CHAPTER – 1
INTRODUCTION

1.1 OVERVIEW

At the start of the digital revolution, when most of the printed information was being uploaded on the web, manual data entry of such humongous printed data (like those of newspaper collections) became a task that required time and patience. This data entry task was also prone to human errors. The outcome of this problem was in the form of the birth of OCR. It was invented in the early eighties. OCR is now mature enough to grab characters and words from images to extract meaningful information. This technology has now attained a near-perfect text detection accuracy.

Optical Character Recognition (OCR) technology has been around for almost a century. It's a great solution for converting handwritten text or text images into machine-readable digital format. This digitization solution automates the data extraction which further is used for data collection, processing, and analysis. In today's age, an increase in demand for digitization has fueled massive growth in almost every industry. It is used to read text from images and converting them into text data for digital content management across many industries. It is mainly used as a substitute for data entry and also for information gathering, analysis purposes, and various other purposes.

1.2 PURPOSE

The main benefit of optical character recognition (OCR) technology is that it simplifies the data-entry process by creating effortless text searches, editing and storage. OCR allows businesses and individuals to store files on their computers, laptops and other devices, ensuring constant access to all documentation.

OCR technology is used to convert images containing written text (typed, handwritten, or printed) into machine-readable text data. It is widely used as a form of data reading/entry/update from printed records.

CHAPTER – 2

LITERATURE SURVEY

2.1 Existing Problem

Manual Typing:

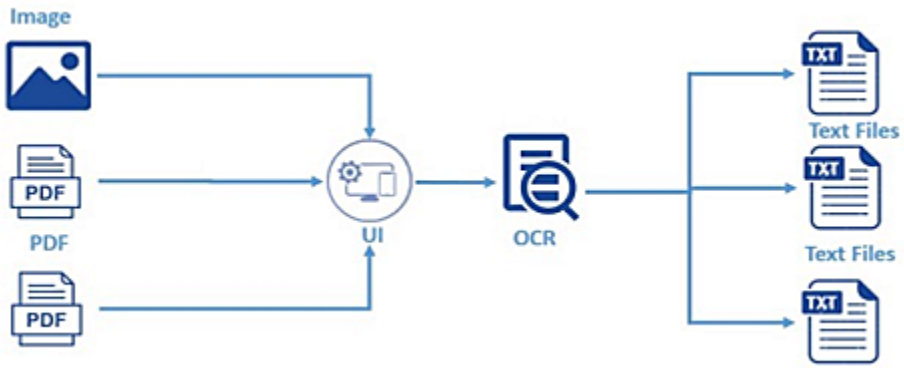
One should manually type the data in the file into another file if he wants to maintain a copy of that file. If the data is too large it will take a lot of time to complete the process

2.2 Proposed Solution

Optical Character Recognition (OCR) creates a digital version of a printed, typed or handwritten document that computers can read. It avoids the need to manually type or enter text. Why not just take a photo? Because you wouldn’t be able to edit anything or search the text because it would just be an image. Scanning the document and running OCR software quickly turns the file into something you can edit and search.

CHAPTER – 3
THEORITICAL SURVEY

3.1 Block diagram



- This image is shows that any image or pdf that coverts into text file

  For Example:

  Take an image in text format that converts into text file and also it creates a pdf.

3.2 Hardware/Software Designing

Necessary Installations:

To complete this project you should have the following packages and Softwares

- **Python IDE**- For programming
- **pytesseract** -OCR package in python
- **pdf2image**- Converting PDF to Image
- **tesseract-ocr execution file** -Backend used for pytesseract
- **poppler**-Supporting file for pdf2image package
- **Flask**-Build web application

CHAPTER-4
EXPERIMENTAL INVESTIGATION

What does OCR mean?

Optical character recognition (OCR) technology is a business solution for automating data extraction from printed or written text from a scanned document or image file and then converting the text into a machine-readable form to be used for data processing like editing or searching.

OCR solutions improve information accessibility for users

A common application of OCR technology is the automated conversion of an image based PDF, TIFF or JPG into a text based machine-readable file. OCR-processed digital files, such as receipts, contracts, invoices, financial statements and more, can be:

- Searched from a large repository to find the correct document
- Viewed, with search capability within each document
- Edited, when corrections need to be made
- Repurposed, with extracted text sent to other systems

How automated OCR capabilities for data entry benefits business operations and workflows

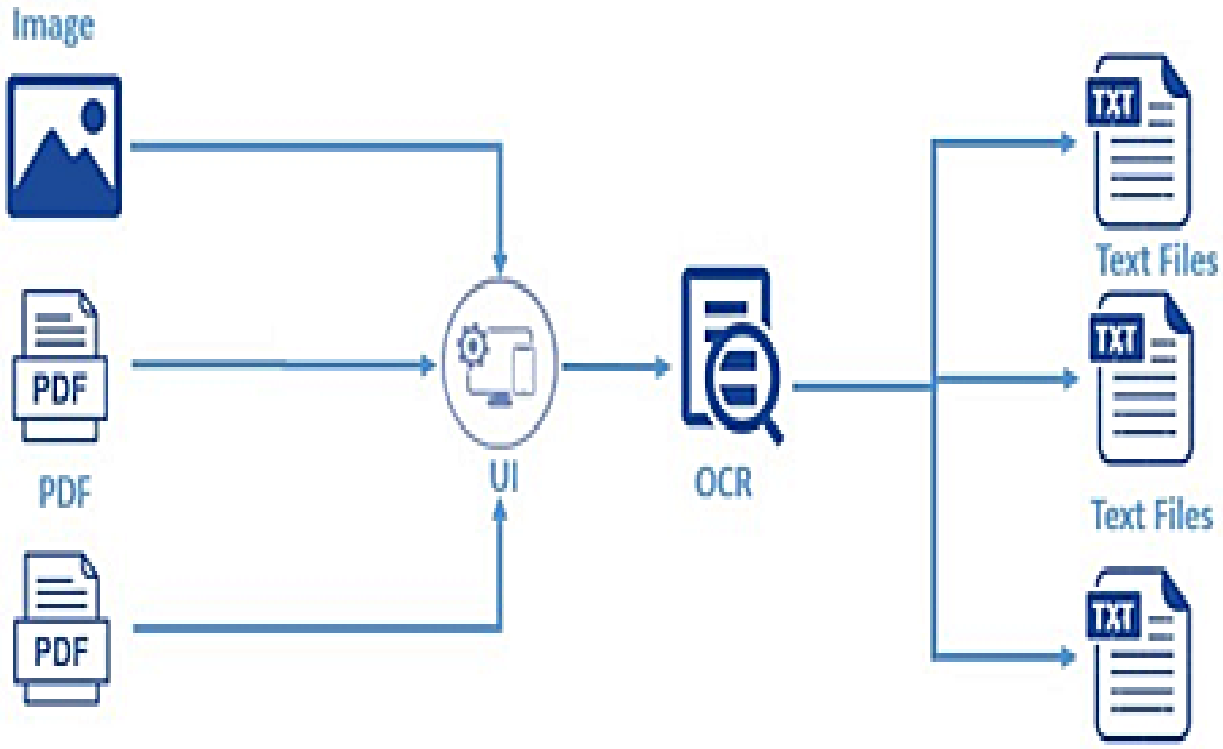
Businesses that employ OCR capabilities to convert images and PDFs (typically originating as scanned paper documents) save time and resources that would otherwise be necessary to manage unsearchable data. Once transferred, OCR-processed textual information can be used by businesses more easily and quickly.

The benefits of OCR technology to businesses include:

- Elimination of manual data entry
- Resource savings due to the ability to process more data faster and with fewer resources
- Error reductions
- Reallocation of physical storage space
- Improved productivity

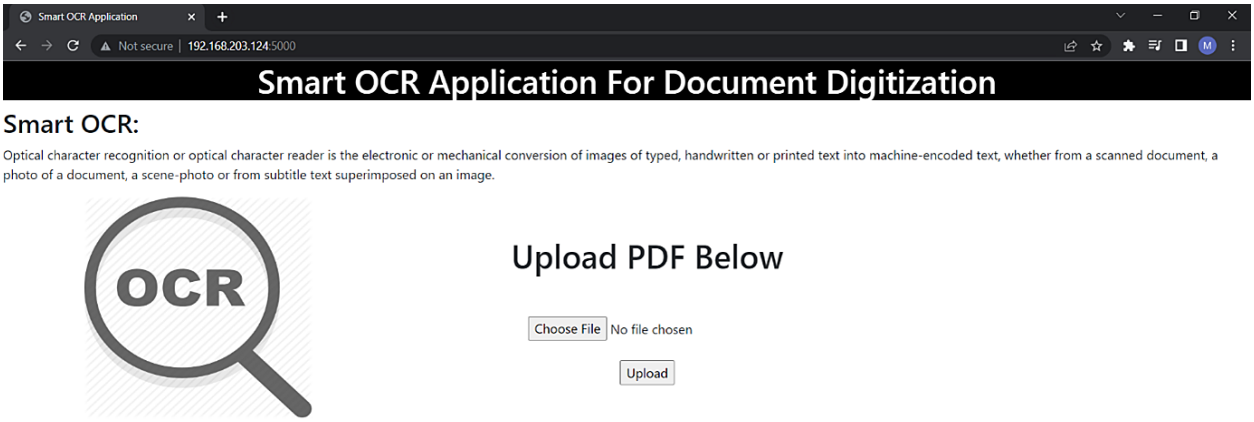
CHAPTER-5

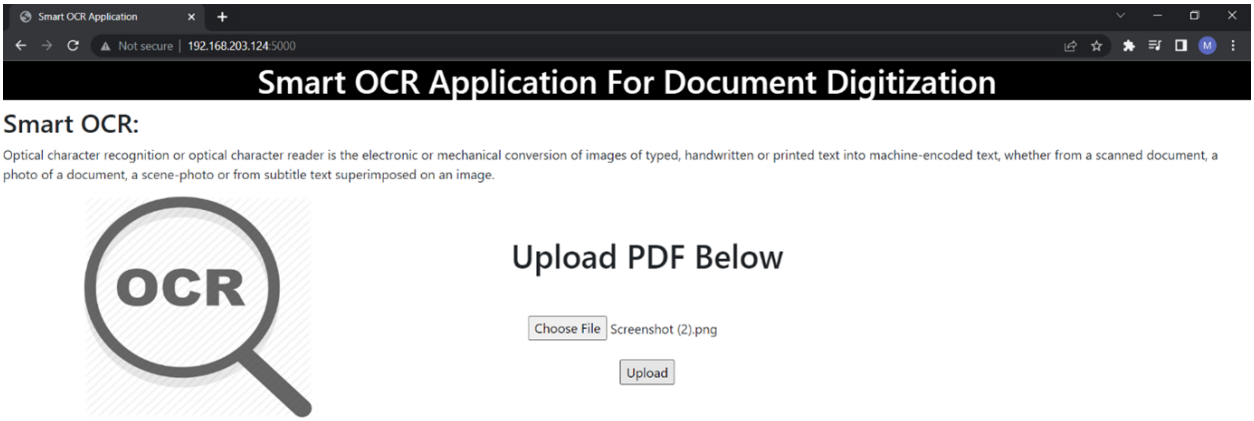
FLOWCHART



CHAPTER-6

RESULT





As shown in the figure, it is the output screen of the project. We have to choose a file/ Image and after that select upload it will show the path of the output.

CHAPTER-7

ADVANTAGES AND DISADVANTAGES

7.1 Advantages

- OCR can be readable with high degree of accuracy.
- A paper-based form is often became an electronic form which is straight forward 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.
- This process is much faster as compared to the manual typing the information into the system.

7.2 Disadvantages

- OCR systems are expensive.
- The quality of the image can be lost during this process.
- 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.

CHAPTER-8

APPLICATIONS

- Data entry
- Text entry
- Process automation
- Aid for visually impaired people
- Automatic plate numbers readers
- Automatic cartography
- Signature verification and identification

CHAPTER-9

CONCLUSION

This project aims at creating an application form where the user can upload a pdf document/ Image containing text, the document is analyzed by an Optical character recognition (OCR) to extract text from it. The extracted text is again saved in a text document in the local drive.

CHAPTER-10

FUTURE SCOPE

We have created an application where the user can upload a pdf document/ Image containing text, the document is analyzed by an Optical character recognition (OCR) to extract text from it. This can be extended in the future by including other document formats.

CHAPTER-11

BIBILOGRAPHY

- **https://towardsdatascience.com/implementing-optical-character-recognition-ocr-using-pytesseract-5f42cf62ddcc**
- **https://pyimagesearch.com/2020/09/07/ocr-a-document-form-or-invoice-with-tesseract-opencv-and-python/**
- **analyticsvidhya.com/blog/2022/06/text-detection-using-craft-text-detector/**

APPENDIX

SOURCE CODE:

```
from __future__ import division, print_function
from flask import Flask, request, render_template
#from werkzeug import secure_filename
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer
import numpy as np
import cv2
from PIL import Image
import pytesseract
import sys
import img2pdf
from pdf2image import convert_from_path
import os
pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files (x86)\Tesseract-OCR\tesseract.exe'
import os.path
import glob
import random
app = Flask(__name__, static_url_path='')
@app.route('/')
def home():
    return render_template('index.html')
@app.route('/upload', methods=['POST'])
```

```
def upload():
    if request.method == 'POST':
            f = request.files['filename']
            basepath = os.path.dirname(__file__)
            file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename))
            f.save(file_path)
            if ('.jpg' in f.filename) or ('.png' in f.filename):
                # opening image
                image = Image.open(file_path)
                # storing pdf path
                file_path = os.path.join(basepath, 'uploads', secure_filename(f.filename)[:-3]+"pdf")
                # converting into chunks using img2pdf
                pdf_bytes = img2pdf.convert(image.filename)
                # opening or creating pdf file
                file = open(file_path, "wb")
                # writing pdf files with chunks
                file.write(pdf_bytes)
                # closing image file
                image.close()
                # closing pdf file
                file.close()
            PDF_file = file_path
            pages = convert_from_path(PDF_file)
            image_counter = 1
            for page in pages:
```

```
            filename = "page_"+str(image_counter)+".jpg"
            page.save(filename, 'JPEG')
            image_counter = image_counter + 1
        filelimit = image_counter-1
        # Creating a text file to write the output
        basepath = os.path.dirname(__file__)
        file_path2 = os.path.join( basepath, 'outputs', "output"+str(random.randint(1,
100000))+".txt")
        f = open(file_path2, "a")
        for i in range(1, filelimit + 1):
            filename = "page_"+str(i)+".jpg"
            text = str(pytesseract.image_to_string(Image.open(filename)))
            text = text.replace('-\n', '')
            f.write(text)
        f.close()
    return 'Output Text File is saved at '+file_path2
port=os.getenv('VCAP_APP_PORT','5000')
if __name__=='__main__':
    app.secret_key=os.urandom(12)
    app.run(debug=True,host='0.0.0.0',port=port)
```