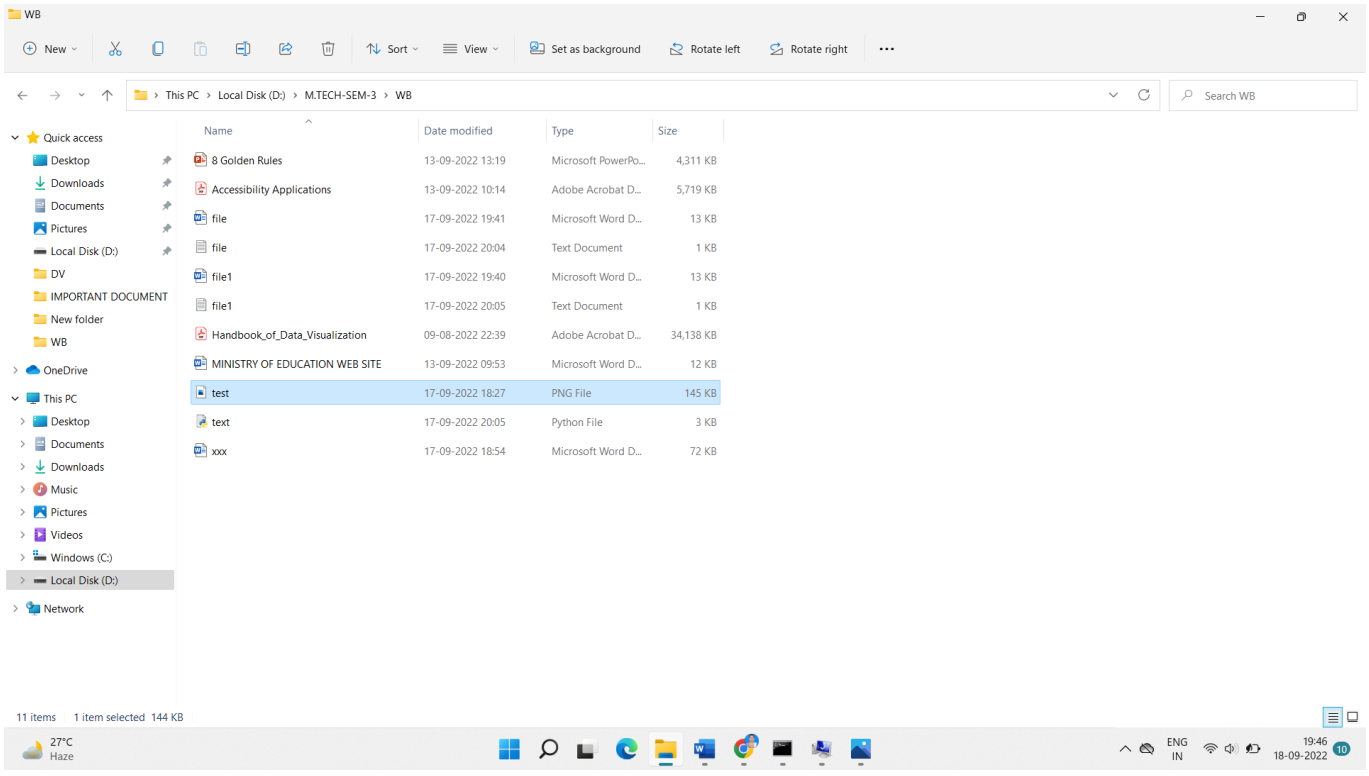
Converting image file into text file using Tesseract OCR Tool

Step 1: File name “test.PNG” (Input File)



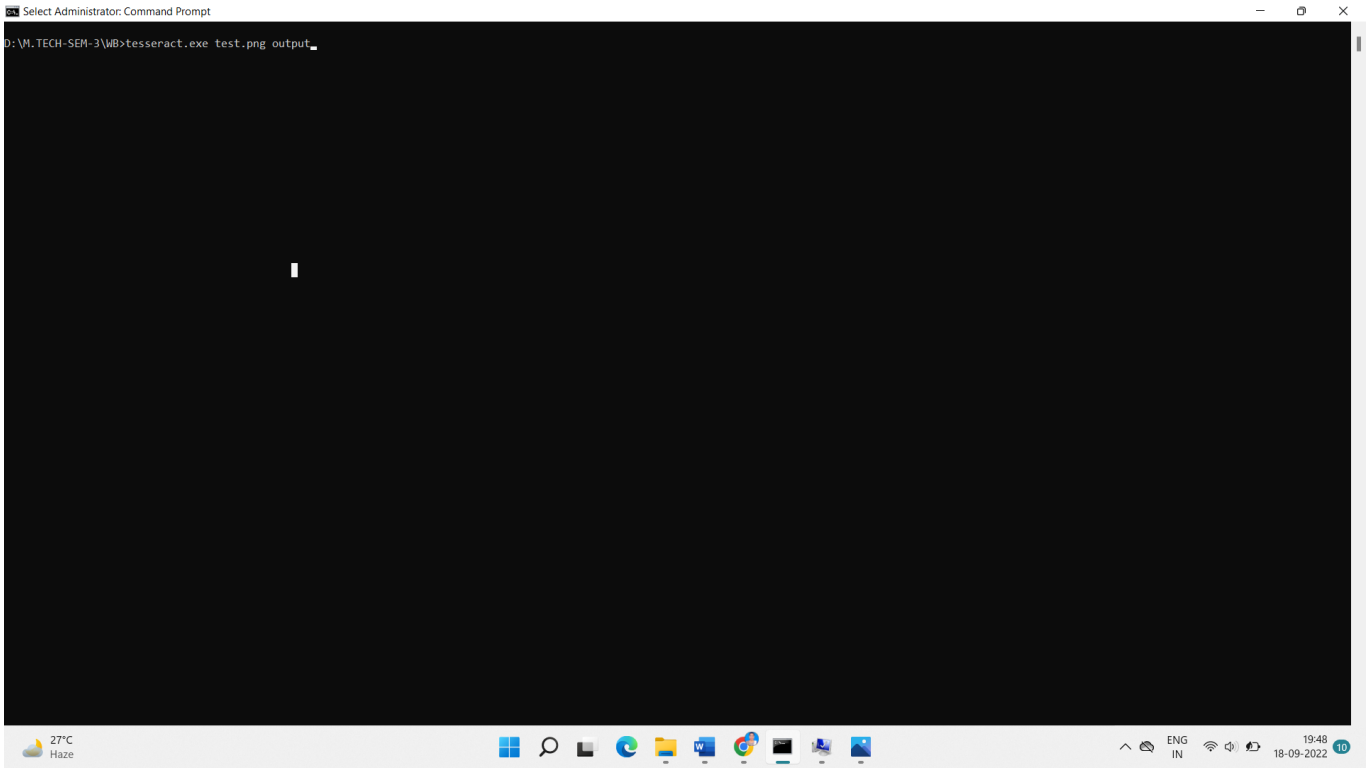
Step-2: Input file loaction

Input file name: test.png (already in my local drive)

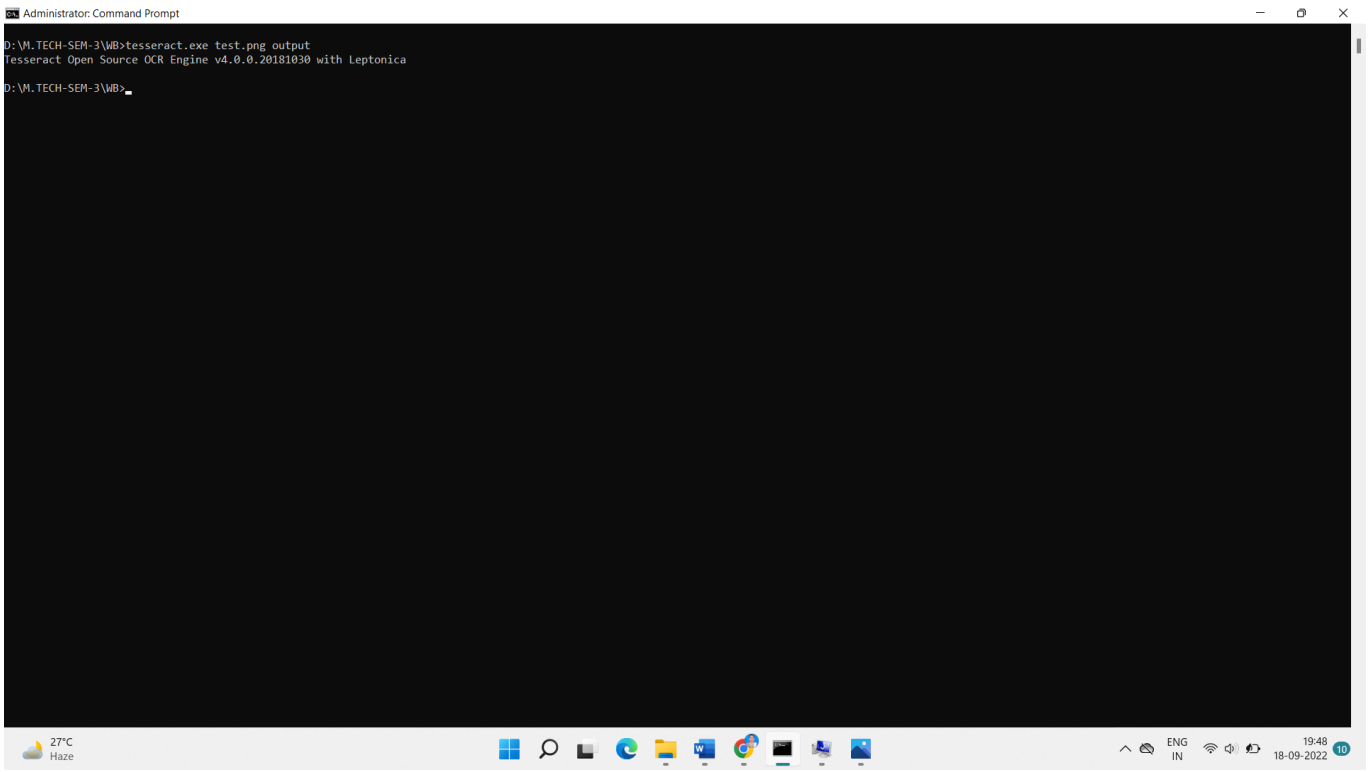


Step 3: Open CMD and type “tesseract.exe test.png output”

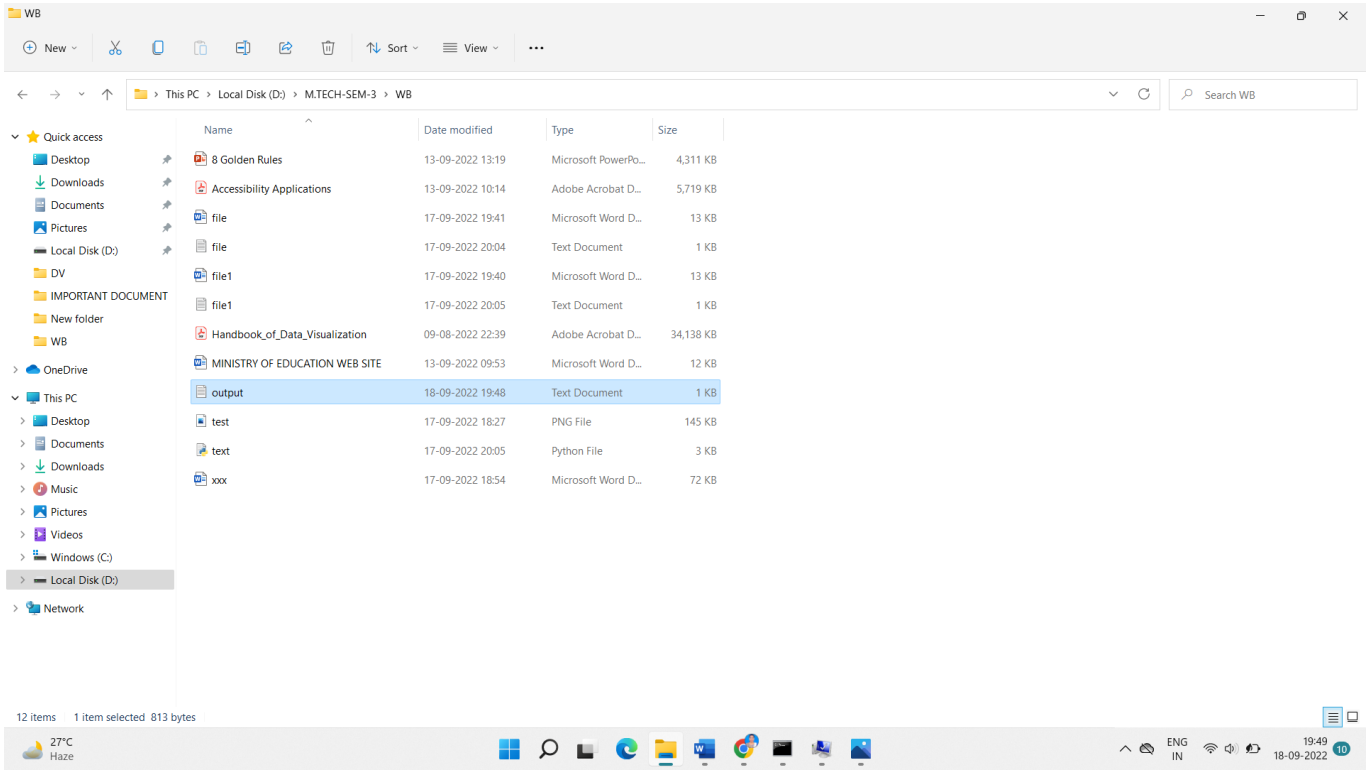
Output file name: output



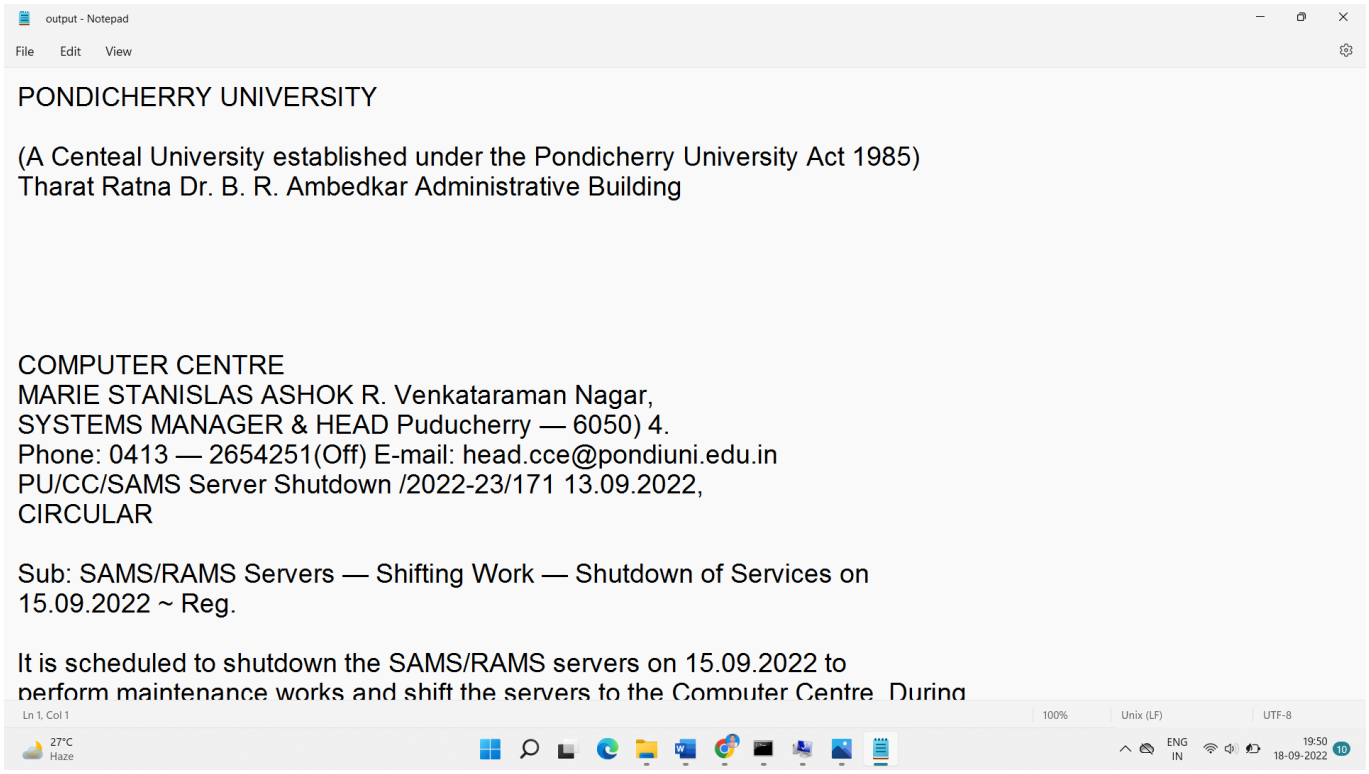
After click enter(run OCR tool)



Step 4: Now output file is created by OCR tool (output.txt)



Step 5: After applying OCR tool is converted into



2. Google Lens

- Google Lens is an image recognition technology developed by Google.
- It is released in October 04, 2017
- OS used for Google Lens are Android, iOS (Google Mobile), Microsoft Windows, macOs (Google Chrome, only for searching images) and web
- It can translate text into most Google Translate languages.
- It also can search images by searching, text, or translate.
- Shopping results are available in Austria, Australia, Belgium, Brazil, Canada, Chile, Colombia, Czech Republic, Denmark, France, Germany, India, Indonesia, Ireland, Italy, Japan, Malaysia, Mexico, Netherlands, New Zealand, Norway, Philippines, Poland, Portugal, Russia, Singapore, South Africa, South Korea, Spain, Sweden, Switzerland, Turkey, UAE, United Kingdom, and the United States.

Advantages of Google Lens

- It can use it to copy text easily
- It can find similar products with Google Lens
- Gather Quick Information, Especially Books
- Easily Open Links
- It Can Get Reviews on Venues
- It Can Identify Plants and Animals
- It Can Discover Good Music with Google Lens
- It Can Scan QR Codes

Disadvantages of Google Lens

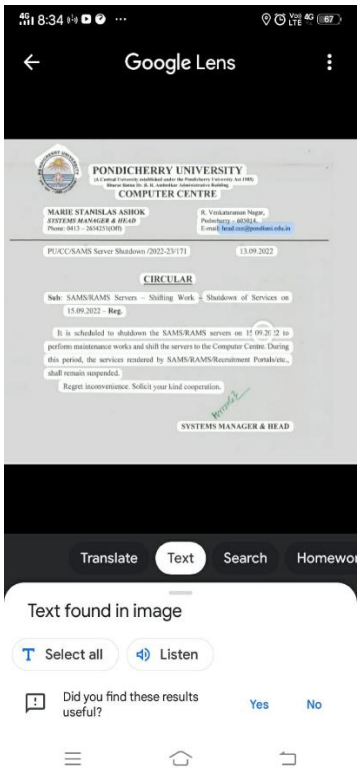
- Google Lens will collect information from you
- Not 100% Accurate

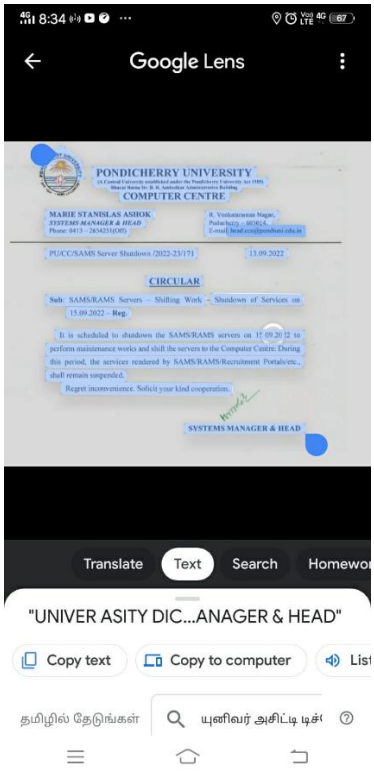
Google Lens works

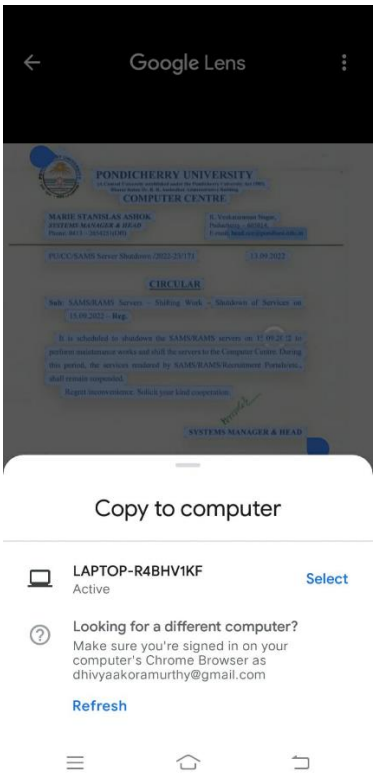
- Google Lens is best described as a search engine for the real world.
- It uses artificial intelligence to identify text and objects both within images and in a live view from your phone's camera, and it then lets you learn about and interact with those elements in all sorts of interesting ways.

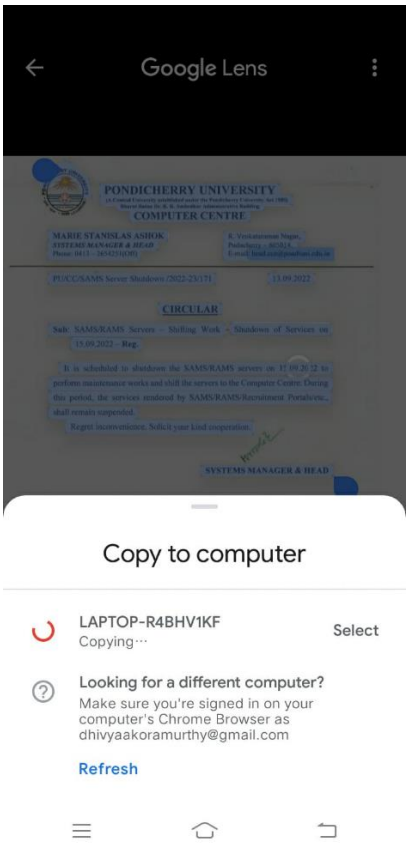
Converting image file into text file using Google Lens

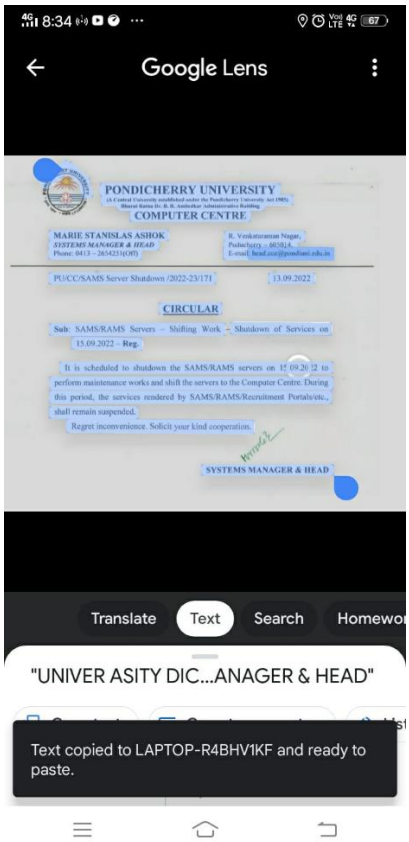
Step 1: Open Google Lens and select the image file and copy the text into computer screen

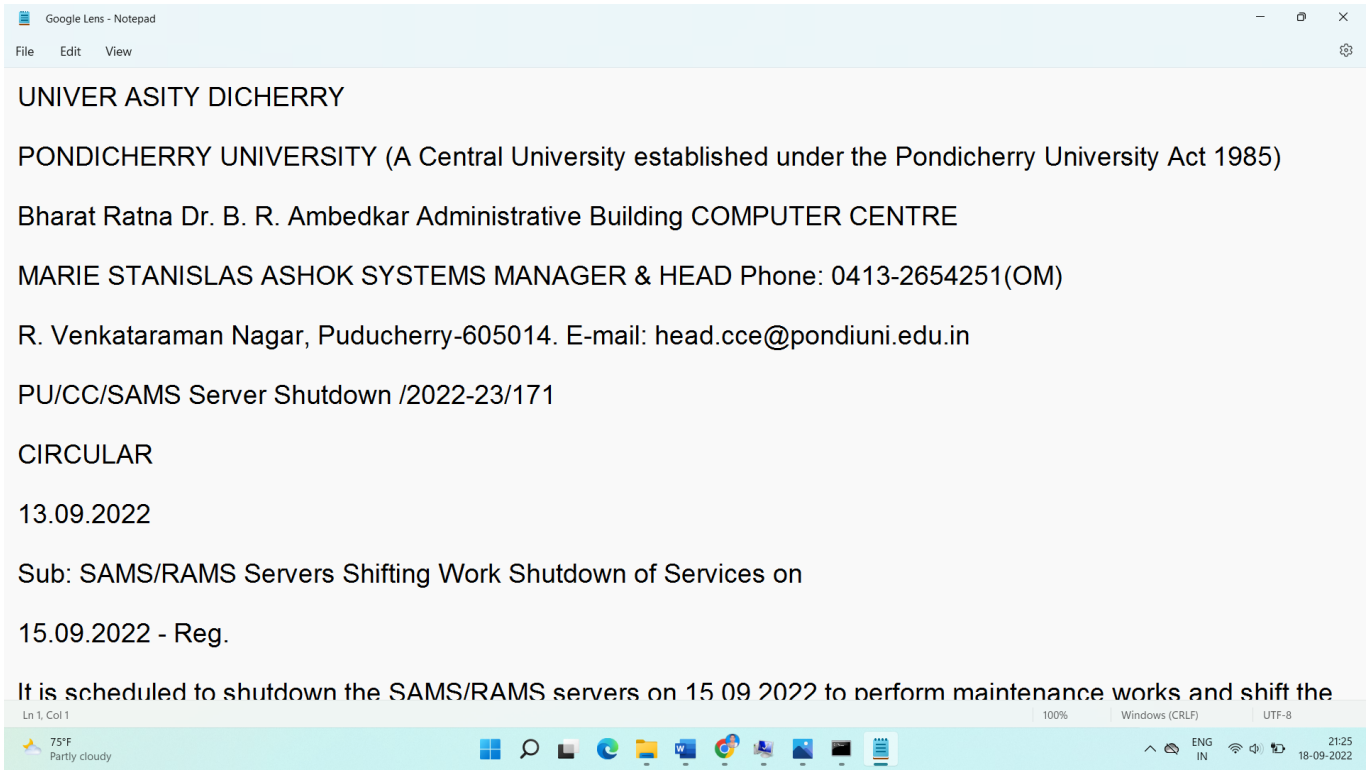











Google Lens - Notepad
File Edit View
UNIVER ASITY DICHERRY
PONDICHERRY UNIVERSITY (A Central University established under the Pondicherry University Act 1985)
Bharat Ratna Dr. B. R. Ambedkar Administrative Building COMPUTER CENTRE
MARIE STANISLAS ASHOK SYSTEMS MANAGER & HEAD Phone: 0413-2654251(OM)
R. Venkataraman Nagar, Puducherry-605014. E-mail: head.cce@pondiuni.edu.in
PU/CC/SAMS Server Shutdown /2022-23/171
CIRCULAR
13.09.2022
Sub: SAMS/RAMS Servers Shifting Work Shutdown of Services on
15.09.2022 - Reg.
It is scheduled to shutdown the SAMS/RAMS servers on 15.09.2022 to perform maintenance works and shift the
Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
75°F
Partly cloudy
ENG
IN
21:25
18-09-2022

3. Online OCR tool (www.newocr.com)

- NewOCR.com is a free online OCR (Optical Character Recognition) service, can analyze the text in any image file that you upload, and then convert the text from the image into text that you can easily edit on your computer

Features

- Unlimited uploads
- No registration required
- Keeps your data safe and secure
- Based on Tesseract OCR engine
- 122 recognition languages and fonts support
- Multi-language recognition
- Mathematical equations recognition
- Page layout analysis (multi-column text recognition)
- Selection of area on page for OCR
- Page rotation: clockwise/counterclockwise 90°, 180°
- Different ways to display and process the resulting text
  - ✓ Download as file
  - ✓ Edit in Google Docs
  - ✓ Translate using Google Translate or Bing Translator
  - ✓ Publish online (Pastie.com or Pastebin.com)
  - ✓ Copy to Clipboard
- Supports poorly scanned and photographed pages
- Supports low-resolution images

Input file formats

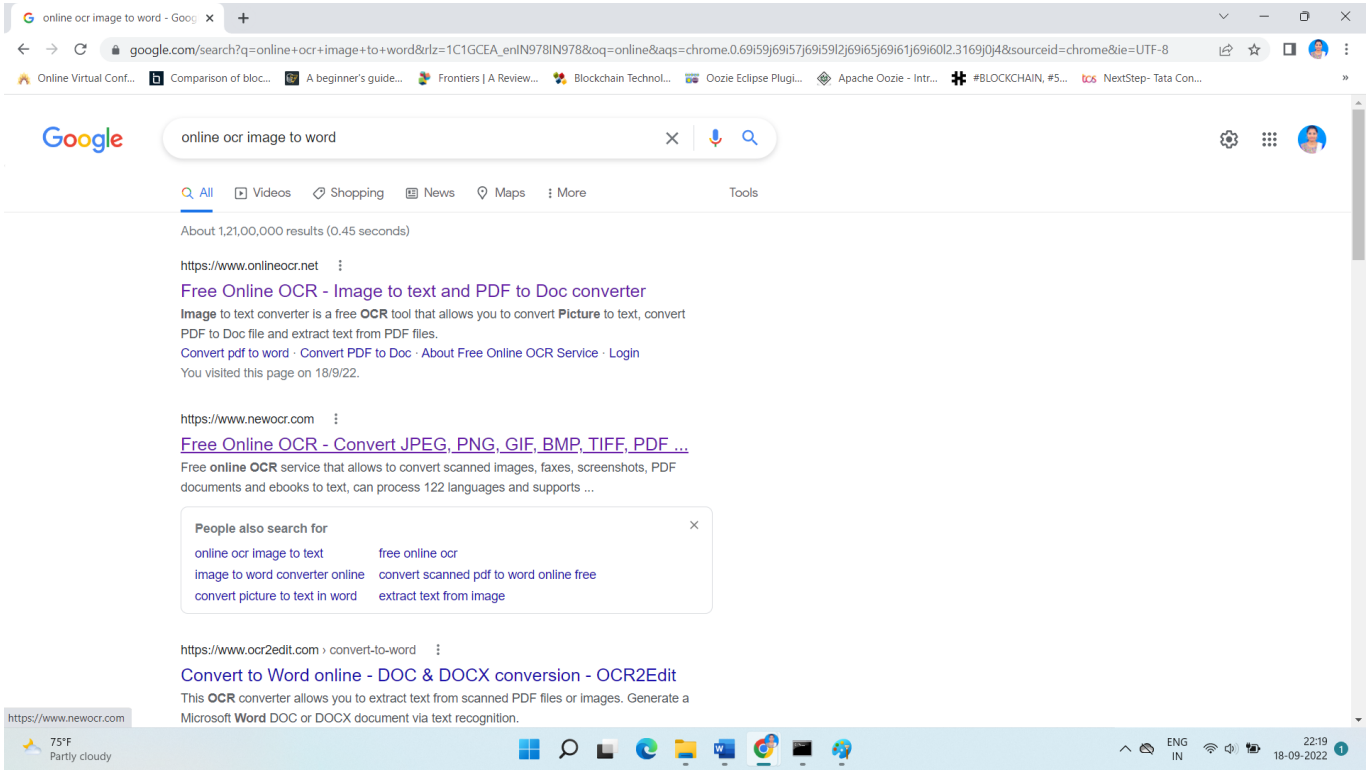
- JPEG, JFIF, PNG, GIF, BMP, PBM, PGM, PPM, PCX
- Compressed files: Unix compress, bzip2, bzip, gzip
- Multi page documents: TIFF, PDF, DjVu
- DOCX, ODT files with images
- Multiple images in ZIP archive

Output file formats

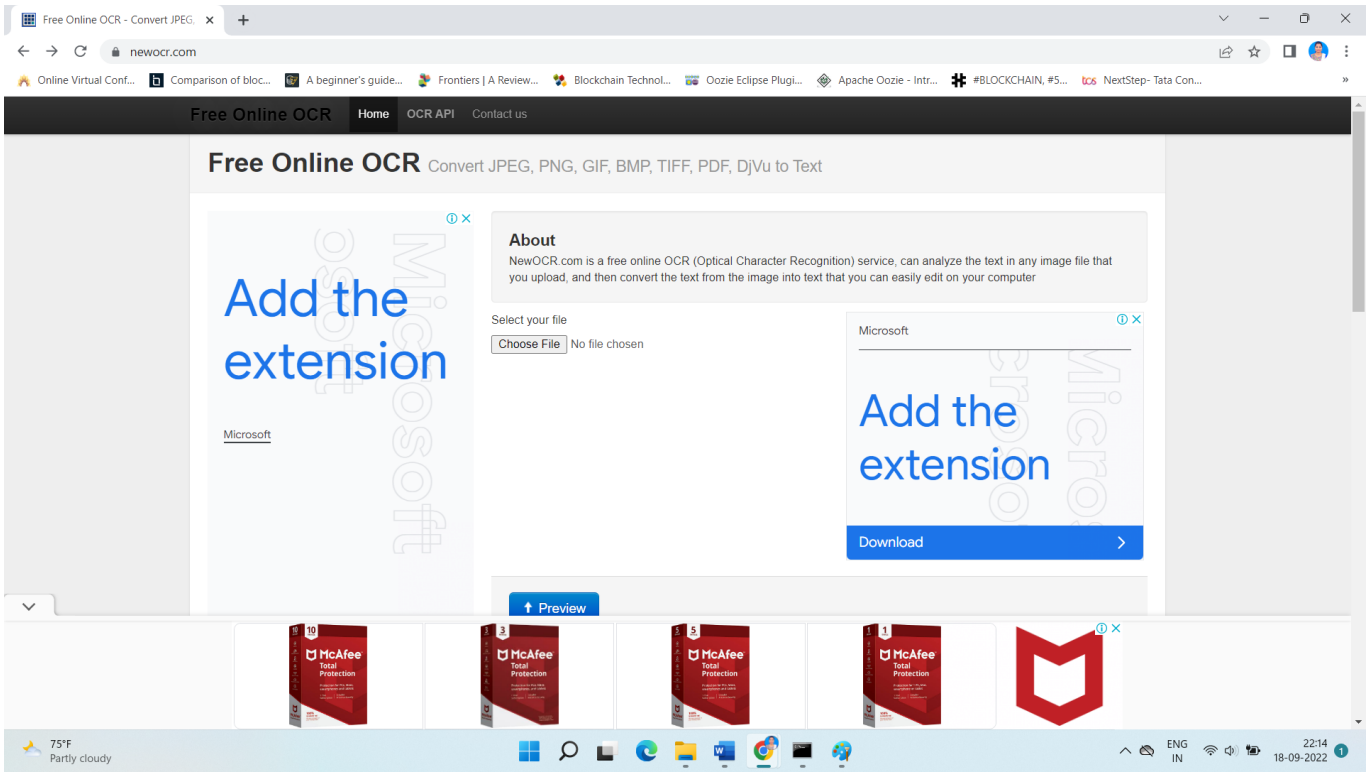
- Plain text (TXT)
- Microsoft Word (DOC)
- Adobe Acrobat (PDF)

Converting image file into text file using online OCR

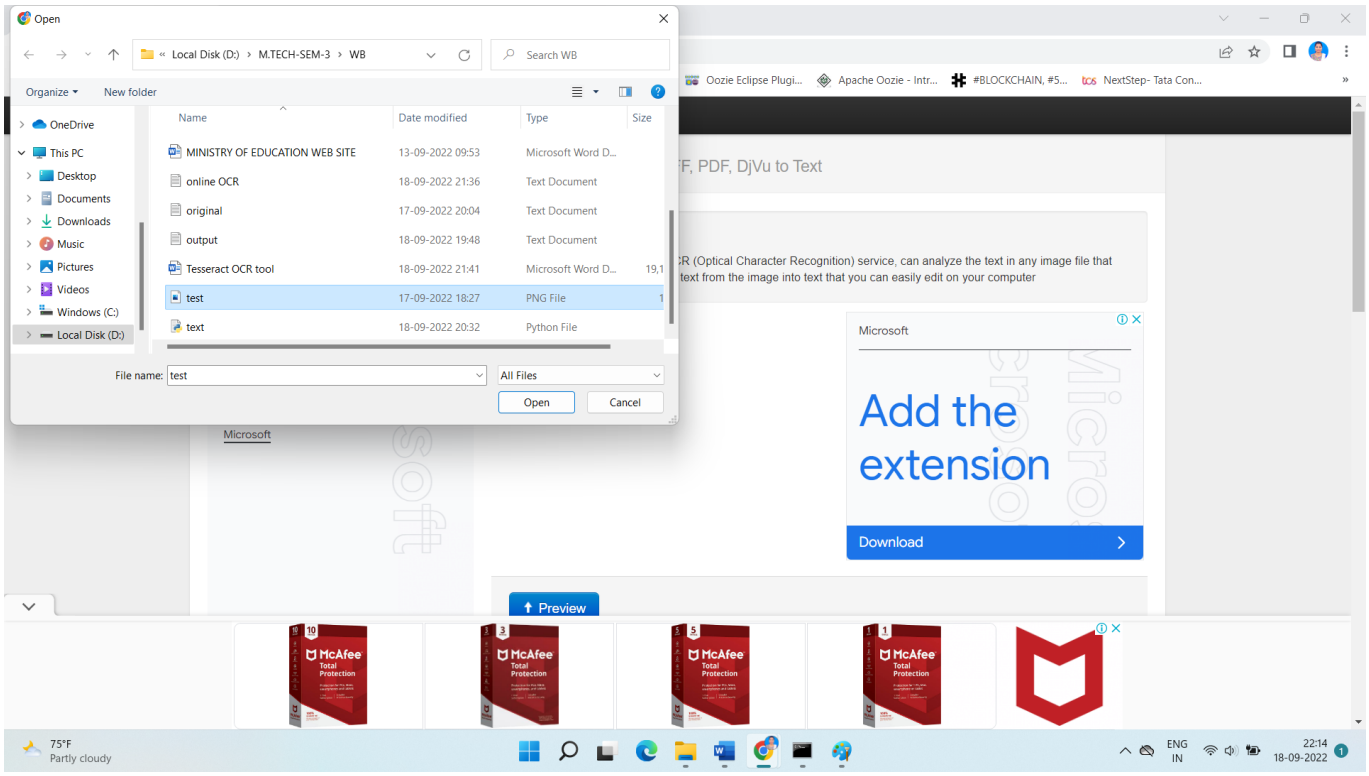
Step 1: In Google Type “online ocr image to word” and select newocr website(www.newocr.com)



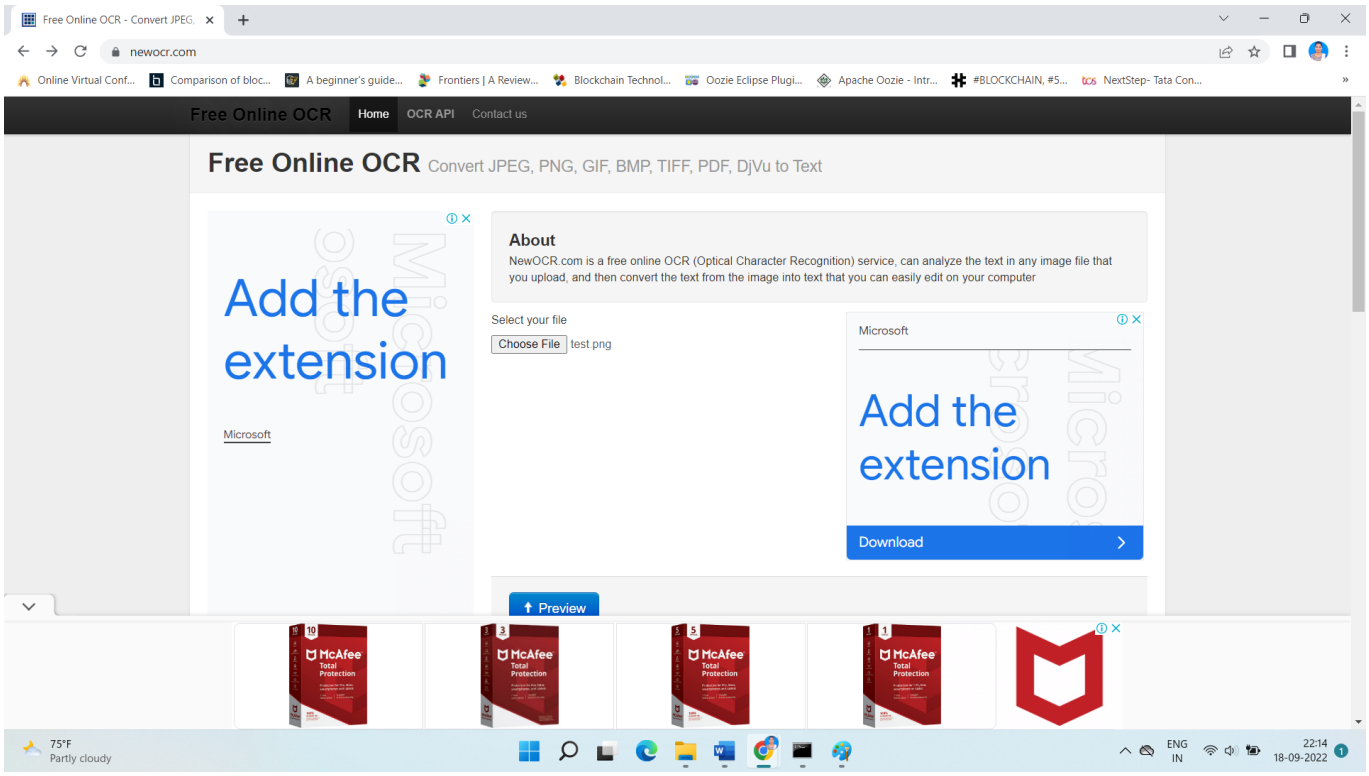
Step 2: Click second link and web page will appear, Click Choose file option



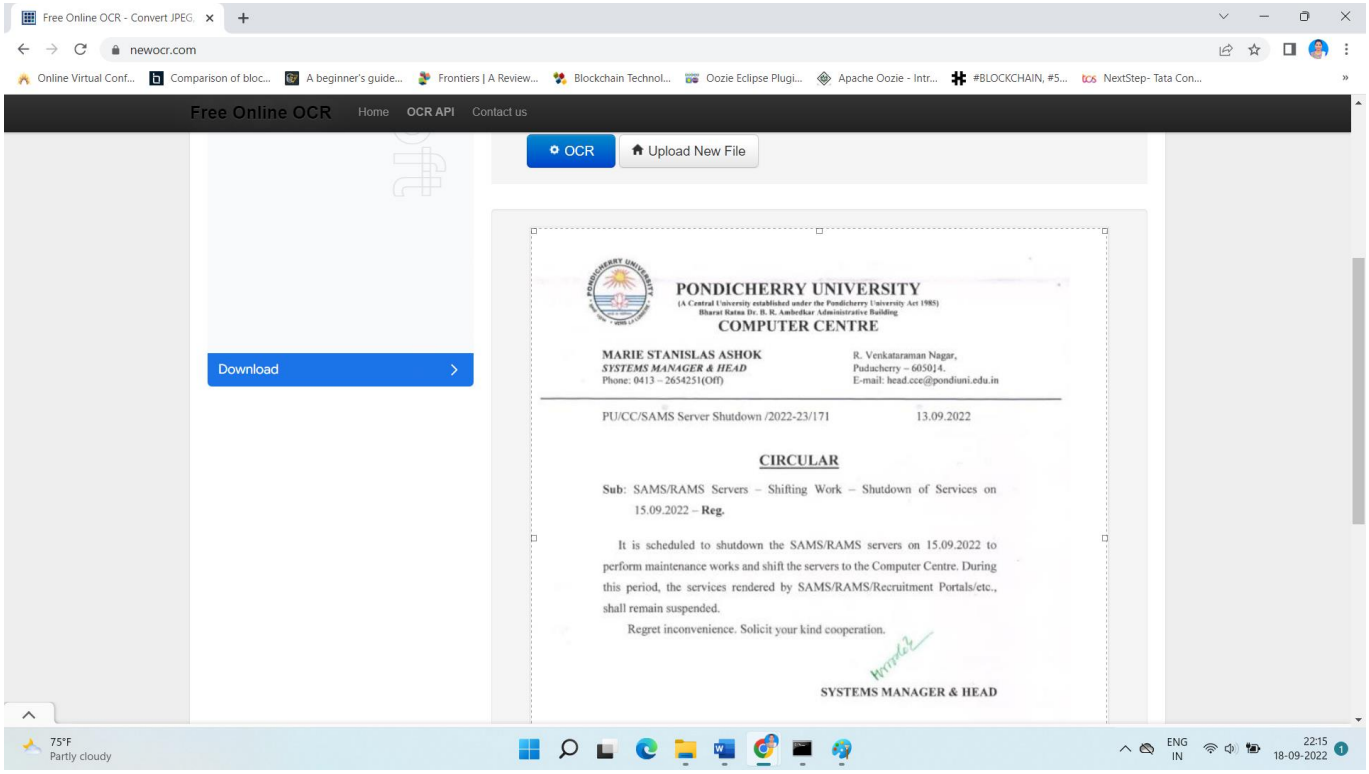
Step 3: Choose file from my local drive and click open



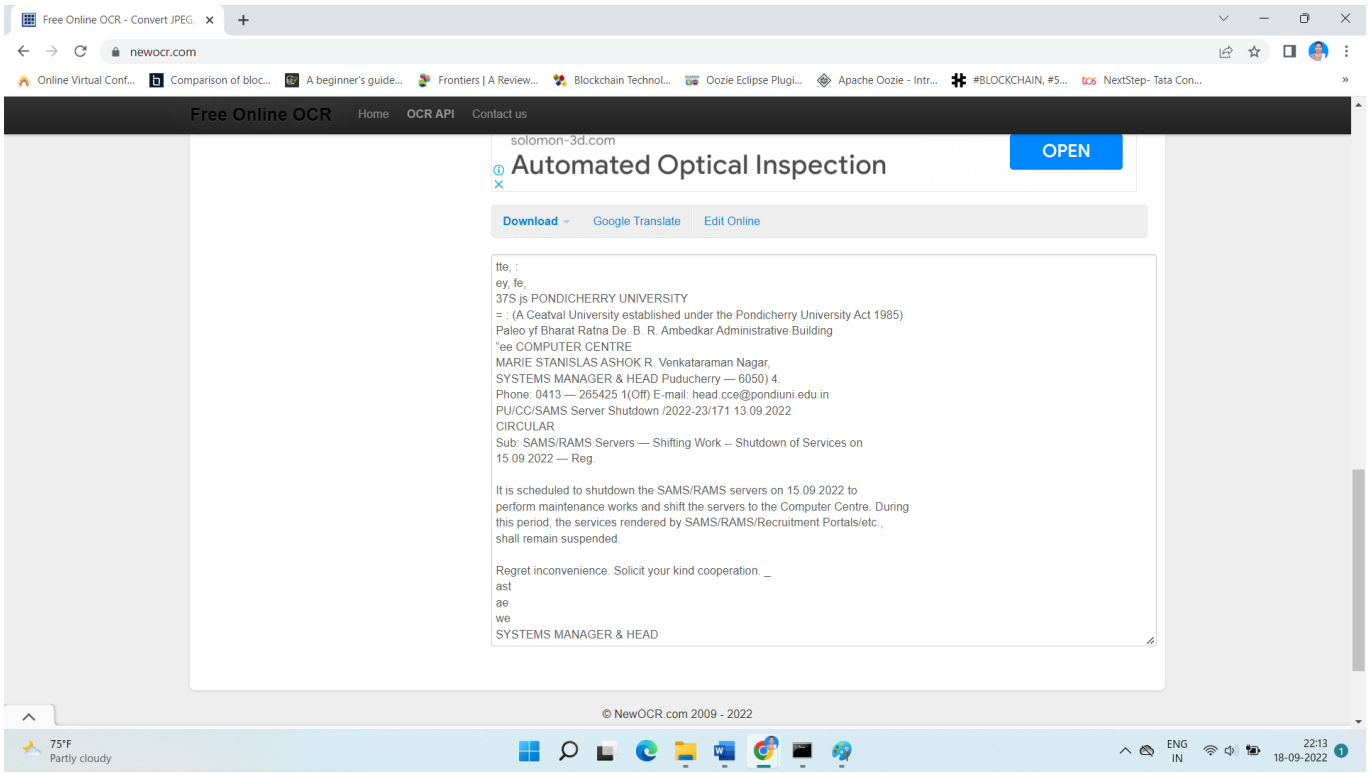
Step 4: Now “text.png” file is selected



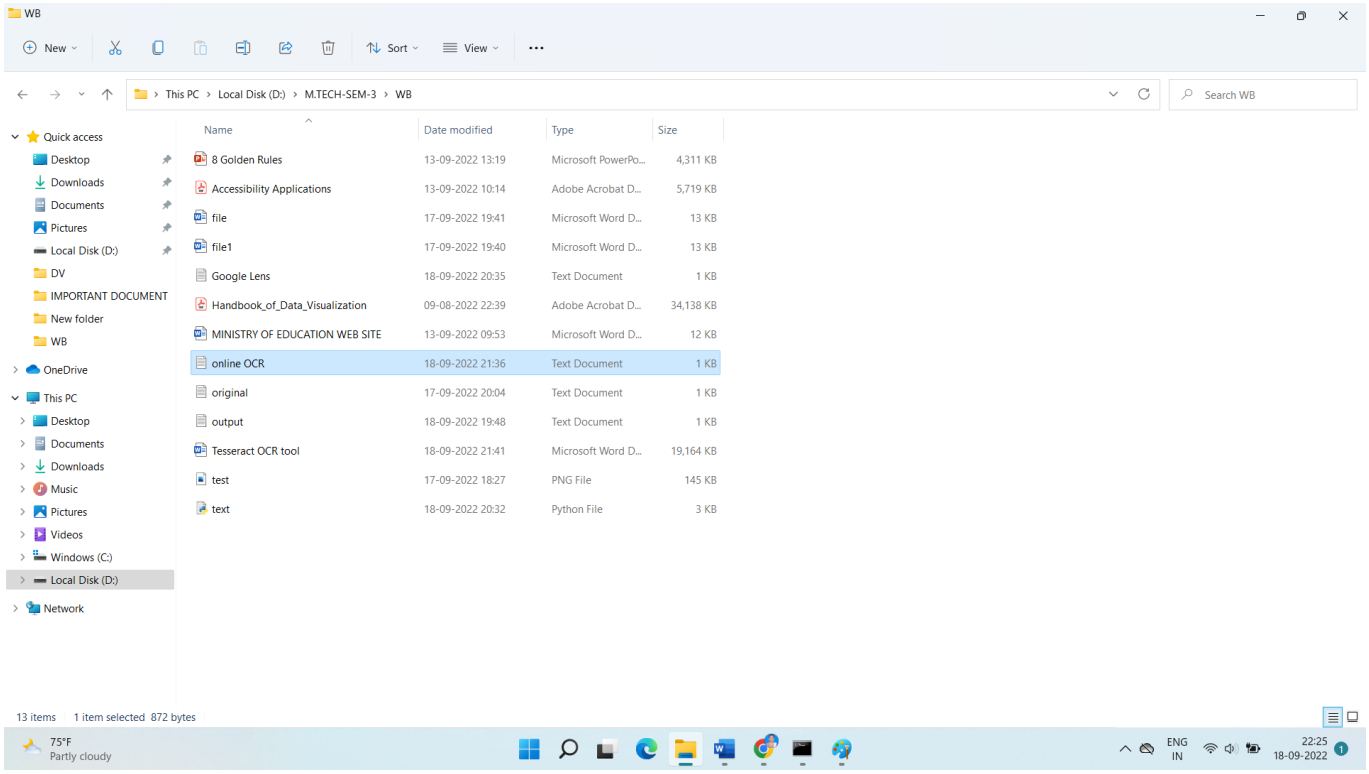
Step 5: Preview file and click OCR button

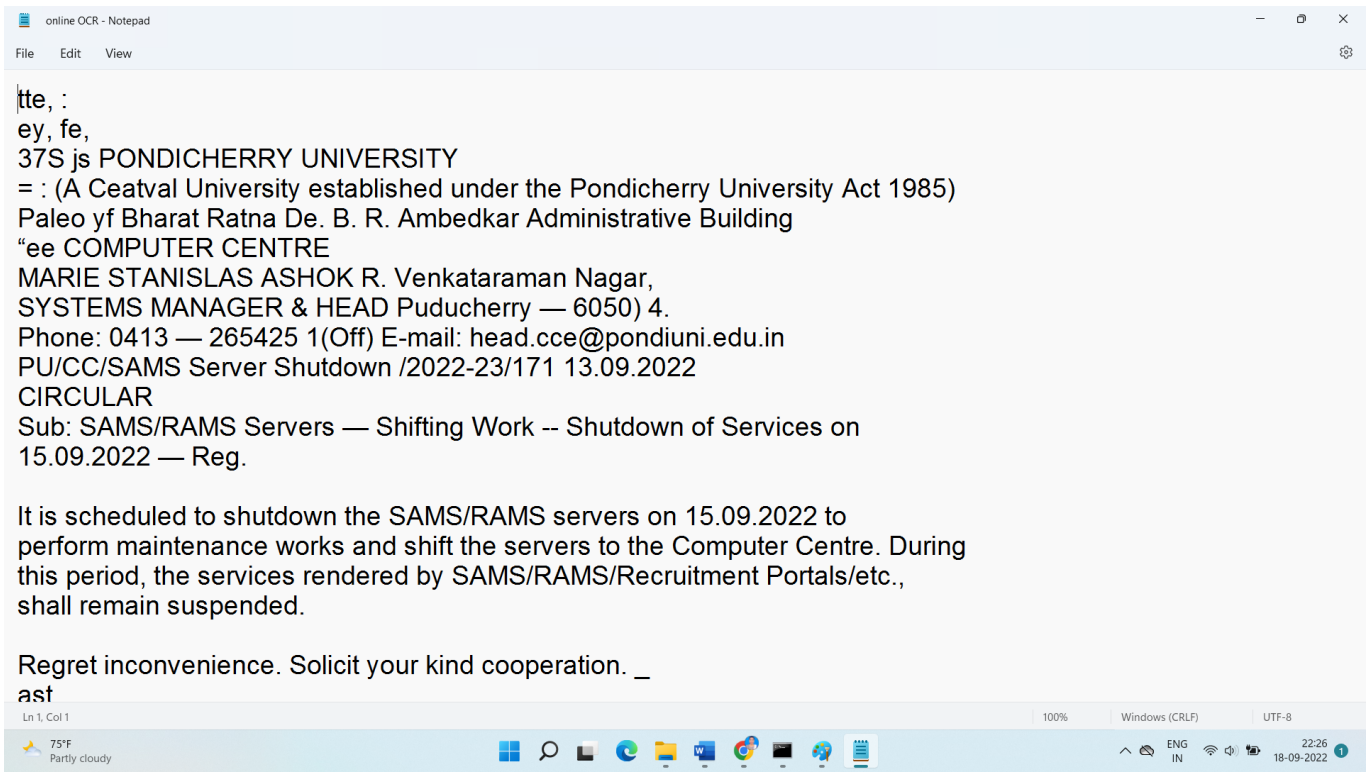


Step 6: Now scroll down and see text is converted



Step 7: Copy the text and save it in local drive as “online OCR”





Document Similarity

- Document similarity, as the name suggests determines how similar are the two given documents. By “documents”, we mean a collection of strings.
- Document similarity is also known as Distance between the Document.

Document Similarity in Python Coding for Tesseract software

```
import math

import string

import sys

# reading the text file

# This functio will return a

# list of the lines of text

# in the file.

def read_file(filename):

    try:

        with open(filename, 'r') as f:

            data = f.read()

        return data

    except IOError:

        print("Error opening or reading input file: ", filename)

        sys.exit()

# splitting the text lines into words

# translation table is a global variable

# mapping upper case to lower case and

# punctuation to spaces

translation_table = str.maketrans(string.punctuation+string.ascii_uppercase,
                                  " "*len(string.punctuation)+string.ascii_lowercase)

# returns a list of the words
```

```
# in the file
def get_words_from_line_list(text):
        text = text.translate(translation_table)
        word_list = text.split()
        return word_list
# counts frequency of each word
# returns a dictionary which maps
# the words to their frequency.
def count_frequency(word_list):
        D = {}
        for new_word in word_list:
                if new_word in D:
                        D[new_word] = D[new_word] + 1
                else:
                        D[new_word] = 1
        return D
# returns dictionary of (word, frequency)
# pairs from the previous dictionary.
def word_frequencies_for_file(filename):
        line_list = read_file(filename)
        word_list = get_words_from_line_list(line_list)
        freq_mapping = count_frequency(word_list)
        print("File", filename, ":", )
        print(len(line_list), "lines, ", )
        print(len(word_list), "words, ", )
        print(len(freq_mapping), "distinct words")
```

```
        return freq_mapping
# returns the dot product of two documents
def dotProduct(D1, D2):
        Sum = 0.0
        for key in D1:
                if key in D2:
                        Sum += (D1[key] * D2[key])
        return Sum
# returns the angle in radians
# between document vectors
def vector_angle(D1, D2):
        numerator = dotProduct(D1, D2)
        denominator = math.sqrt(dotProduct(D1, D1)*dotProduct(D2, D2))
        return math.acos(numerator / denominator)
def documentSimilarity(filename_1, filename_2):
# filename_1 = sys.argv[1]
# filename_2 = sys.argv[2]
        sorted_word_list_1 = word_frequencies_for_file(filename_1)
        sorted_word_list_2 = word_frequencies_for_file(filename_2)
        distance = vector_angle(sorted_word_list_1, sorted_word_list_2)
        print("The distance between the documents is: % 0.6f (radians)"% distance)
# Driver code
documentSimilarity('original.txt', 'file.txt')
```

OUTPUT RESULT AND SCREENSHOT: (For Tesseract OCR file similarity)

File original.txt :

799 lines,

121 words,

86 distinct words

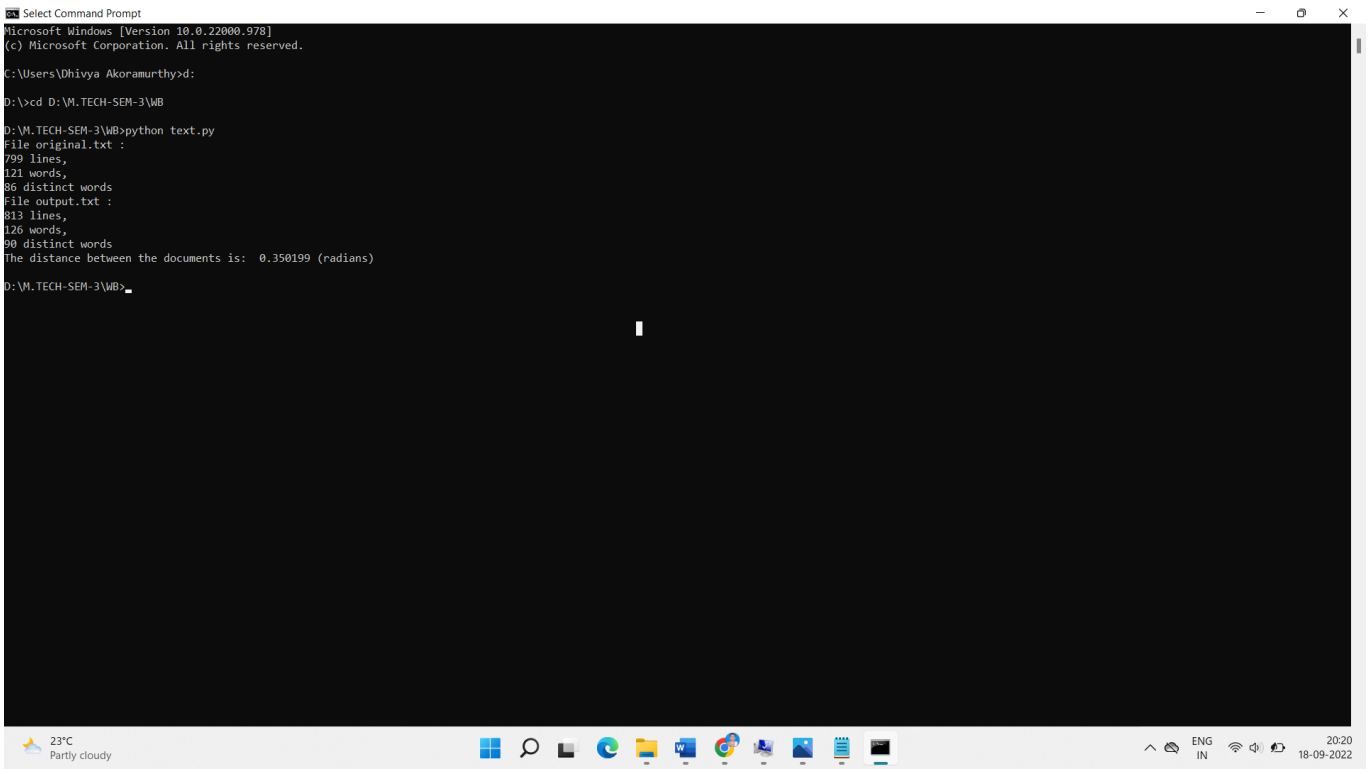
File output.txt :

813 lines,-

126 words,

90 distinct words

The distance between the documents is: 0.350199 (radians)



Document Similarity in Python Coding for Tesseract software

```
import math
import string
import sys
# reading the text file
# This functio will return a
# list of the lines of text
# in the file.
def read_file(filename):
        try:
                with open(filename, 'r') as f:
                        data = f.read()
                return data
        except IOError:
                print("Error opening or reading input file: ", filename)
                sys.exit()
# splitting the text lines into words
# translation table is a global variable
# mapping upper case to lower case and
# punctuation to spaces
translation_table = str.maketrans(string.punctuation+string.ascii_uppercase,
                                  " "*len(string.punctuation)+string.ascii_lowercase)
# returns a list of the words
# in the file
def get_words_from_line_list(text):
        text = text.translate(translation_table)
```

```
        word_list = text.split()
        return word_list
# counts frequency of each word
# returns a dictionary which maps
# the words to their frequency.
def count_frequency(word_list):
        D = {}
        for new_word in word_list:
                if new_word in D:
                        D[new_word] = D[new_word] + 1
                else:
                        D[new_word] = 1
        return D
# returns dictionary of (word, frequency)
# pairs from the previous dictionary.
def word_frequencies_for_file(filename):
        line_list = read_file(filename)
        word_list = get_words_from_line_list(line_list)
        freq_mapping = count_frequency(word_list)
        print("File", filename, ":", )
        print(len(line_list), "lines, ", )
        print(len(word_list), "words, ", )
        print(len(freq_mapping), "distinct words")
        return freq_mapping
# returns the dot product of two documents
def dotProduct(D1, D2):
```

```
        Sum = 0.0
        for key in D1:
                if key in D2:
                        Sum += (D1[key] * D2[key])
        return Sum
# returns the angle in radians
# between document vectors
def vector_angle(D1, D2):
        numerator = dotProduct(D1, D2)
        denominator = math.sqrt(dotProduct(D1, D1)*dotProduct(D2, D2))
        return math.acos(numerator / denominator)
def documentSimilarity(filename_1, filename_2):
# filename_1 = sys.argv[1]
# filename_2 = sys.argv[2]
        sorted_word_list_1 = word_frequencies_for_file(filename_1)
        sorted_word_list_2 = word_frequencies_for_file(filename_2)
        distance = vector_angle(sorted_word_list_1, sorted_word_list_2)
        print("The distance between the documents is: % 0.6f (radians)"% distance)
# Driver code
documentSimilarity('original.txt', Google Lens.txt')
```

OUTPUT RESULT AND SCREENSHOT: (For Google Lens file similarity)

File original.txt :

799 lines,

121 words,

86 distinct words

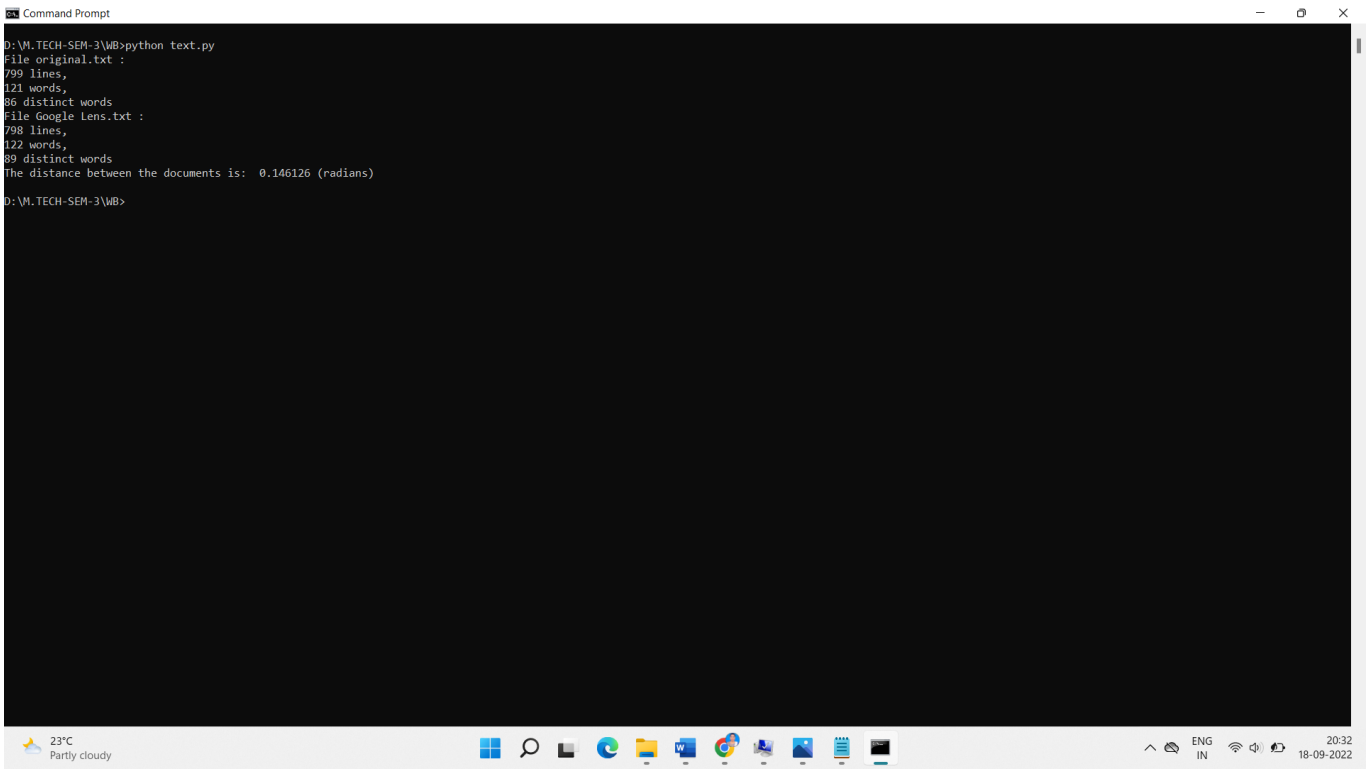
File Google Lens.txt :

798 lines,

122 words,

89 distinct words

The distance between the documents is: 0.146126 (radians)



Document Similarity in Python Coding for online newocr website

```
import math
import string
import sys
# reading the text file
# This functio will return a
# list of the lines of text
# in the file.
def read_file(filename):
        try:
                with open(filename, 'r') as f:
                        data = f.read()
                return data
        except IOError:
                print("Error opening or reading input file: ", filename)
                sys.exit()
# splitting the text lines into words
# translation table is a global variable
# mapping upper case to lower case and
# punctuation to spaces
translation_table = str.maketrans(string.punctuation+string.ascii_uppercase,
                                  " "*len(string.punctuation)+string.ascii_lowercase)
# returns a list of the words
# in the file
def get_words_from_line_list(text):
        text = text.translate(translation_table)
```

```
        word_list = text.split()
        return word_list
# counts frequency of each word
# returns a dictionary which maps
# the words to their frequency.
def count_frequency(word_list):
        D = {}
        for new_word in word_list:
                if new_word in D:
                        D[new_word] = D[new_word] + 1
                else:
                        D[new_word] = 1
        return D
# returns dictionary of (word, frequency)
# pairs from the previous dictionary.
def word_frequencies_for_file(filename):
        line_list = read_file(filename)
        word_list = get_words_from_line_list(line_list)
        freq_mapping = count_frequency(word_list)
        print("File", filename, ":", )
        print(len(line_list), "lines, ", )
        print(len(word_list), "words, ", )
        print(len(freq_mapping), "distinct words")
        return freq_mapping
# returns the dot product of two documents
def dotProduct(D1, D2):
```

```
        Sum = 0.0
        for key in D1:
                if key in D2:
                        Sum += (D1[key] * D2[key])
        return Sum
# returns the angle in radians
# between document vectors
def vector_angle(D1, D2):
        numerator = dotProduct(D1, D2)
        denominator = math.sqrt(dotProduct(D1, D1)*dotProduct(D2, D2))
        return math.acos(numerator / denominator)
def documentSimilarity(filename_1, filename_2):
# filename_1 = sys.argv[1]
# filename_2 = sys.argv[2]
        sorted_word_list_1 = word_frequencies_for_file(filename_1)
        sorted_word_list_2 = word_frequencies_for_file(filename_2)
        distance = vector_angle(sorted_word_list_1, sorted_word_list_2)
        print("The distance between the documents is: % 0.6f (radians)"% distance)
# Driver code
documentSimilarity('original.txt', online OCR.txt')
```

OUTPUT RESULT AND SCREENSHOT: (For online OCR file similarity)

File original.txt :

799 lines,

121 words,

86 distinct words

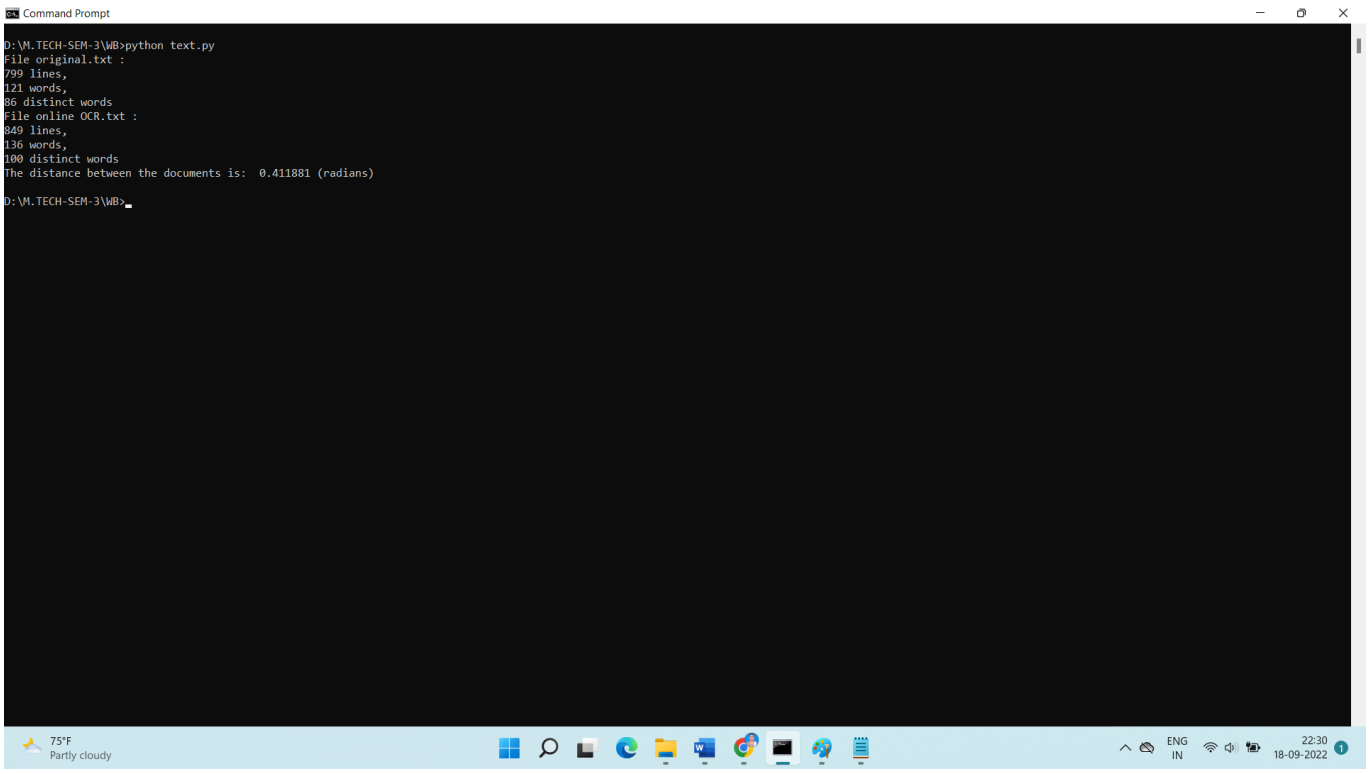
File online OCR.txt :

849 lines,

136 words,

100 distinct words

The distance between the documents is: 0.411881 (radians)



Comparative analysis of OCR tools

Content	Tesseract OCR Tool	Google Lens	Online OCR
Storage	32184 KB	11000KB	NIL
Time Required	1 minutes	1 minutes	1 minutes
Line of text	813	798	849
Words	126	112	136
Distinct Words	90	89	100
Distance between the Documents in Radians	0.350199(radians)	0.146126(radians)	0.411881(radians)
Percentage	5.57359%	2.32566%	6.55528%
Languages	5	159	122
Software Required	Yes	No	No
Application Required	No	Yes	No
Internet Required	No	Yes	Yes
Cost	Yes	No	No
Output file	Only Text	Text, QR code, google information, etc,	Text, PDF, MS word

Conclusion:

From the above result, conclude that **Google Lens** has low cost, identify more languages and also image file, storage also low and less accuracy percentage compare with Tesseract and Online OCR. But Google Lens also not 100% accurate but comparing with Tesseract and Online OCR, **Google Lens is best**.