

Text Extraction from Images and Audio Generation Using NLP

¹Purvish Kodi, ²Narahari Raghava, ³Dr. V. Pandimurugan

^{1,2}Student, ³Assistant Professor, Department of Networking and Communication, SRM Institute of Science and Technology, Kattankulathur, India.

¹pk9668@srmist.edu.in, ²nc5664@srmist.edu.in, ³pandimuv@srmist.edu.in

Abstract - Digitization allows us to immortalize a physical entity by creating a digital representation of it on our devices. It saves us time in manually sifting through physical storage units such as albums and notebooks and provides us with programs to manage and secure our data. We often take images of Receipts or Invoices, Identity Cards, and nutritional labels to save a copy of their details. This can be taken a step further by automating the process of information extraction and documentation. Advancements in computer vision have provided us with the expertise to create tools for text detection and extraction. But it is still an ongoing challenge because documents with unstructured layouts, poor image quality, and noise around the text yield very low accuracy in text extraction results. Conquering this challenge would require the image to be highly enhanced through pre-processing techniques such as Brightness Correction, Contour Detection, Skewness Correction, Morphology, and Binarization. A mechanism made from the best combination of image pre-processing techniques prior to text extraction can improve text accuracy to a large extent.

Keywords — Tesseract, OpenCV, Tkinter, Python, Pillow, NTKL, Digitalization

I. INTRODUCTION

In today's day and age, an increase in demand for digitization has fueled a massive growth in technology and communication and the use of printed materials such as books and papers has significantly reduced. It is easier to organize digitized data and analyze it for various purposes with many advanced techniques like artificial intelligence etc. To translate physical and handwritten documents into digital copies, optical character recognition (OCR) has come into the sight of researchers and since its first advent it has undergone significant changes in methodology and made considerable progress towards its goal of text detection and extraction.

Text extraction from images involves using optical character recognition (OCR) techniques to extract text from images. This can be useful in cases where the text in an image is the only available source of information, such as in scanned documents.

Audio generation using natural language processing (NLP) involves using machine learning models to generate spoken language from written text. This can be useful in cases where it is desired to have the written content of a document read aloud, or to create a more accessible version of written content for those who may have difficulty reading involves using machine learning models to generate spoken language from written text. This can be useful in cases where it is desired to have the written content of a document read aloud, or to create a more accessible version of written content for

those who may have difficulty reading many advanced techniques like artificial intelligence etc. To translate physical and handwritten documents into digital copies, optical character recognition (OCR) has come into the sight of researchers and since its first advent it has undergone significant changes in methodology and made considerable progress towards its goal of text detection and extraction.

II. MODULE DESCRIPTION

A. Module 1- Image Pre-Processing

The Pre-processing module takes in an image and aims to eliminate the challenges that may occur while detecting text, caused due to noise, blurring effects, uneven lighting, skewness. In this stage, the image input scanned and uploaded by the user is processed to rotate to correct skew and detect and crop the document out of the background, remove noise, sharpen the image, and remove the blurriness that may affect the image during extraction. The preprocessed image is then sent through for text-extraction.

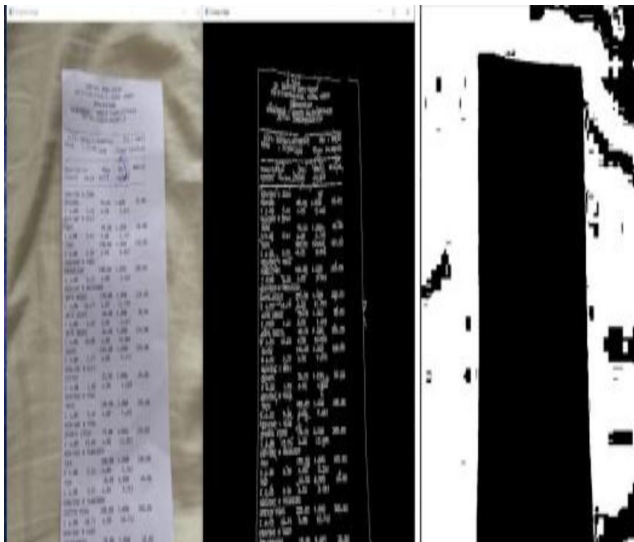


Fig 2.1 Edge Detection

When an image of a receipt is taken, it is highly unlikely that all images are taken in high definition, and with the receipts containing no creases. These images might not just contain the receipt, but also the surrounding area the receipt was placed on or held against. This will decrease the chances of the text being recognized. So first we find a way for the edges of the receipt to be detected and cropped off the image. This is a task that must be handled by the software as the application is aimed towards decreasing the work of the user. With OpenCV, this task can be accomplished if proper steps are followed. The edges of

the Once the dimensions of the receipt are saved, they are sent as input to the preprocessing methods to enhance the quality of the image. For the text on the receipt to be recognized by the OCR engine, the images need to be preprocessed. Most OCR engines work well on Black & White images. Common preprocessing methods include – Gray scaling, Thresholding (Binarization) and Noise removal. Gray scaling is simply converting a RGB 20 image to a grayscale image. Noise removal is done using morphology techniques such as Blur and Dilate. Thresholding involves the assignment of pixel values in relation to a threshold value provided. Each pixel value is compared with the threshold value. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value (generally 255). OpenCV provides various thresholding options - Simple Thresholding, Adaptive Thresholding.



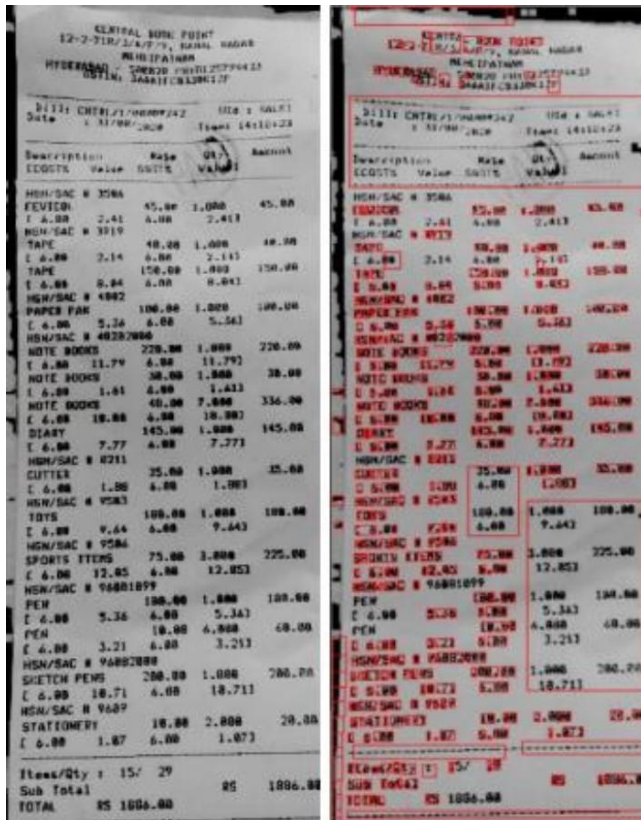
Fig 2.2 Thresholding

Once the preprocessing phase is complete, we can expect a higher chance of our text being recognized correctly by the OCR engine

B. Module 2- Text - Extraction

The next step is Optical Character Recognition (OCR). It is a widespread technology used to recognize text inside images, such as scanned documents and photos. OCR technology is used to convert virtually any kind of images containing written text (typed, handwritten or printed) into machine-readable text data. Once a scanned paper document goes through OCR processing, the text of the document can be edited with word processors. Before OCR technology was available, the only option to digitize printed paper documents was to manually re-type the text. Not only was this very time consuming, but it also came with lots of typing errors. Fortunately, there are very good 21 open-source OCR libraries. The most popular option was the Tesseract OCR engine. OCRopus was also a good option, especially for those who want to explore the inner workings of OCR. We used Tesseract-OCR, Google's open-source OCR engine. By using a method to draw boxes around the text that has been detected by the OCR engine,

as shown in Figure 3.4, we can observe that most of the text present on the receipt was detected. Here, we use pytesseract, which is a simple wrapper around Tesseract. Using pytesseract is simple: call image_to_string() to convert the image into a single formatted string. Calling image_to_data() will return individual text fragments with recognition confidence and other useful information.



Description	Value	Rate	Qty	Amount
HEM/SAC # 3586	45.00	1.000	45.00	
EVICOL	2.41	6.00	2.41	
HEM/SAC # 3919	48.00	1.000	48.00	
TAPC	2.14	6.00	2.14	
HEM/SAC # 4882	100.00	1.000	100.00	
PAPER FOR	5.36	6.00	5.36	
HEM/SAC # 4882	220.00	1.000	220.00	
NOTE BOOKS	11.79	6.00	11.79	
NOTE BOOKS	38.00	1.000	38.00	
NOTE BOOKS	48.00	7.000	336.00	
DEAT	18.00	6.00	18.00	
HEM/SAC # 4882	145.00	1.000	145.00	
CUTTER	25.00	1.000	25.00	
HEM/SAC # 4882	1.00	6.00	1.00	
HEM/SAC # 4882	100.00	1.000	100.00	
HEM/SAC # 4882	9.44	6.00	9.44	
SPORTS ITEMS	75.00	3.000	225.00	
HEM/SAC # 4882	12.85	6.00	12.85	
HEM/SAC # 4882	100.00	1.000	100.00	
HEM/SAC # 4882	5.36	6.00	5.36	
HEM/SAC # 4882	18.00	6.00	18.00	
HEM/SAC # 4882	3.21	6.00	3.21	
HEM/SAC # 4882	200.00	1.000	200.00	
HEM/SAC # 4882	18.71	6.00	18.71	
HEM/SAC # 4882	18.00	2.000	36.00	
HEM/SAC # 4882	1.87	6.00	1.87	
Items/Qty : 15/ 29				
Sub Total				85 1006.00
TOTAL				85 1006.00

Fig 2.3 Text Detection done by the OCR Engine

C. Module 3 - Post Processing with NLP

Natural Language Processing, or NLP for short, is defined as the automatic manipulation of natural language, like speech and text, by software. NLP primarily comprises of Natural Language Understanding (human to machine) and Natural Language Generation (machine to human). Our project deals with NLU. NLU aids in extracting valuable information from text such as social media data, customer surveys, and complaints. In our project we used a common technique of NLP called Named Entity Recognition (NER). This is the most basic and useful technique in natural language processing for extracting entities in text. 4 Named entity recognition (NER) identifies entities such as people, locations, organizations, dates, etc. from the text. For example, for text obtained from our receipt image, the following output can be expected: Vendor Name, Date, Total, Address. NER is generally based on grammar rules and supervised models. However, there are NER platforms such as open NLP that have pre-trained and built-in NER models. For the given entity categories above, a JSON file is created containing the extracted classes. In many cases, especially for data other than Total Amount, labels aren't found for the rest. In such a case, the model requires us to create labels for each of the inputting tokens (words or characters), which needs to be created separately. For this, we train a model with a predefined dataset containing images of receipts and their corresponding data in a JSON file. The model learns to identify what an address is, what a vendor name typically etc. In our model, we were able to successfully train it using a dataset found on Kaggle, to

identify vendor name and total amount accurately.

Module 4 - Integrating into application.

After the code to parse the receipt image is developed, the entire pipeline needs to be deployed. This was done using a Python framework called Django. Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. The web application was also built using Django. Django takes care of much of the hassle of web development, so that the developer can focus on writing the application without needing to spend time on building it from scratch. It is free and open source, has a thriving and active community, great documentation, and many options for free and paid-for support. The web application consisted of a single home website, which navigates the user to upload the image, as well as download the output text file and audio file to their local system.

A system Graphical User Interface (GUI) was also implemented to be used as an in-house application for the university. The GUI application was built with the help of Python's GUI toolkit, tkinter.

III. LITERATURE SURVEY

This subsection of the paper deals with the efforts carried out by various developers towards the core of recognizing text from different images.

Yang Zhang and Hao Zhang HaoranLi [11] described in their paper, an information extraction pipeline used for event flyers. The major steps in the pipeline include image capture and upload, image preprocessing, text detection, OCR and NLP information extraction. The paper lists several situations where a raw image could cause inaccurate results. The OCR engine would assume the picture is taken from a perpendicular upright view, but images taken from a handheld camera could contain distortions. The illumination of the image not being uniform throughout and an image containing multiple blocks of text of different sizes and colors could also affect the output. The image preprocessing methods included were edge detection, geometric correction(transformation), and Binarization.

Lavanya Bhaskar and R Ranjit [12] discuss an event planner for the brochure images, that implements text extraction by convolution followed by MSER feature extraction and Stroke width method. The event planner then directly links the event text to the google calendar for scheduling the events. However, the algorithm is not tested for event information taken from handwritten images and complex font text present in the images.

Brijesh Kumar Y. Panchal, and Gaurang Chauhan [13] proposed an implementation on the Android Application to extract using Tesseract OCR in which the following concepts are used, which are Adaptive Thresholding, Connected Component, Fine Lines, and Recognize Word. Using this Optical Character Recognition (OCR)

Technology. The Application generates text, which is printed on a clean, B/W or colorful background and then can be converted into a computer readable form ASCII. With the help of this Android Application using Tesseract OCR, the system has two ways for Text Extraction. The first one is to capture a photo while the second one uploads an image from the gallery. After that the system can proceed as per the user requirement which portion of the image they want to crop or edit. After editing the picture, it converts into the text. This Android Application is for two languages, English and Hindi.

Salvador España-Boquera, Maria J. C. B., Jorge G. M., and Francisco Z. M. [14], this paper outlines the hybrid Hidden Markov Model (HMM) used to conceive the unconstrained offline handwritten texts. The main characteristics of the recognition systems is to produce a new way in the form of preprocessing and recognition which are both based on ANNs. The preprocessing is used to clean the images and to enhance the non-uniform slant and slope correction. Whereas the recognition is used to estimate the emission probabilities.

K.Gaurav, Bhatia P. K. [15], this paper deals with assorted pre-processing techniques used for handwritten recognition which consists of different images starting from a simple handwritten document and extending its radius to complex background and diverse image intensities. The pre-processing techniques that were included are contrast stretching, noise removal techniques, normalization, and segmentation, binarization, morphological processing techniques. They concluded that no technique for preprocessing can single handedly be used to produce an image. All the techniques go hand in hand. Even though after applying all the said techniques, the accuracy of the image is not up to the mark.

II. IMPLEMENTATION

Figure 4.1 shows the flow from one sub process to another in our proposed pipeline. When the user uploads an image of their document to the interface, it goes through a series of preprocessing methods to enhance the text present on the image and the text is then detected and extracted. Useful information is then extracted from this raw text to form structured data as well as in audio format.

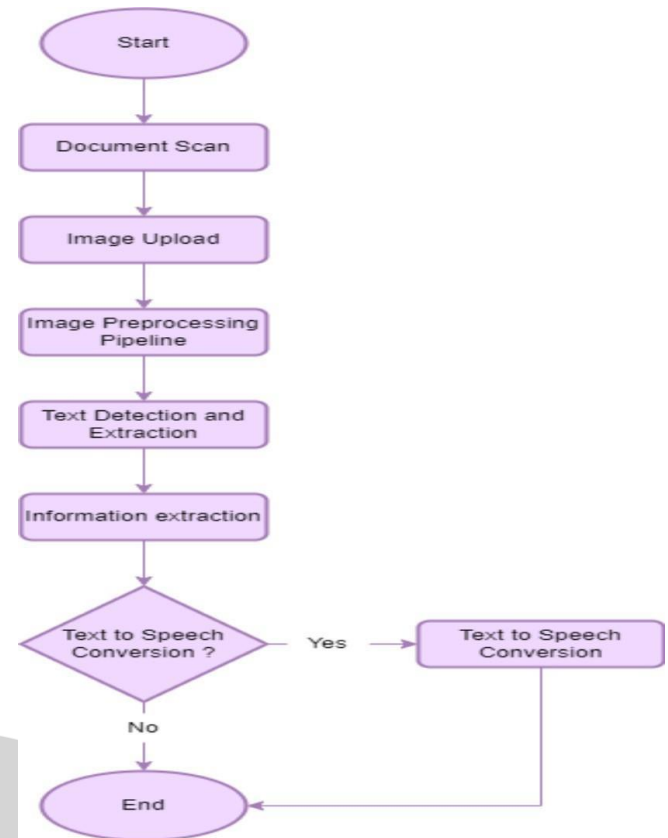


Fig 4.1. System Architecture

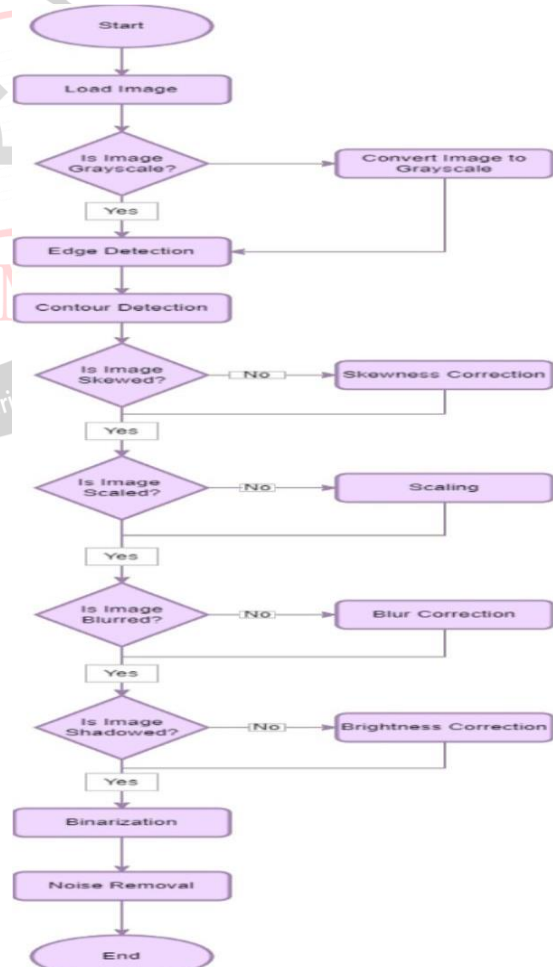


Fig 4.2. Image preprocessing pipeline

Figure 4.2, Image preprocessing pipeline describes the

various sub processes involved in the image preprocessing pipeline. The image is converted to grayscale and then corrected in case the image has a gradient brightness, is at an angle, contains a background, is blurry or contains noise.

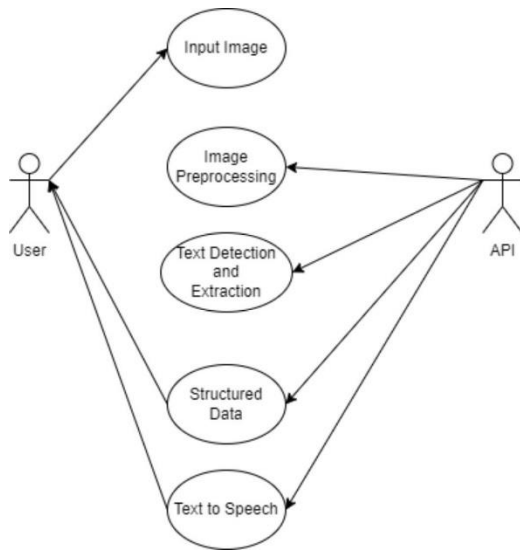


Fig 4.3. Use Case Diagram

Fig 4.3. Use case diagrams are dynamic in nature and there are some internal or external factors for making the interactions. These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. Hence to model the entire system, several use case diagrams are used. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. When a system is analyzed to gather functionalities, use cases are prepared and actors are identified. The use case diagrams are then modeled to present the outside view.

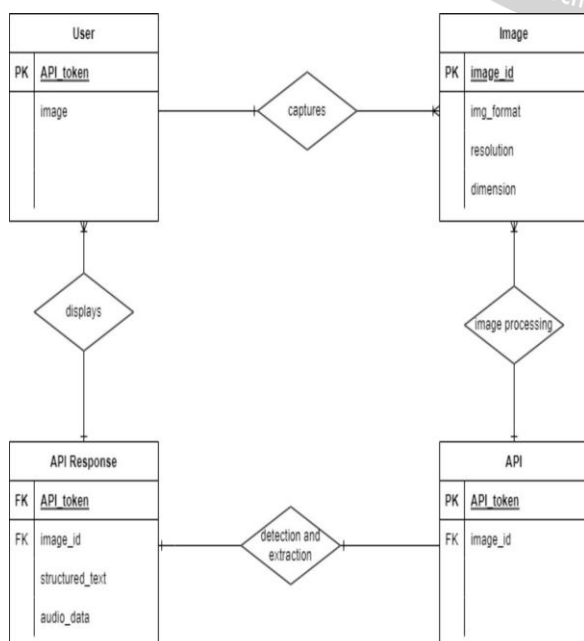


Fig 4.4 Entity-Relationship Diagram

Fig 4.4. An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. ER diagrams help to explain the logical structure of databases. They are created based on three basic concepts: entities, attributes, and relationships. It is used to represent the entity framework infrastructure.

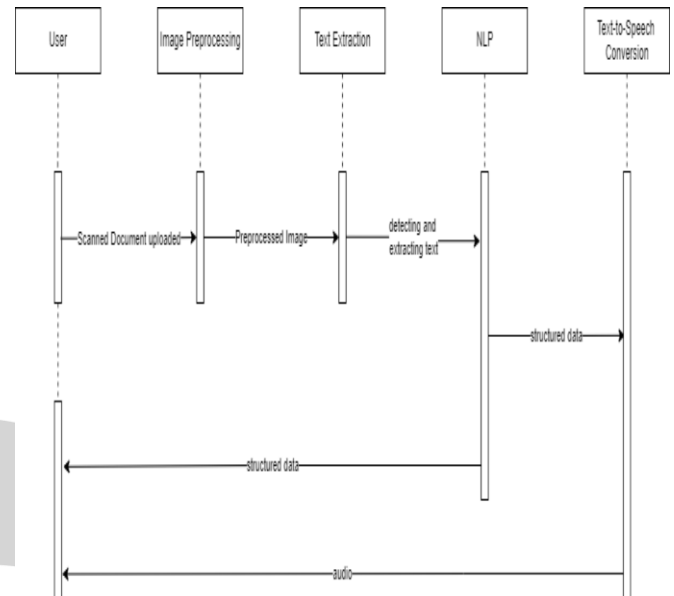


Fig 4.5 Sequence diagram

Fig 4.5. Sequence diagram depicts interaction between objects in a sequential order i.e., the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.

III. SCOPE OF THE PROJECT

The scope of our project on a grid infrastructure is to provide an efficient and enhanced software tool for the users to perform document image analysis, document processing by reading and recognizing the characters in research, academic, governmental, and business organizations that are having a large pool of documented, scanned images. Irrespective of the size of documents and the type of characters in documents, the product is recognizing them, searching them, and processing them faster according to the needs of the environment.

IV. RESULT ANALYSIS

We performed a comparative study of previous works and the limitations observed in each of these works. We then summarize our method and that the results that we have observed not only reached the objectives we have set but have also addressed a majority of the limitations seen in previous works in the field.

This paper, Text extraction using OCR: A Systematic Review [1], only focused on skew correction, all the images were of high quality and hence they didn't require to consider the use of pre-processing methods.

The methodology used in the paper, Improve OCR Accuracy with Advanced Image Preprocessing using Machine Learning with Python [2], does not cover images with uneven brightness, watermarks, or different fonts.

A reinforcement learning approach is used in “Improving the Accuracy of Tesseract 4.0 OCR Engine Using Convolution-Based Pre-processing” [16] and does not guarantee that local optimums will be avoided each time. The algorithm gets stuck on kernel configuration.

In the paper, Text Recognition from Images: A Study [3], their methodology is targeted at, and successfully performs noise removal by applying Gaussian filter and mean filter.

BLSTM-Bidirectional Long Short-Term Memory is used by the authors of the paper titled, Recognition of Printed Devanagari Text Using BLSTM Neural Network [4], which did not require segmentation. The also observed a very low accuracy.

For Information extraction, in the paper titled, System that automatically converts word into text [10], they concluded that an n-gram dictionary can be used so that words may be correctly identified that are not in the dictionary of geographical names.

A Study on Various Image Processing Techniques [6], in this paper the authors performed removal of noise using Laplacian and Harr Filtering only.

The algorithm used in “Robust Text Extraction in Images for Personal Event Planner” [12], is not tested for event information taken from handwritten images and complex fonts of printed text present in the images.

The authors of paper titled, Text recognition on images from social media [5], worked on a dataset collected from social media where the images were of high quality and of only printed text.

The Deep Learning Model for Text Recognition in Images [7], suggests the use of Convolutional Neural Network and Long Short-term Memory for text recognition.

In the paper, Design and implementation of android application to extract text from images by using tesseract for English and Hindi [13], the user edits or crops the image based on what text he wants; the process isn't automated.

The pre-processing techniques that were included in this paper title, Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition [15], are contrast stretching, noise removal techniques, normalization and segmentation. Though it was a detailed pipeline, the accuracy of the pre-processed image doesn't improve the text extraction accuracy.

The authors of the paper titled, Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models [14], proposed a new method in the form of pre-processing and recognition which are both based on ANNs.

In the paper, Steps Involved in Text Recognition and Recent Research in OCR; A Study [8], outlined only a text recognition flow and suggested classifiers such as ANN and SVM and didn't focus on pre-processing the images or how the classifiers can be implemented.

Our proposed solution has an image pre-processing pipeline that customises the methods it applies based on problems detected in the image such as gradient brightness, skewness, blurriness, noisy images etc. We pre-process the image by sharpening, brightening, deblurring, noise removal, de-skewing, and detecting documents from the background. We then detect and extract printed text from images efficiently and additionally convert the data into an audio file.

Images used in our dataset contained a count of words ranging from **800 to 1500 per document**. We observed an **average error rate of 11% at word level**.

The entire processing of the image to the conversion to speech takes about **3-5 seconds**.

V. FUTURE SCOPE

More features that can be added to this application are listed below. These will be developed in the near future, and the project will be kept alive with regular updates and modifications.

Training and implementing a model to recognize and extract handwritten characters. In a document containing both printed and hand written content, we want to be able to list the data in handwritten format in the same order it appears alongside and amidst the printed text.

The entire process starting from uploading the image to extracting downloadable data takes 3 seconds on an average, and up to 5 seconds for images with more content. We want to decrease this wait time for the user down to 1 second maximum.

VI. CONCLUSION

The objective of the project is explained, the task is introduced - including an explanation of the approach and the solution is illustrated. To achieve better performance, carefully chosen and placed elaborate image processing techniques are used, to increase the probability of retrieving desired text from the OCR engine. An NLP model converts raw data that may include misspelled words to structured data that can be downloaded as a copy. Acting as an additional support to the visually impaired, the system also converts the structured data to speech data in the form of an audio file that can be downloaded.

VII. REFERENCES

[1]. Mittal, Rishabh; Garg, Anchal, “Text extraction using OCR: A Systematic Review”, 2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA), 2020, pp 357–362.

- [2]. Sanjeev Kumar, Mahika Sharma, Kritika Handa, Rishika Jaiswal, "Dan", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-7, May 2020.
- [3]. Sahana K Adyanthaya, "Text Recognition from Images: A Study", International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Volume-8 Issue-13, 2020.
- [4]. Naveen Sankaran and C.V Jawahar, "Recognition of Printed Devanagari Text Using BLSTM Neural Network", 21st International Conference on Pattern Recognition (ICPR), November 11-15, 2012. Tsukuba, Japan.
- [5]. M.S. Akopyan, O.V. Belyaeva, T.P. Plechov and D.Y. Turdakov, "Text recognition on images from social media", Ivannikov Memorial Workshop (IVMEM), 2019.
- [6]. Dr. PL Chithra, P Bhavani, P., "A Study on Various Image Processing Techniques", International Journal of Emerging Technology and Innovative Engineering (ISSN (print): 2394 – 6598) Volume 5, Issue 5, May 2019.
- [7]. Anupriya Shrivastava, Amudha J., Deepa Gupta, Kshitij Sharma, "Deep Learning Model for Text Recognition in Images", 10th ICCCNT 2019 July 6-8, 2019, IIT - Kanpur, Kanpur, India.
- [8]. Sai Abhishikth Ayyadevara, P N V Sai Ram Teja, Bharath K P Rajesh Kumar M, "Handwritten Character Recognition Using Unique Feature Extraction Technique", International Research Journal of Modernization in Engineering Technology and Science, Volume-3 Issue-10, Jan 2020.
- [9]. Karen Kukich, "System that automatically converts word into text", ACM Computing Surveys, Vol. 24, No. 4, December 1992.
- [10]. Yang Zhang, Hao Zhang, HaoranLi, "Event Info Extraction from Flyers", 2021
- [11]. Bhaskar, L., & Ranjith, R., "Robust Text Extraction in Images for Personal Event Planner", 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) July 1-3, 2020, IIT-Kharagpur, Kharagpur, India.
- [12]. Brijeshkumar Y. Panchal and Gaurang Chauhan, "Design and implementation of android application to extract text from images by using tesseract for English and Hindi", 3rd International Scientific Conference of Engineering Sciences and Advances Technologies (ICESAT), Journal of Physics: Conference Series, Volume 1973, 4-5 June 2021.
- [13]. Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4, April 2011.
- [14]. K. Gaurav and Bhatia P. K., "Analytical Review of Preprocessing Techniques for Offline Handwritten Character Recognition", 2nd International Conference on Emerging Trends in Engineering & Management, ICETEM, 2013.