HAPPY HEALTH: Intelligent Prescription Image Recognition and Medication Management System with Voice-Assistant and nearby Hospital Search

Simhadri Sevitha
211AI036
Dept of IT, NITK

Kesanam Ashinee
211AI023
Dept of IT, NITK

Mauli Mehulkumar Patel
211AI024
Dept of IT, NITK

ABSTRACT: Recognition of handwritten text is still a research need in the field of optical character recognition (OCR). This study offers a simple and effective method for creating a system that can recognize handwritten text, offering a creative approach for digitizing and automating the prescription administration process. Modern optical character recognition (OCR) technology and Open-CV are used by the system to convert handwritten doctor's prescriptions into text that can be read by machines, identify the medication mentioned in the prescription, and provide information on nearby hospitals within a predetermined range.Additionally, the system also offers a visual depiction of the medication to help patients recognize it. One possible future enhancement of the system could include the implementation of voice recognition, so that patients who have difficulty reading can dictate their prescriptions.It also extracts information regarding the dosage of the prescribed medication, including how many tablets should be taken each day. This makes it possible for patients to understand and effectively follow their drug schedule.Modern machine learning methods are used by the proposed system to precisely recognize the text in the prescription and extract the necessary details about the recommended drug. Additionally, the system uses location data to give patients a thorough list of close-by hospitals, obviating the need for manual search and delivering an intuitive experience. In order to help patients recognize their medication, the system also offers a visual depiction of the medication given, such as an image of the medication or a medicine sheet.By drastically lowering the likelihood of errors, speeding the workflow, and giving patients a more convenient and effective experience, the introduction of this technology has the potential to completely transform the prescription management process. It is a significant step toward the automation and digitization of the healthcare sector and has enormous potential to enhance patient outcomes.

I. INTRODUCTION

The art of handwriting allows each person to uniquely convey their ideas on paper. Depending on the individual, it might vary considerably. Specifically, when discussing a doctor's busy

schedule, more consultations are scheduled quickly, and the diagnosis is given more importance than the prescription's handwriting. As a result, they frequently need better handwriting, making it sometimes challenging to read the prescription and recognize the drugs and their possible dosages. It is challenging for patients and young pharmacists to distinguish the doctor's handwriting. The outcomes from drug errors range from no apparent symptoms to death. It can sometimes result in a new ailment that is either transient or permanent, such as itchiness, rashes, or skin deformity. Despite being rare, drug mistakes can seriously harm individuals. A study from Texas, USA, that is cited mentioned a case where the cardiologist prescribed 20 mg of a drug called Isordil, which was then read as Plendil by the pharmacist. The dosage prescribed exceeded the maximum daily dosage of 10 mg for Plendil, causing severe consequences to the patient. This medical overdose then led to his eventual death. Though not all cases are fatal, these still strengthen the assertion that the legibility of a doctor's handwriting in medical prescriptions should be treated with utmost importance.

II. IMPLEMENTATION

We primarily use easy OCR to read the image from the predefined dataset and preexisting trained model. Moreover, we also use the result derived from the above code to derive the coordinates for the various text dispersed among the image. During this process, we also print the text from the image. Segmentation is also implemented in this part, and rectangles are created around the image's recognizable text. Further down, the widget following the above code asks for the user's input regarding the medicine within the image. The recognized text within the user inputted is used as the input for the medicines database. Hence, the image of the medicine is displayed. Using the disposable dataset containing the hospitals and using the current location coordinates, the nearby hospitals with emergency services are displayed on the terminal. An adjustable parameter of the radius is also allotted to define the nearby radius. As stated above, easyocr is used for text recognition from the image and segmentation of the image. OpenCV is a library of Python bindings designed to solve computer vision problems. It is used in our code to load the prescription from the file. Pandas are used for the cv handling inside the files. We uploaded the nearby hospital's information and medicine dataset as a csv file. Python Imaging Library (PIL) is also used for text extraction from the prescription. GeoPy is a Python library that we used for geographical calculations regarding nearby emergency services. BytesIO implements read and write bytes data in memory in the code. The Base64 encoding is used in our project to convert bytes that have binary or text data into ASCII characters.From matplotlib.widgets, we import RectangleSelector, to allow the user to select the medicine part of the prescription.With the help of the Python glob module, we can search for all the path names which are looking for files matching a specific pattern; in our case, the pattern is the file extension. Pyttsx3 that converts text to voice.

PHASE-1 : Uses the easyocr library to perform Optical Character Recognition (OCR) on an image located at the file path specified in the variable IMAGE_PATH. The OCR is performed on an image with the English language and it does not use GPU. The result is a list of tuples, where each tuple contains the bounding box coordinates of a recognized text region and the recognized text.The code then loops through the detection list and for each detection it gets the top left and bottom right coordinates of the bounding box and converts them to integers. Then it gets the text from the detection and appends it to a list. Then it draws a rectangle on the image using the OpenCV library with the top left and bottom right coordinates and a green color and thickness of 3 pixels. Finally, it displays the image using the matplotlib library.Then the code defines a function onselect() which is called when the user makes a rectangular selection on the image. It takes the click and release coordinates as input, and assigns them to the global variables x1, y1, x2 and y2. Then it prints the values of these variables.The code then opens the same image again using OpenCV and displays it using matplotlib. It creates a RectangleSelector object

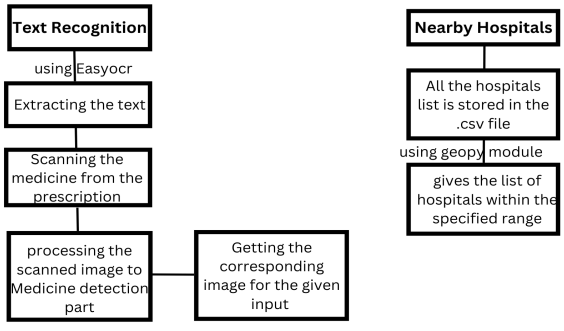
and attaches it to the current axis of the plot. It sets the 'onselect' function as the callback function to be called when the user makes a selection. The user can make a selection by clicking and dragging the mouse on the image. The function will be called with the click and release coordinates of the selection. Then it connects the "key_press_event" to the lambda function which will exit the program if the user presses the 'escape' key.Then it shows the image. Once user makes a rectangular selection, the onselect function gets called, then the image is cropped using the x1,y1,x2,y2 values obtained from onselect function. Then OCR is applied on the cropped image and the output is printed.

PHASE-2 : Uses the pandas library to read a CSV file named 'medicine_dataset.csv' and stores it in the dataframe 'med'. It then sets the maximum width of columns to None so that the images in the dataframe can be displayed fully.The code then defines a function 'get_thumbnail' that takes the path of an image file and opens it using the PIL library, resizing it to a thumbnail of size (150, 150) using Lanczos resampling.The code then defines a function 'Image_base64' that takes an image and converts it to a base64 encoded string. If the input is a file path, it calls the 'get_thumbnail' function to resize the image before encoding.The code then defines a function 'Image_formatter' that takes an image and returns an HTML img tag with the image data in base64 format.The code then calls the glob library to look for all .png files in the current directory and store it in a list. Then it creates a new column 'img_file' in the 'med' dataframe that maps the name of the medicine to the corresponding image file. It drops the rows that don't have an image file.The code then creates a new column 'Image' in the 'med' dataframe that maps

the image file to the corresponding resized image using the 'get_thumbnail' function.The code then uses the pandas 'to_html' method to convert the 'med' dataframe to an HTML table and display it in the notebook. It uses the 'Image_formatter' function to format the 'Image' column as an HTML img tag.The code then takes the text of first OCR detection and lowercases it, splits it into a list of words, and searches for each word in the 'Names' column of the 'med' dataframe. It creates a new column 'indexes' that stores the index of the first occurrence of each word. It then finds the maximum index and filters the 'med' dataframe to only include rows where the index is equal to the maximum index.The code then again uses the pandas 'to_html' method to convert the filtered 'med' dataframe to an HTML table and display it in the notebook using the 'img_file' column and 'Image_formatter' function.Finally, the code reads the image file of the first row of the filtered dataframe using OpenCV and displays it using matplotlib.

PHASE-3 : Uses the pandas library to read a CSV file named 'hospitals.csv' and stores it in the dataframe 'df'.It sets the coordinates of a location as a tuple (74.7943, 13.0108) , which is the coordinates of NITK, and radius as 5.It then defines an empty list 'distances' and uses the 'iterrows' method to iterate over the rows of the dataframe. For each row, it gets the latitude and longitude of the hospital from the dataframe, creates a tuple of the hospital location, and uses the 'distance' function to calculate the distance in kilometers between the location and the hospital location.It then appends the calculated distance to the 'distances' list.It then creates a new column 'distance' in the dataframe that contains the distances calculated above, and filters the dataframe to only include rows where the distance is less than the radius.Finally, it iterates over the filtered dataframe using the 'iterrows' method and for each row it prints the name, address and longitude of the hospital.

PHASE-4: This code uses the pyttsx3 library to initialize a Text-to-Speech (TTS) engine.It then gets the list of available voices from the engine using the 'getProperty' method and 'voices' attribute.It then sets the voice of the engine to the second voice in the list using the 'setProperty' method and 'voice' attribute.It then uses the 'say' method to provide the text that the engine should speak, which is the extracted medicine name from the prescription.Finally, it uses the 'runAndWait' method to run the engine and play the spoken text.



III. CONCLUSIONS

A proposed method of recognizing handwritten characters has been developed and tested. A comparison with pertinent studies has been offered. In-depth instructions for translating handwritten doctor's prescriptions to machine-readable text, identifying the medicine mentioned, providing dosage information, a graphic representation of the medication, and information about nearby hospitals have all been provided by this study.Additionally, the system also offers a visual depiction of the medication to help patients recognize it. One possible future enhancement of the system could include the implementation of voice recognition, so that patients who have difficulty reading can dictate their prescriptions. In order to automate the prescription management process and increase its accuracy and efficiency, this system makes use of contemporary OCR technology and Open-CV. The use of this technology has the potential to totally alter the healthcare industry by reducing errors, enhancing workflow, and providing patients with a more straightforward and accessible experience. With the help of technology, patients now have instant access to the medical equipment and services they need, which not only solves the problem of managing prescriptions. This research emphasizes how this strategy has the potential to enhance patient outcomes and shift healthcare toward a patient-centric model.

IV. FUTURE WORK

1. Personalized dosage recommendations: The system can be enhanced to provide personalized dosage recommendations based on the patient's age, weight, and medical history
2. Implementing AI chatbot: an AI chatbot can be implemented to interact with patients and guide them through the process.
3. Adding barcode scanner functionality: The system can be enhanced by adding a barcode scanner functionality to scan the barcode on the medicine and fetch the information about it.
4. Automated refill reminders: The system can be configured to send automated refill reminders to patients, reminding them to refill their prescriptions before they run out

V. REFERENCES

https://towardsdatascience.com/rendering-images-inside-a-pandas-dataframe-3631a4883f60

https://ieeexplore.ieee.org/document/9073521

https://arxiv.org/ftp/arxiv/papers/2210/2210.11666.pdf

https://medium.com/mlearning-ai/tesseract-vs-keras-ocr-vs-easyocr-ec8500b9455b

https://www.analyticsvidhya.com/blog/2021/06/text-detection-from-images-using-easyocr-hands-on-guide/

https://geopy.readthedocs.io/en/stable/#module-geopy.units

https://pyimagesearch.com/2021/01/19/crop-image-with-opencv/