A

Major project work

On

OCR OF HANDWRITTEN FORMS

(Submitted In Partial fulfillment of the requirements for award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

PRATYASHA PANDA	**187R1A0590**
N. PRERANA	**187R1A0569**
SAQUIB AHMED KHAN	**187R1A0599**

UNDER THE GUIDANCE OF

Dr. PRABHU A
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi) Recognized Under Section 2(f) & 12(B) of the UGCAct.1956, Kandlakoya (V), Medchal Road, Hyderabad-501401.

2018-22

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled **“OCR OF HANDWRITTEN FORMS”** was submitted by **PRATYASHA PANDA, N. PRERANA, SAQUIB AHMED KHAN** bearing the **187R1A0590, 187R1A0569, 187R1A0599** roll numbers in partial fulfillment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering of the Jawaharlal Nehru Technological University Hyderabad, during the year 2021-2022. It is certified that they have completed the project satisfactory.

The results embodied in this thesis have not been submitted to any other university or institute for the award of any degree or diploma.

Dr. Prabhu A
Associate Professor
INTERNAL GUIDE

Dr. A. Raji Reddy
DIRECTOR

Dr. K. Srujan Raju
HEAD OF THE DEPARTMENT

EXTERNAL EXAMINER

Submitted for viva voce Examination held on______________.

ACKNOWLEDGEMENT

Apart from our efforts, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We take this opportunity to express our profound gratitude and deep regard to our guide **Dr. Prabhu A**, Associate Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. A. Uday Kiran**, **Mr. J. Narasimha Rao, Dr. T. S. Mastan Rao, Mrs. G. Latha, Mr. A. Kiran Kumar** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju**, Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy**, Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to **Sri. Ch. Gopal Reddy**, Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

PRATYASHA PANDA	**187R1A0590**
N. PRERANA	**187R1A0569**
SAQUIB AHMED KHAN	**187R1A0599**

ABSTRACT

Handwriting recognition tends to be significantly harder than traditional OCR that uses specific fonts/characters. The reason for this is unlike computer fonts, there are infinite variations of handwriting styles. Every human has its own way of writing which is specific and unique. The variations in handwriting styles creates problem for the Optical Character Recognition engines, which are typically trained on computer fonts, not handwriting fonts.

There can be cases where the characters are connected to each other, touches one after other, may miss some characters in between or does spelling mistakes, etc which makes it challenging for OCR algorithms to separate them or do the correction with higher accuracy which can be done easily for computer fonts rather than handwritten forms, ultimately leading to incorrect OCR results. Nevertheless, it's a crucial problem to solve for multiple industries like healthcare, insurance and banking. There is no engine that provides OCR results in bulk but recognizes one paper at a time or the selected lines or images.

This project uses deep learning to solve this problem by making the algorithms advance by training them with more data which will provide he users an interface where they can convert their desired documents into text forms with much higher accuracy.

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

This project is created to give ser an engine which they can use to convert their documents into text form in case of blur, shady or old documents. It is a simple and cost-effective solution.

1.1 PROJECT SCOPE

This platform will provide individuals, organizations, industries or businesses to get their data in one form for easy usage. This will help in getting large data into text form with higher quality and in minimum time. It will improve customer experience which will massively improve customer service.

1.2 PROJECT PURPOSE

Our purpose is to provide a system where we don't have to do paper work or use some software that does not fulfill all the requirements. This system is efficient for customers/clients. The newly introduced system will provide easy access to the system and it will contain user friendly functions with an attractive interface. The system will give better options for the problem of handling large scale physical file systems, for the errors occurring in calculations and all the other required tasks that have been specified by the client.

1.3 PROJECT FEATURES

The main aim of the entire activity is let the customer easily convert their files into text forms whether it contains bulk or single pages with the feature of choosing templates of their choice i.e., specifically choosing lines or part of document they want to convert instead of converting the whole document. In our system they don't have to login or sign up, they can simply choose the file to convert and our system will do it for them.

2. SYSTEM ANALYSIS

In this Phase we have to consider the existing system and its limitations. Every minute detail should be taken care of to provide the best product. We have to find a solution for the complications in the existing system and determine a new system. It is a good introductory guide that provides an overview of all the concepts necessary to build a system. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose.

2.1 PROBLEM DEFINITION

We often come around situations where we sometime require some documents which are old (agreements, loans, attachments, etc.) or any official or unofficial documents for some work purpose but we find it difficult to read whether it is soft or hard copy because of shady, blur fonts or due to fading of texts or less light. In these cases, we might face some problems so why not build an engine which can convert these kinds of documents in text forms which will be readable and editable.

OCR of handwritten forms provides a new engine which will be produced by the project team in order to overcome the problems that have occurred due to the current outdated engine and manual work. The newly introduced system will provide easy access to the system and it will contain user friendly functions with attractive interfaces. The system will give better options for the problem of handling large data containing files, for the errors occurring while recognition and conversion and all the other required tasks. The final outcome of this project will increase the efficiency of the texts in a much convenient manner.

2.2 EXISTING SYSTEM

The existing OCR engines are made up in a manner that they have the knowledge about the spaces and type of fonts, size. Coming to handwritten forms it cannot recognize the texts efficiently as it was made for the existing fonts so it will be time consuming to read each line of the entire document. With that the unnecessary part of the document is taken out manually. It works efficiently for printed texts and not for handwritten texts which results in texts with errors and reduced quality. It also doesn’t treat the same kind of documents in the minimum time possible.

2.2.1 LIMITATIONS OF THE EXISTING SYSTEM

- Cannot be used for extremely shabby/blur handwriting.
- Selected templates couldn’t be possibly converted.
- Slow process of conversion for large data.
- Not for image documents.
- Inaccuracy of results.

2.3 PROPOSED SYSTEM

In the proposed system, it will be able to recognize handwritten texts however it is written and it will provide recognition in bulk for the same kind of data (e.g., Bank forms). It will also be able to recognize the desired part of the document with 90-94% of accuracy. It will be having a simple interface where the user/client can just select the document and rest will be done by the engine which will provide results as per the user and their preference.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

- Reduces costs like printing, shipping and copying.
- Massively improves customer service.
- It will reduce data inaccuracy.
- High level of accessibility.
- Highly productive.

2.4 FEASIBILITY STUDY

Feasibility study of the system is a very important stage during system design. Feasibility study is a test of a system proposal according to its workability impact on the organization, ability to meet user needs, and effective use of resources. Feasibility study decides whether the system is properly developed or not.

Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Behavioural Feasibility

2.4.1 ECONOMICAL FEASIBILITY

As a part of this, the costs and benefits associated with the proposed system are compared and the project is economically feasible only if tangible and intangible benefits outweigh the cost. The cost for the proposed engine is outweighing the cost and efforts involved in converting the documents in desired form. The engine also reduces the additional human work to do various jobs that a single updated engine can do. So, this system is economically feasible.

2.4.2 TECHNICAL FEASIBILITY

In this, one has to test whether the system can be developed using existing technology or not. We have used Python as front-end, Excel to maintain records while executing the operation and OpenCV & OCR for recognition. It is evident that necessary hardware and software are available for development and implementation of the proposed system.

2.4.3 BEHAVIOURAL FEASIBILITY

The project would be beneficial because it satisfies the objectives when developed and performed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE AND SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are some hardware requirements -

Table 2.1: Hardware requirements

Speed	3.9 GHz
Ram	8.00 GB
Hard Disk	500 MB
Processor	Intel(R) Core (TM) i5-8265U CPU @ 1.60GHz 1.80 GHz
System type	64-bit operating system, x64-based processor

2.5.2 SOFTWARE REQUIREMENTS

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements -

Table 2.2: Software requirements

Technology	PYTHON 3.8
Framework	Django, OpenCV, Tesseract OCR
Web technologies	HTML, CSS, JavaScript
IDE	PyCharm 3.0
Web Server	Django Server
Database	Excel

3. ARCHITECTURE

3.1 PROJECT ARCHITECTURE

The project architecture describes how data will be stored, processed and maintained. This also tells the overall idea about the system and processing of each component and working of each module.

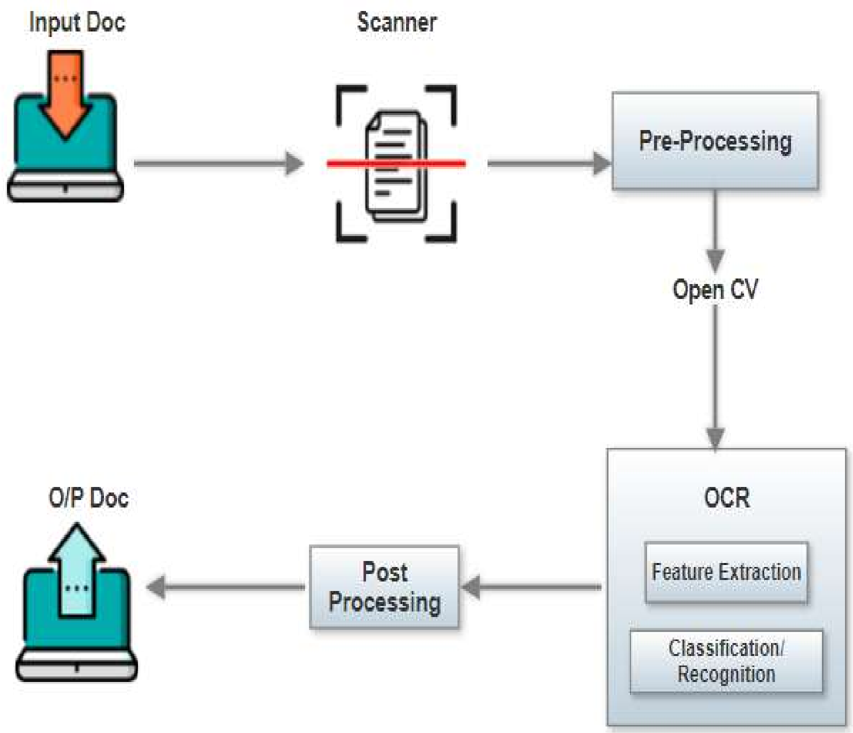


Figure 3.1 Project architecture

Here we can see that when a user will open the website, they can choose their file from their directories which can be in any format. The user can also select some part of the document that they want to convert instead of the whole document. Later with the use of OpenCV it will do the image processing for the whole document and then select the templates if given by the user to convert then, with the help of Tesseract OCR and new algorithms it will do the conversion by undergoing different stages where the image will be divided in parts and algorithms will apply and change character to character and followed by organizing them back and providing the text format to the user. This output file will be editable as well.

3.2 DESCRIPTION

When a user logs into the website, they can choose their file from their directories and upload. The user can also select templates of their choice from the document for generating excel sheets. It will return the resulting text form or excel sheet to the user.

1. IMAGE CAPTURING

This step consists of scanning. In scanning, a digital image of typewritten or hand-written or image document is captured. To perform this method, optical scanners are used which are easily accessible from the market. Optical scanners convert colored levels into gray-levels for machine understanding purposes.

2. PRE-PROCESSING

Preprocessing of an image is performed to enhance the possibilities of successful recognition. Normally noise filtering, smoothing filing and thinning are performed during this step. Image that was scanned during image capturing might contain a precise amount of noise.

3. FEATURE EXTRACTION

In this part, features of each character are extracted. The extracted features from input character ought to permit classification in an exceedingly distinctive approach. Different types of features such as the image itself, geometrical features and statistical features can be used.

4. CLASSIFICATION AND RECOGNITION

The aim of creating a part of the recognition system is the classification stage. Here the matrix containing the image of input characters is directly matched with the set of paradigm characters representing every possible category. The category of prototype giving the simplest match is allotted to the pattern.

5. POST-PROCESSING

At the final stage, it prints the recognized characters in structured text form. We get a set of individual characters, but these characters in themselves do not contain meaningful information. But once grouped in the form of string or word they can convey a meaning.

3.3 MODULE DESCRIPTION

This gives us the modules that are being used in the project and tells us how it will perform in the project.

1. **USER/CLIENT –**

   The users/clients can upload the document which they want to convert with which they can also choose particular fields to be converted if necessary.

2. **ADMINISTRATOR –**

   This is basically the interference which will take a user given document as input and do the processing in order to convert it into text form and will provide the editable output file to the user as a result of the user's action on SUBMIT.

3.4 UML DIAGRAMS

UML (Unified Modelling Language) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems.

3.4.1 USE CASE DIAGRAM

Use case diagrams are a set of use cases, actors, and their relationships. In here the user/client can perform tasks such as uploading document to the website and view the output whereas the administrator will do the part of proving the resulted text form or excel sheet.

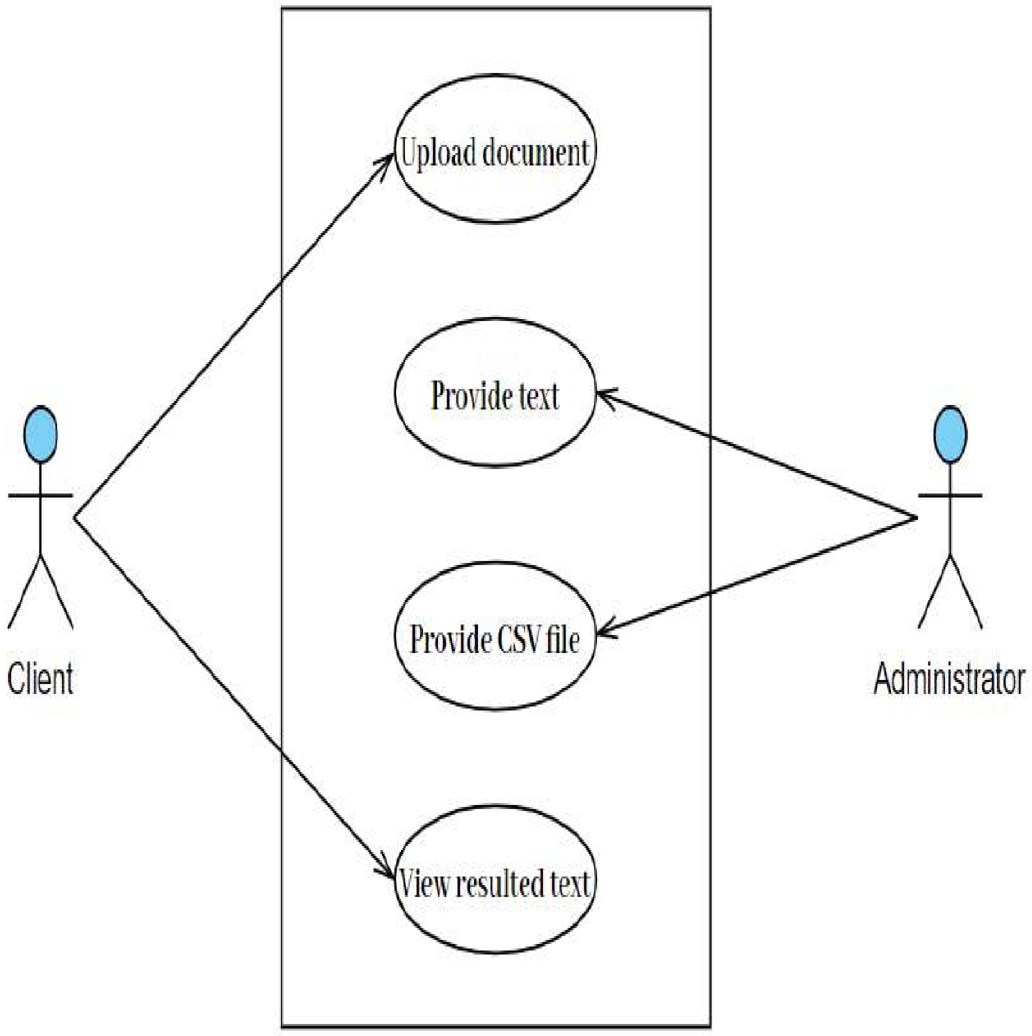


Figure 3.2 Use case diagram

3.4.2 CLASS DIAGRAM

Class diagrams basically represent the object-oriented view of a system. Each object is provided with some buttons or functionalities which perform certain process. Each button has some specific operation and causes desired results such as when user clicks the upload button the document undergoes conversion process and when given the path for output and clicking o submit the resulted excel sheet will get saved. The website will take only pdfs or image as input while performing the conversion.

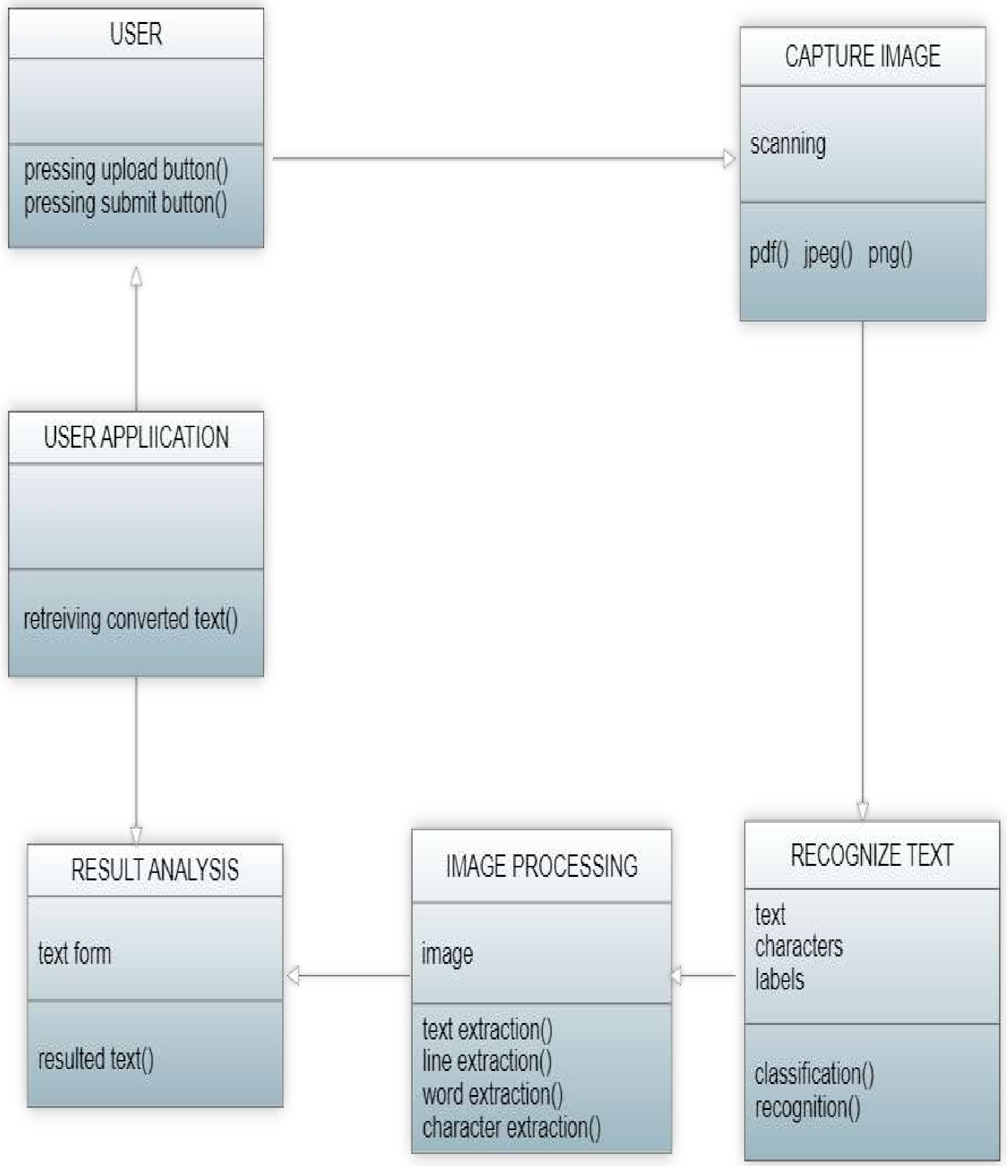


Figure 3.3 Class diagram

3.4.3 SEQUENCE DIAGRAM

A sequence diagram simply depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. Following depicts that the user can log into the website and upload the document and wait for the conversion of file as a result to which user can view the text form in the website itself or can also access the excel sheet saved in folder given by user. After successful conversion and viewing of result user can log off the website.

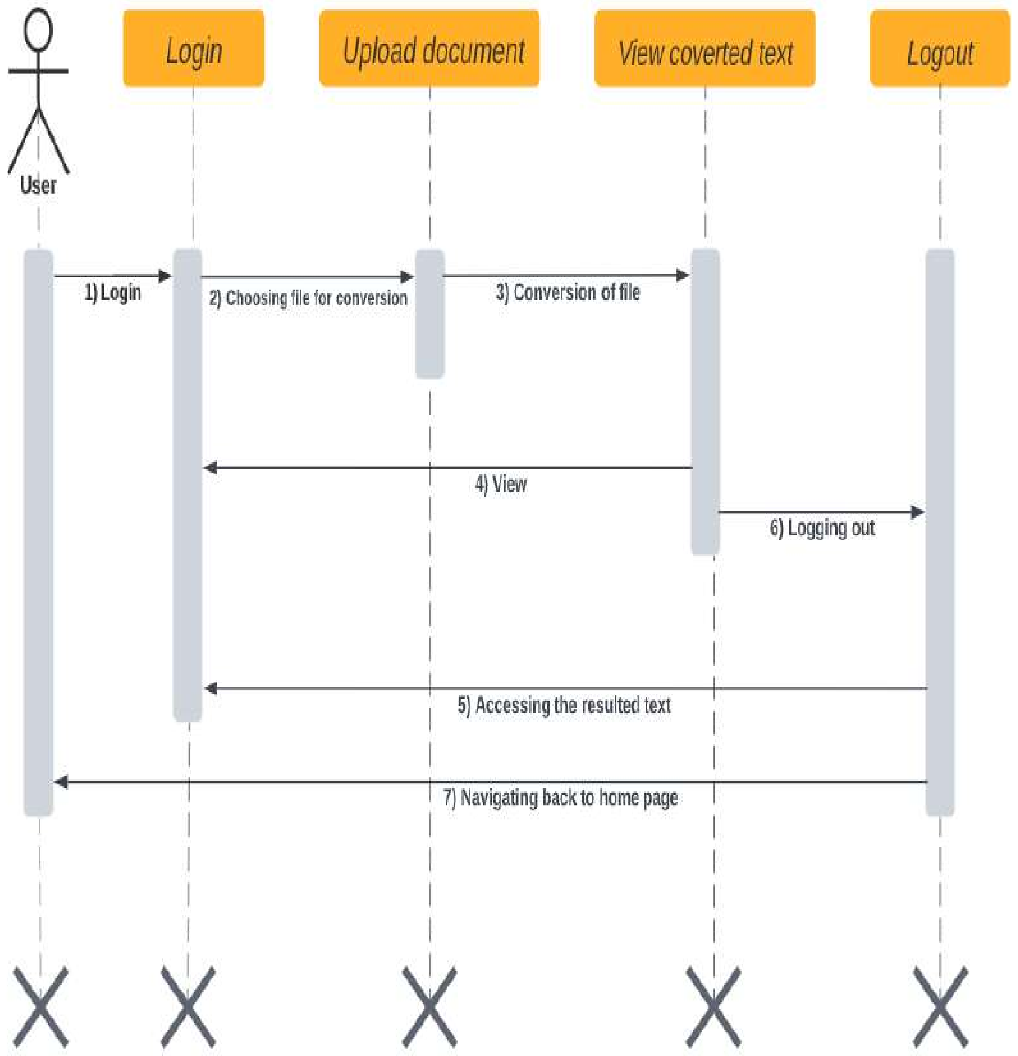


Figure 3.4 Sequence diagram

3.4.4 ACTIVITY DIAGRAM

It represents the flow from activity to activity. The activity takes place by importing packages like cv2, os, blob, pdf2image which are necessary in order to run on any OS, to convert colours, to retrieve files. Pytesseract is used in order to OCR the content of pdf or image into text form where the alphabets, numbers, special characters are read and divided into clusters and for every new character it finds its place in the nearest or similar cluster. With sentiment analysis grammar is checked as well as spellings and throws output in the website itself for the user to access it.

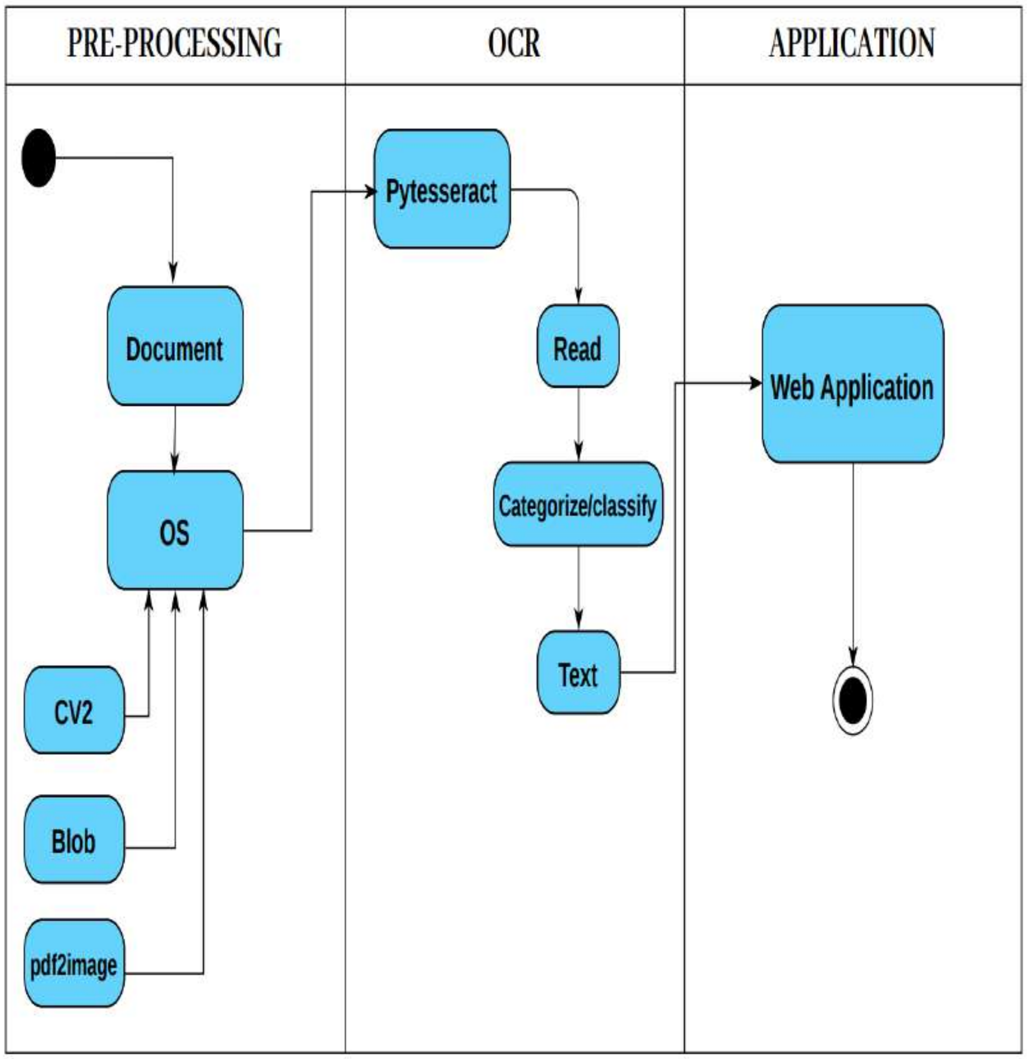


Figure 3.5 Activity diagram

4. IMPLEMENTATION

4.1 SAMPLE CODE

Before executing the program, we should make databases, connect the database to the server, write HTML codes for the website and use other functionalities to run the project as well make it look good.

4.1.1 CONVERSION OF ONLY IMAGE OR PDF FILES

```
import pytesseract
import glob
from pdf2image import convert_from_path

images = glob.glob(r"Images\*.png")
for img in images:
   image_text = pytesseract.image_to_string(img)
   with open(f'{img}.txt','w') as the_file:
       the_file.write(image_text)
pdfs= glob.glob(r"SCANNED_PDFS\*.pdf")

for pdf_path in pdfs:
   pages = convert_from_path(pdf_path, 500)

   for pageNum, imgBlob in enumerate(pages):
       text = pytesseract.image_to_string(imgBlob, lang='eng')

       with open(f'{pdf_path[:-4]}_page_{pageNum}.txt', 'w') as the_file:
           the_file.write(text)
```

4.1.2 CONVERSION OF ALL TYPES OF FILES

```
import pytesseract
import glob
from pdf2image import convert_from_path

all_files = glob.glob(r"All Files\*")
for file in all_files:
   # if file is an images, use list for too many file extensions
   if '.png' in str(file): # For Image
       image_text = pytesseract.image_to_string(file)
       with open(f'{file}.txt', 'w') as the_file:
           the_file.write(image_text)

   elif '.pdf' in str(file):  # For PDF
       pages = convert_from_path(file, 500)
       for pageNum,imgBlob in enumerate(pages):  # For Each Page
           text = pytesseract.image_to_string(imgBlob,lang='eng')
           with open(f'{file[:-4]}_page_{pageNum}.txt', 'w') as the_file:
               the_file.write(text)
```

4.1.3 IMPORTING PACKAGES & SETTING PATHS

```
import os
import shutil
import pytesseract
from flask import Flask, jsonify, render_template, request
from flask_cors import CORS, cross_origin
from pdf2image import convert_from_path

homeDirectory = os.path.expanduser('~')
pytesseract.pytesseract.tesseract_cmd = os.path.join(homeDirectory, 'Tesseract-OCR',
"tesseract.exe")
app = Flask(__name__)
app.config.from_object(__name__)
app.config['UPLOAD_FOLDER'] = 'static/js/uploads'
```

4.1.4 ENABLING CORS

```
CORS(app, resources={r"/api/*": {"origins": "*"}}, supports_credentials=True)
global files, is_pdf, ocrPage, img
files = []
is_pdf = False
ocrPage = 1
def delete_folder(path):
   if os.path.exists(path):
       shutil.rmtree(path)

def create_folder(path):
   print(f"Creating Folder - {path}")
   if not os.path.exists(path):
       os.makedirs(path)

@app.route('/')
def hello_world():
   return render_template('index.html')
```

4.1.5 OCR OF TEXTS

```
@app.route('/ocr', methods=['POST'])
def get_ocr_text():
   global ocrPage, img
   print('Image width and height', img.shape[1], img.shape[0])
   try:
       width, height = request.json['width'], request.json['height']
       x, y = int(request.json['x']), int(request.json['y'])
       gray_image = img[y: y + height, x: x + width]
       gray_image = Image.fromarray(gray_image)
       print(x,y, width, height)
       text = pytesseract.image_to_string(gray_image)
       print(text)
       return jsonify({'result': text})
   except Exception as e:
       print(e)
       return jsonify({'result': "Something went wrong"})
```

4.1.6 IMAGE CONVERSION PROCESS

```
def imageGenericProcess(coordinates, files, savePath):
   print(coordinates)
   data = []
   try:
       for file in files:
           print(f'Processing - {file}')
           image = c.imread(file, 0)
           temp = {'file': os.path.basename(file)}
           for i, coords in enumerate(coordinates, 1):
               x, y, width, height = coords['x'], coords['y'], coords['width'], coords['height']
               gray_image = image[y: y + height, x: x + width]
               gray_image = Image.fromarray(gray_image)
               text = pytesseract.image_to_string(gray_image)
               try:
                   temp[coords['colName']] = text
               except KeyError as k:
                   print('The colName field was not added')
                   temp[f'Columns {i}'] = text
           data.append(temp)
       pd.DataFrame(data).to_excel(savePath, index=False)
   except Exception as e:
       print(e)
       return False
   return True
```

4.1.7 PDF CONVERSION PROCESS

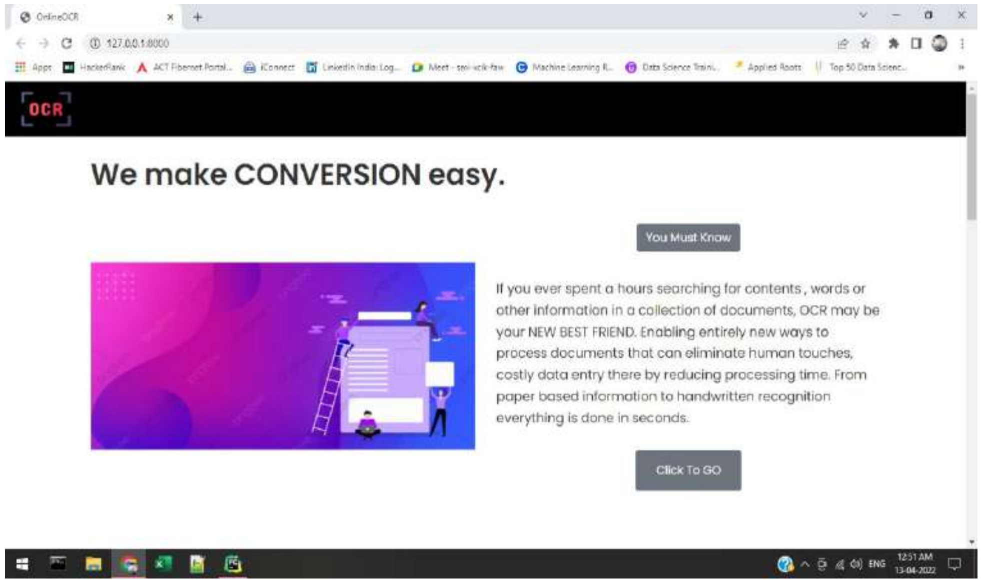
```
def pdfGenericProcess(pageCoorinates, files, savePath):
   print(pageCoorinates)
   data = []
   try:
       for file in files:
           print(f'Processing {file}')
           try:
               images = convert_from_path(os.path.join(app.config['UPLOAD_FOLDER'], file),
poppler_path=r'poppler-r'.51\bin')
           except Exception as e:
               print(e)
               print('Problem Converting PDF to Image.')
           for i, image in enumerate(images, 1):
               image = np.array(image)
               page_data = {'Page': i}
               try:
                   currentCoordinates = pageCoorinates[str(i)]
               except KeyError:
                   continue
               for coords in currentCoordinates:
                   x, y, width, height = coords['x'], coords['y'], coords['width'], coords['height']
                   img = image[y: y + height, x: x + width]
               page_data['file'] = os.path.basename(file)
               data.append(page_data)
       print(data)
       pd.DataFrame(data).to_excel(savePath, index=False)
       except Exception as e:
print(e)
```

5. SCREENSHOTS

After executing the code, we will get a website link which will direct us to the page where we can see our whole output.

5.1 WEBSITE FOR CONVERSION INTO TEXT

5.1.1 WEBSITE HOMEPAGE

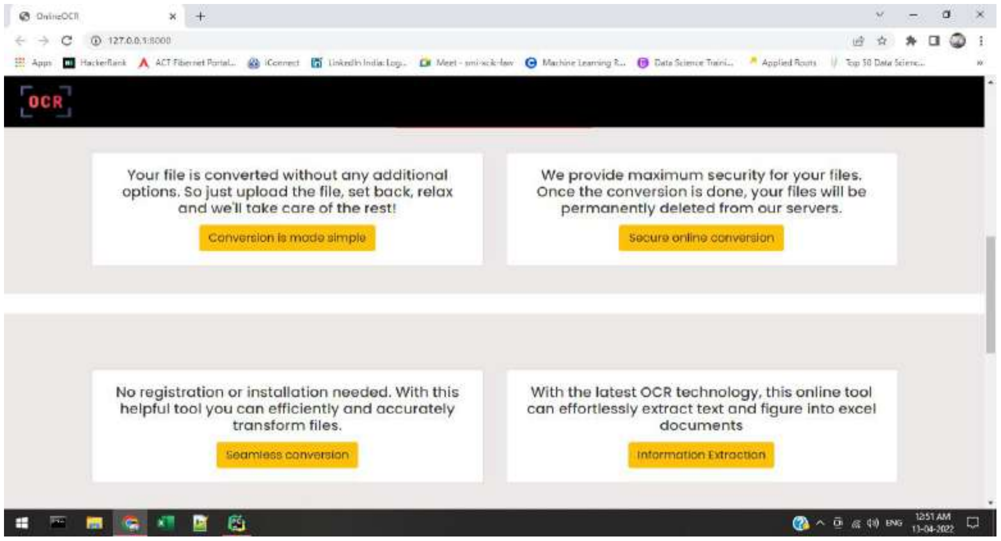


We make CONVERSION easy.

If you ever spent a hours searching for contents , words or other information in a collection of documents, OCR may be your NEW BEST FRIEND. Enabling entirely new ways to process documents that can eliminate human touches, costly data entry there by reducing processing time. From paper based information to handwritten recognition everything is done in seconds.

Screenshot 5.1 Homepage of the website

5.1.2 WEBSITE FEATURES



Your file is converted without any additional options. So just upload the file, set back, relax and we'll take care of the rest!

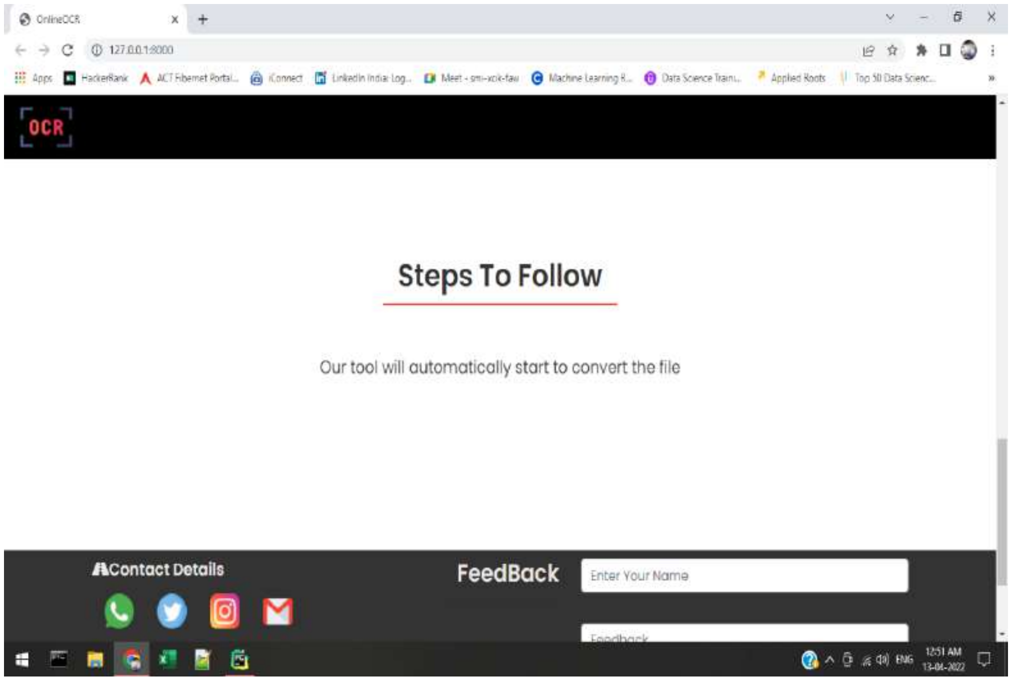
We provide maximum security for your files. Once the conversion is done, your files will be permanently deleted from our servers.

No registration or installation needed. With this helpful tool you can efficiently and accurately transform files.

With the latest OCR technology, this online tool can effortlessly extract text and figure into excel documents

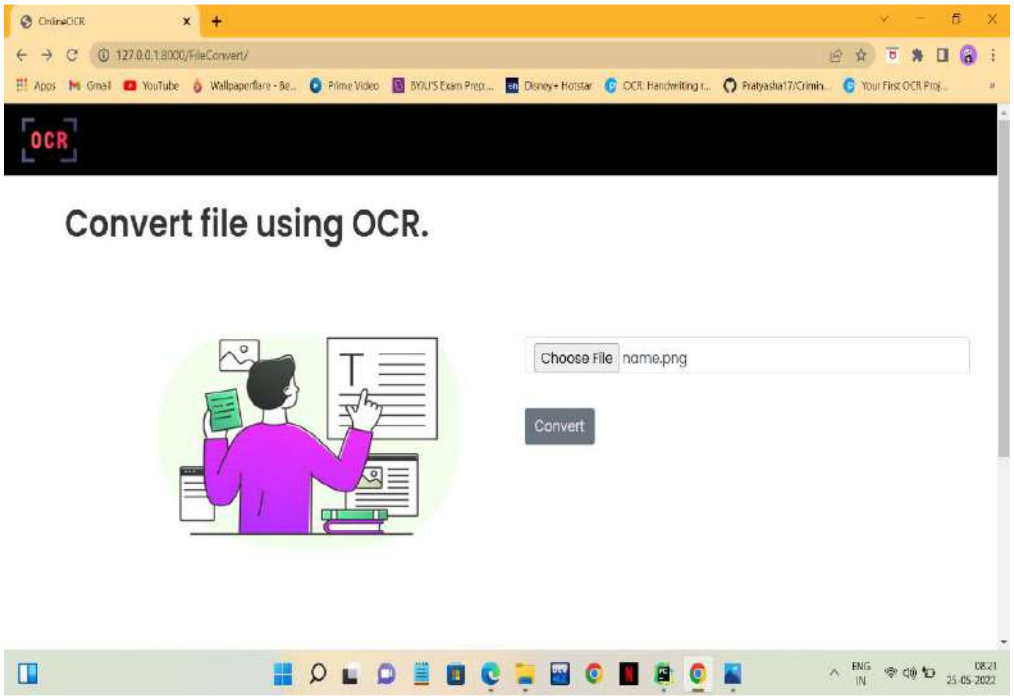
Screenshot 5.2 Features of the website

5.1.3 HOW TO UPLOAD A DOCUMENT



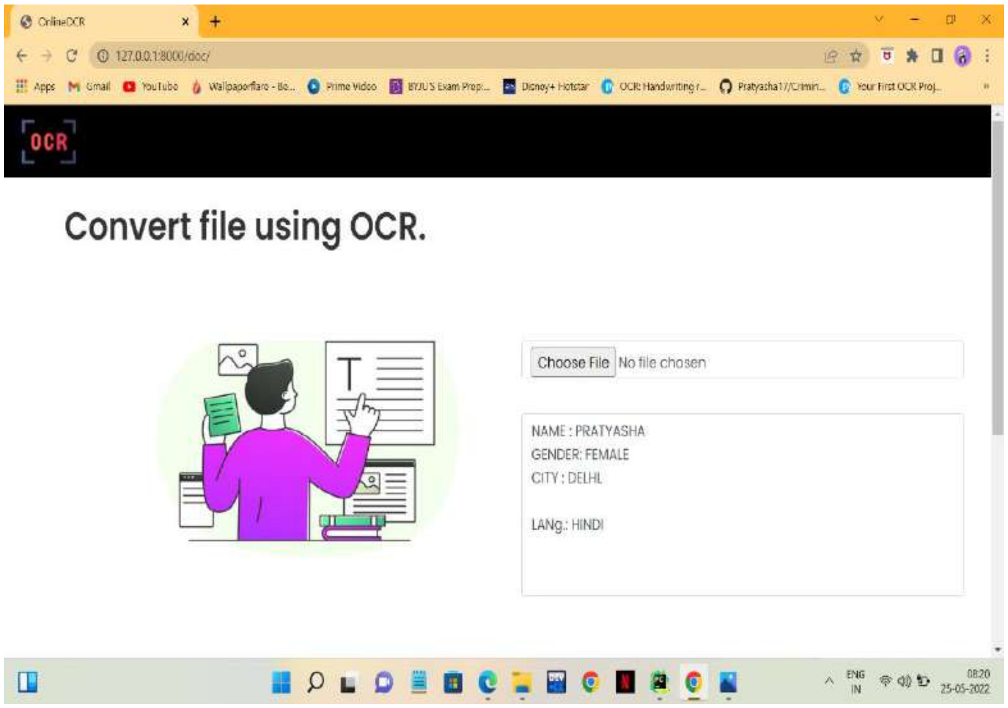
Screenshot 5.3 Steps to load document

5.1.4 CHOOSING FILE FOR CONVERSION 1



Screenshot 5.4 Taking image 1 from device

5.1.5 CONVERTED TEXT RESULTS 1

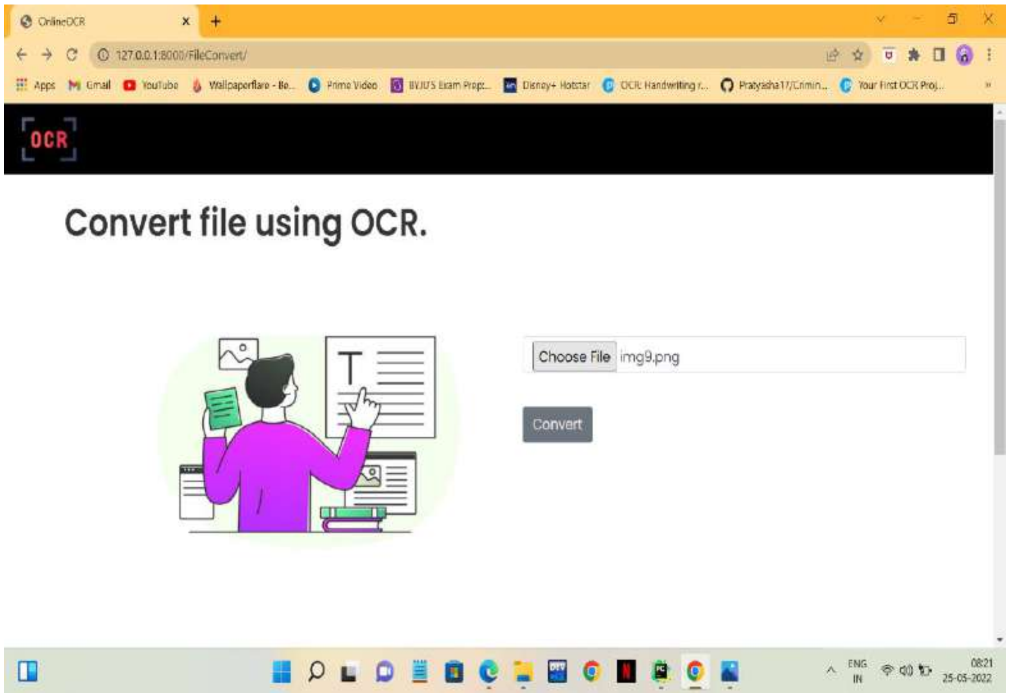


NAME : PRATYASHA
GENDER: FEMALE
CITY : DELHI

LANG.: HINDI

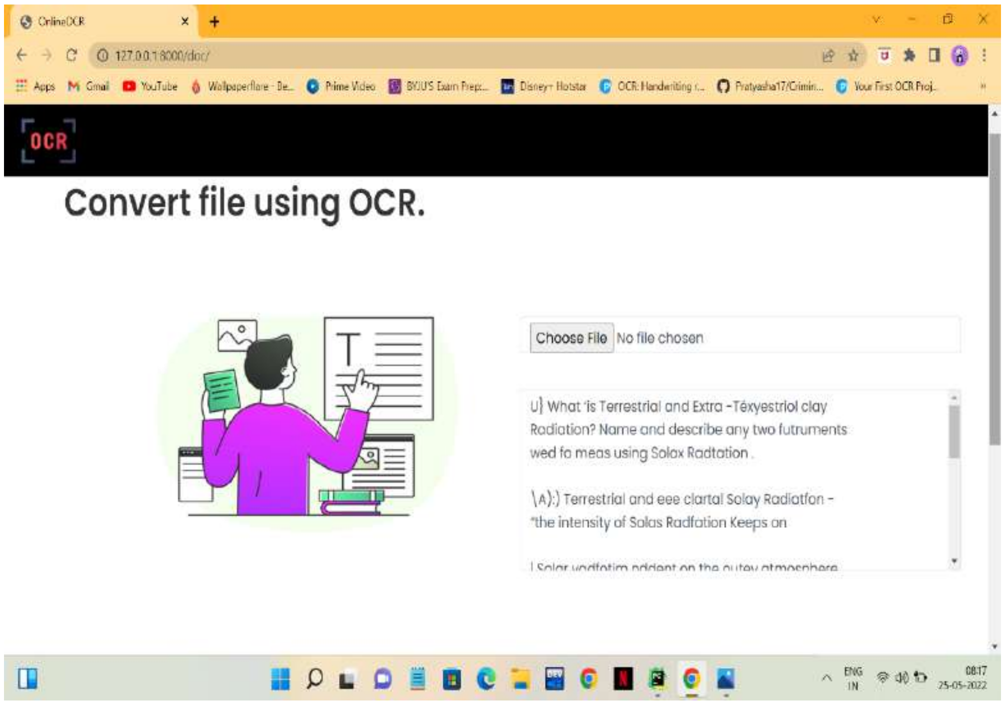
Screenshot 5.5 Result from image 1

5.1.6 CHOOSING FILE FOR CONVERSION 2



Screenshot 5.6 Taking image 2 from device

5.1.7 CONVERTED TEXT RESULTS 2



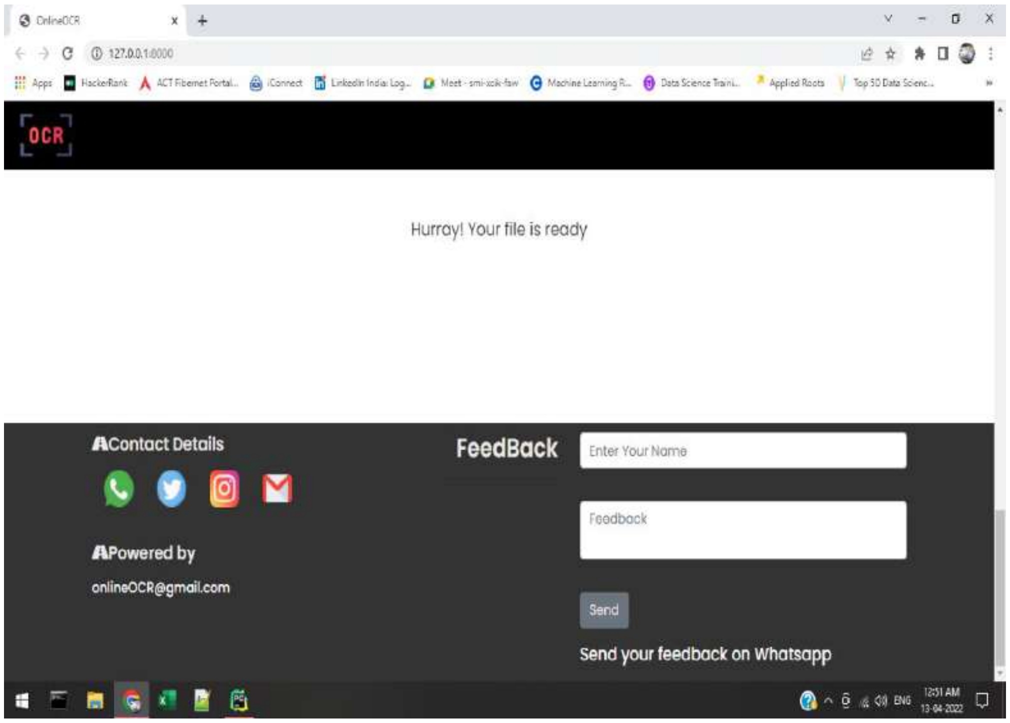
U} What 'is Terrestrial and Extra -Téxyestriol clay Radiation? Name and describe any two futruments wed fo meas using Solax Radtation .

\A):) Terrestrial and eee clartal Solay Radiatfon - "the intensity of Solas Radfation Keeps on

I Solar vadfatim nddent on the outey atmosphere

Screenshot 5.7 Result from image 2

5.1.8 CONTACT DETAILS AND FEEDBACK PROVISION



Hurray! Your file is ready

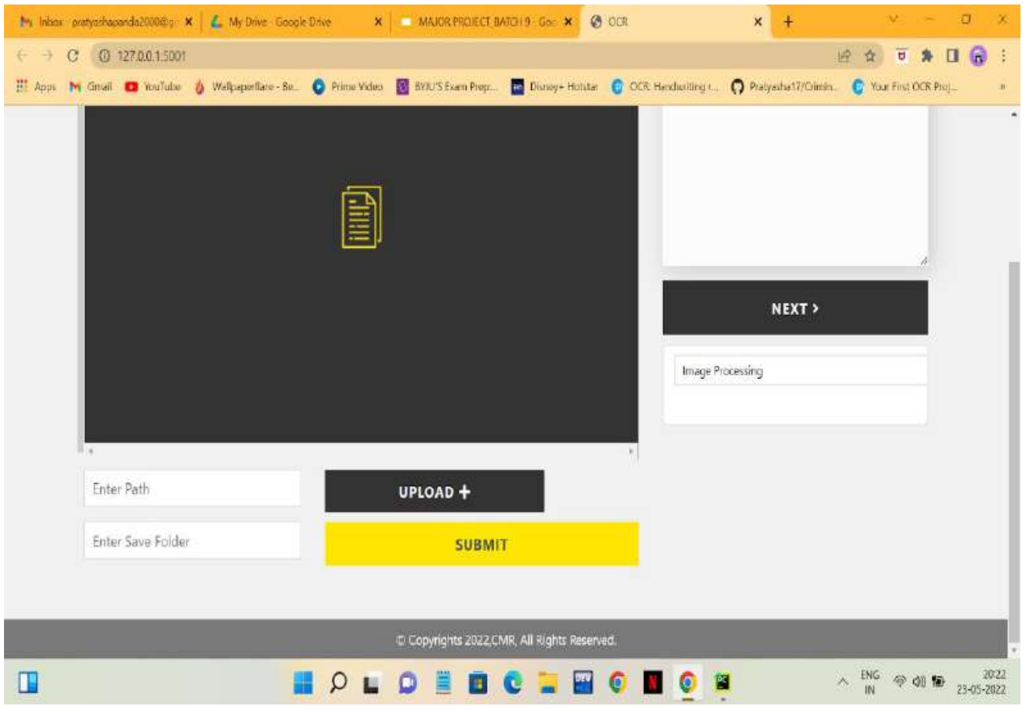
onlineOCR@gmail.com

Send your feedback on Whatsapp

Screenshot 5.8 Contact and feedback details

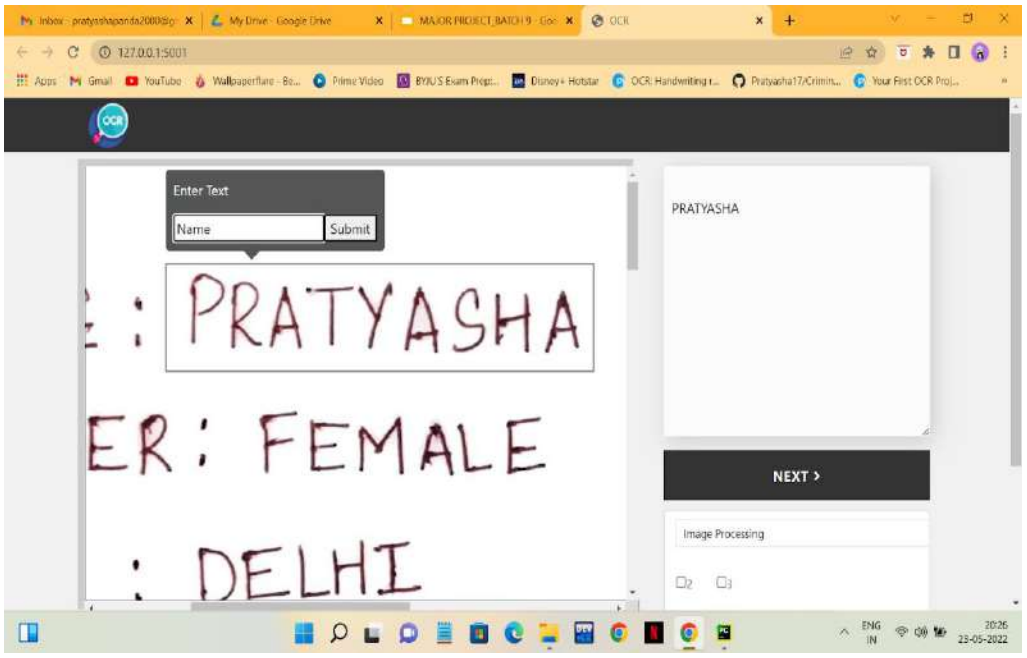
5.2 WEBITE FOR TEMPLATE SELECTION

5.2.1 PATH FOR CONVERSION AND SAVING DATA



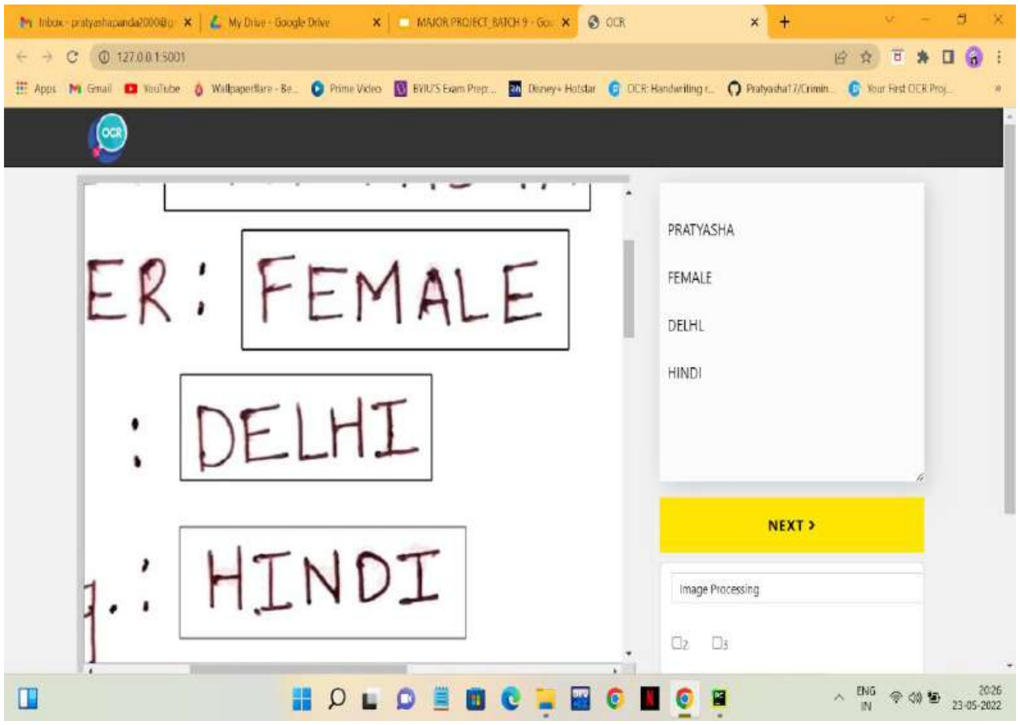
Screenshot 5.9 Providing path

5.2.2 ADDING COLUMN NAMES



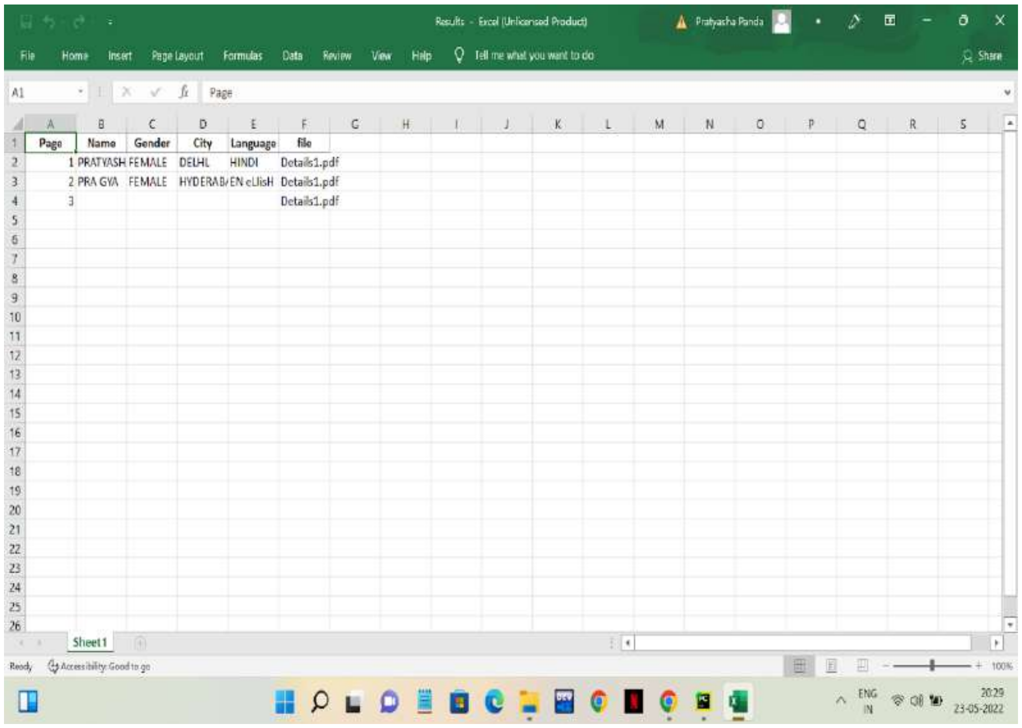
Screenshot 5.10 Giving column name

5.2.3 SELECTING REQUIRED TEMPLATES


PRATYASHA
FEMALE
DELHL
HINDI

Screenshot 5.11 Details captured

5.2.4 EXCEL SHEET OF SELECTED TEMPLATES


Page	Name	Gender	City	Language	file
1	PRATYASH	FEMALE	DELHL	HINDI	Details1.pdf
2	PRA GYA	FEMALE	HYDERAB/	EN eLlisH	Details1.pdf
3					Details1.pdf

Screenshot 5.12 Resulted excel sheet

6. TESTING

The software system needs to be checked for its intended behaviour and direction of progress at each development stage to avoid duplication of efforts, time and cost overruns, and to assure completion of the system within stipulated time.

6.1 SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

6.2 TYPES OF TESTS

In order to make sure that the system does not have errors, the different levels of testing strategies that are applied at differing phases of software development are:

6.2.1 WHITE BOX TESTING

In this the test cases are generated on the logic of each module by drawing flow graphs of that module and logical decisions are tested on all the cases. It has been uses to generate the test cases in the following cases:

- Guarantee that all independent paths have been Executed.
- Execute all logical decisions on their true and false Sides.
- Execute all loops at their boundaries and within their operational bounds
- Execute internal data structures to ensure their validity.

6.2.2 BLACK BOX TESTING

In this strategy some test cases are generated as input conditions that fully execute all functional requirements for the program. This testing has been uses to find errors in the following categories:

- Incorrect or missing functions
- Errors in data structure or external database access
- Performance errors
- Initialization and termination errors.

6.2.3 UNIT TESTING

Unit Testing is done on individual modules as they are completed and become executable. It is confined only to the designer's requirements. Unit testing is essentially for the verification of the code produced during the coding phase and the goal is to test the internal logic of the module/program. In the project, the unit testing is done during the coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested whether they are rightly connected or not.

6.2.4 INTEGRATION TESTING

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation modules and insertion modules.

6.2.5 VALIDATION TESTING

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

6.3 TEST CASES

6.3.1 TEST CASE FOR LOGGING INTO WEBSITE

Table 6.1: Test case 1

When a USER tries to enter an incorrect Link then it displays an error message “NOT A VALID URL link”.

Test case name	Test case	Output
URL link	User enter the URL link	Valid URL
URL link	User enter the URL link	Invalid URL

6.3.2 TEST CASE FOR CHOOSING FILE

Table 6.2: Test case 2

When a USER tries to upload a file of wrong format, then this results in displaying an error message “INVALID TYPE REQUESTED”.

Test case name	Test case	Output
PDF, JPEG or PNG file	Chooses the file	Successfully taken
Not a PDF, JPEG or PNG file	Chooses the file	Invalid type

6.3.3 TEST CASE FOR SELECTING CONVERSION TYPE

Table 6.3: Test case 3

When the USER chooses a pdf file but selects “Image processing”, then this results in an error message “SOMETHING WENT WRONG”.

Test case name	Test case	Output
PDF file conversion	Chooses PDF processing	Processing done successfully
PDF file conversion	Chooses Image processing	Something went wrong

7. CONCLUSION

7.1 PROJECT CONCLUSION

This project helps people, organizations or industries to use it for their data collection which can read data in no time and convert unstructured data into structured data. It can work with bulk data at the same time. This will be used in courts to provide proofs which are old and can’t be read by human eye, teachers to read students answer scripts, students to read their old notes or notes from the internet, it can capture YouTube videos or any videos and read from it and so on.

The constraints are met and overcome successfully. The system is designed as it was decided in the design phase. The project gives a good idea on developing a full-fledged application satisfying the user requirements.

The system is very flexible and versatile. Validation checks induced have greatly reduced errors. Provisions have been made to upgrade the software. The application has been tested with live data and has provided a successful result. Hence the software has proved to work efficiently.

7.2 PROJECT SCOPE

Technology has transformed how one sees the images around and makes meaning out of it.

Machines are trained to learn from humans which will lead them to -

- Convert text forms for the crucial data hidden centuries ago written in different languages which humans cannot read.
- OCR’s can be introduced in different languages for people to access easily.
- It will be able to recognize even the blurriest pictures with high efficiency which will ease the human work.
- It can be used in the advancement of robots or to make robots use OCR technique to read papers.

8. BIBLIOGRAPHY

8.1 GITHUB REPOSITORY LINK

https://github.com/Pratyasha17/OCR-of-handwritten-forms.git

8.2 REFERENCES

- Optical Character Recognition by Stephen V. Rice, George Nagy & Thomas A. Nartker
- Fundamentals of Neural Networks by Laurene Fausett
- The Complete Reference HTML & CSS
- HTML Black Book
- Python by Mark Lutz
- Software Engineering by Roger Pressman

8.3 WEBSITES

- https://creately.com/diagram-type/use-case/
- https://lucid.app/documents#/templates?folder_id=home
- https://pyimagesearch.com/2020/08/24/ocr-handwriting-recognition-with-opencv-keras-and-tensorflow/
- http://www.python.com/
- http://www.djangoproject.com/
- http://www.google.com/
- http://stackoverflow.com/
- https://www.geeksforgeeks.or