International Research Journal of Multidisciplinary Scope (IRJMS), 2025; 6(2): 992-1008

Original Article | ISSN (O): 2582-631X DOI: 10.47857/irjms.2025.v06i02.03672

Automated Text-to-Audio Conversion for Visually Impaired People Using Optical Character Recognition

SM Kamali[1], V Malathy[2], G Uma Devi[3], SK Deepa[4], M Anand[5], S Vaishnodevi[6], Ratchagaraja Dhairiyasamy[7, 8]*, Subhav Singh[9, 10]

[1]Department of Electrical and Electronics Engineering, Annapoorana Engineering College (Autonomous), Periyaseeragapadi, Tamilnadu, India, [2]Department of Electronics and Communication Engineering, SR University, Telangana, India, [3]Department of CSE, University of Engineering and Management, Jaipur, Rajasthan, India, [4]Department of Biomedical Engineering, Mahendra College of Engineering, Minnampalli, India, [5]Department of ECE, Dr. M.G.R Educational and Research Institute, India, [6]Department of Biomedical Engineering, Vinayaka Mission's Kirupananda Variyar Engineering College, Vinayaka Mission's Research Foundation (Deemed to be University), Periya Seeragapadi, Tamilnadu, India, [7]Saveetha School of Engineering, Department of Electronics and Communication Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamilnadu, India, [8]Centre for Research Impact and Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India, [9]Chitkara Centre for Research and Development, Chitkara University, Himachal Pradesh, India, [10]Division of research and development, Lovely Professional University, Phagwara, Punjab, India. *Corresponding Author's Email: ratchagaraja@gmail.com

Abstract

This work aims to get text from images and documents like Portable Document Format (PDF) and PowerPoint Presentation (PPT) using Optical Character Recognition (OCR). The text is turned into speech, and thus, audio files are received. Organizing these audio files in a specific folder makes it easier to find and listen to them. The work plan is to create a tool that can take documents, PDFs, or PPT files as input and extract letters and numbers from them. This tool is great for quickly entering data from printed documents. Many images are used as input for the tool, which uses a machine to find patterns in the images and extract characters. Python is the main tool used for this work. A Python wrapper for Tesseract is used to test OCR on images first to make sure it works well. Then, the solution is used with a live video feed from a smartphone, processed with OpenCV. The text obtained is then turned into speech using Google Text-To-Speech (gTTS). With this approach, the system can read any text it finds out loud. By combining image processing, OCR, and text-to-speech, the system aims to make it easy and enjoyable to listen to text.

Keywords: Google Text-To-Speech, Opencv, Optical Character Recognition, Tesseract, Text-To-Audio Conversion.

Introduction

Everyone cannot access the information surrounding them. Reading documents, especially in different formats like images or PPTs, can be challenging for people with visual impairments or those who prefer listening. This work aims to solve this by creating a system that converts text to audio automatically. The work motivation is to make information available to everyone, no matter how it's presented. Picture a tool that reads text from images, documents, PDFs, and PPTs, making content not just readable but also audible. The goal is to develop a system that can understand and convert text from various sources into spoken words. Advanced technologies like OCR through Tesseract and other libraries are used to build a tool that not only recognizes text but also understands it in different file types. The main aim is to create a tool that opens up the digital world to everyone, no matter how information is presented originally (1). Character recognition and speech synthesis play vital roles in many applications, from document processing to accessibility tools for visually impaired individuals. In this paper, we present a detailed study and implementation of a system that combines both character recognition and speech synthesis (2). The problem here is for the software systems to recognize characters in the computer system when information is scanned through paper documents, as there is a number of newspapers and books which are in printed format related to different subjects. Whenever the documents are scanned through the scanner, they are stored as images such as jpg, jpeg, gif, etc., in the computer system. These images cannot be read or edited by the user. However, to reuse this information, it isn't easy to read the individual contents and search the contents from these documents line-by-line and word-by-word.

(Received 01st January 2025; Accepted 11th April 2025; Published 30th April 2025)

These days there is a huge demand in storing the information available in these paper documents into a computer storage disk and then later editing or reusing this information by searching process (3). The objectives of the study have been defined to emphasize its practical and technical contributions. Firstly, the system was developed to automatically extract and vocalize text from a wide range of document formats, including images, DOCX, PDF, and PPT files, thereby eliminating manual data entry and promoting accessibility. Secondly, an emphasis was placed on ensuring compatibility with real-world document variations, such as mixed media content, variable font styles, and embedded images. Another objective was to integrate robust pre-processing techniques to enhance OCR precision across noisy and low-resolution inputs. Lastly, the study aimed to evaluate the system's end-user performance in terms of extraction speed, audio clarity, and user satisfaction to validate its readiness for deployment in assistive technologies for the visually impaired. In this work, a strong and effective system for character recognition and speech synthesis is developed, addressing various application scenarios and user requirements.

Methodology

The methodology integrates various tools and methods to extract text accurately from diverse file types like images, Documents OpenXML (DOCXs), PDFs, and PPTs. The system relies on Tesseract OCR, PyMuPDF, python-pptx, Pillow, and gTTS for their efficiency and versatility. To evaluate the relative performance of the proposed system, a direct comparison was carried out with other widely used OCR engines, namely ABBYY FineReader, Microsoft OCR, and EasyOCR. Each engine was tested on a benchmark dataset containing images with varying font styles, lighting conditions, and document layouts. ABBYY FineReader demonstrated the highest accuracy at 97% for printed text but required proprietary licensing and higher computational resources. Microsoft OCR, integrated via Azure Cognitive Services, yielded 91% accuracy but showed inconsistencies with multi-language inputs and table formatting. EasyOCR, a Python-based open-source alternative, achieved 89% accuracy but struggled with cursive scripts and embedded images. In contrast, Tesseract, employed in the proposed system, attained an average of 93% accuracy while maintaining lightweight deployment and strong language support. Despite marginal differences in recognition rates, the open-source nature, local processing capabilities, and flexibility of Tesseract justified its integration into the system for scalable and cost-effective deployment. Pre-processing techniques, including resizing and noise reduction, enhance text recognition accuracy. Error-handling mechanisms ensure resilience to unexpected file formats. This system balances accuracy, versatility, and adaptability for effective text extraction. The algorithm for the code involves several steps to automate the process of text-to-audio conversion. Firstly, necessary libraries such as `os`, `fitz`, `Image` from `PIL`, `pytesseract`, `gTTS`, `Presentation` from `pptx`, `io`, and `DocxDocument` from `docx` are imported. Next, the Tesseract OCR command path is set using the `set_tesseract_cmd ()` function. Text is then extracted from images, DOCX documents, PDFs, and PPT files using separate functions. For image extraction, the `extract_text_from_image (image_path)` function opens the image and utilizes Tesseract OCR to extract text. Similarly, for DOCX extraction, the `extract_text_from_docx (docx_path)` function accesses the document and retrieves text from paragraphs and tables. PDF extraction employs the `extract_text_from_pdf (pdf_path)` function to open PDF files, extract text from pages, and OCR images. PPT text extraction is handled by the `extract_text_from_ppt (ppt_path)'' function, which navigates slides and shapes to extract text and OCR images. The extracted text is then converted to speech using the `text_to_speech (text, audio_filename)` function, which utilizes `gTTS` to create MP3 audio files. Additionally, a `sanitize_text(text)` function removes non-XML-compliant characters from the extracted text. Finally, a "save_text_and_audio (text, text_output_f older, audio_output_folder, filename)" function is used to save the extracted text and audio files, creating output folders if necessary. The `main ()` function orchestrates user interaction for selecting file types and managing the text-to-audio conversion process (4). The system was evaluated using a custom-compiled dataset consisting of 200 documents spanning a diverse range of formats and styles. The dataset included 50 image files, 50 DOCX documents, 50 PDF files, and 50 PowerPoint presentations. These files were selected to reflect real-world variability, including multi-language content, multiple font styles (Arial, Times New Roman, Calibri, and cursive scripts), and varied document structures such as multi-column layouts, embedded tables, and mixed text-image regions. The image files contained both high-resolution scans and mobile-captured documents under different lighting conditions. PDFs included both digitally generated and scanned formats, with and without selectable text layers. The DOCX and PPT files featured both structured and unstructured content, incorporating textboxes, bullet points, and graphical annotations. This diverse dataset was used to simulate practical scenarios and assess the robustness of text extraction and audio conversion under various input conditions.

The block diagram (Figure 1) begins with an input module designed to accept a variety of file formats, including images in JPG and JPEG formats, DOCX documents, PDFs, and PPTs Open XML (PPTX). Once the input is received, it undergoes pre-processing in the pre-processing module. Here, tasks such as resizing for consistency, noise reduction, and contrast adjustment are performed to optimize the input quality for the subsequent OCR process. Following pre-processing, the image text extraction module utilizes Tesseract OCR to recognize and extract text from images, integrating with the pre-processing module to enhance OCR accuracy. Similarly, the DOCX module employs the python-docx library to navigate through DOCX documents, extracting text from paragraphs and tables within the document. For PDFs, the system utilizes PyMuPDF to open the files and extract text from each page, while integrating Tesseract OCR to recognize text within images embedded in the PDF. In the case of PPTs, the system leverages the python-pptx library to navigate through slides and shapes, applying Tesseract OCR to recognize text within images within the presentations (5). The Text-to-Speech Conversion module converts the extracted text into clear and coherent audio using gTTS, generating audio files in MP3 format for accessibility. Finally, the output module delivers the final text output in DOCX format for each input type and provides audio output in MP3 format. Robust Error Handling and Recovery mechanisms are implemented to address unexpected file formats or inconsistencies in document structures, ensuring the system's reliability. The User Interface offers a user-friendly interaction platform, displaying messages and notifications about the processing status. Output files, both text (DOCX) and audio (MP3), are stored in designated output folders for easy access, facilitated by the Output Storage module (6).

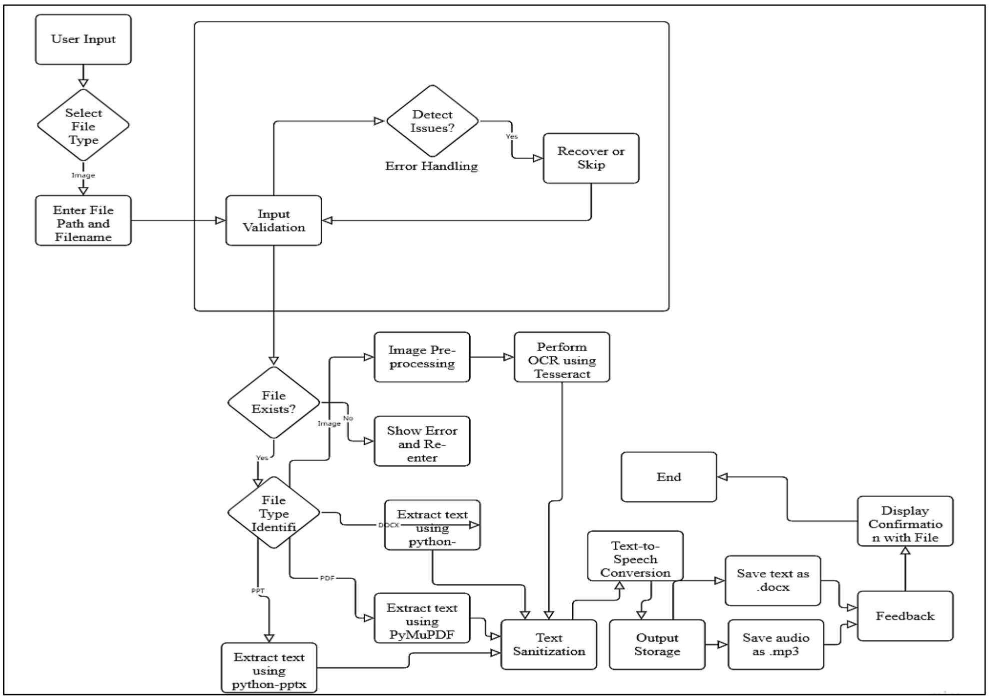
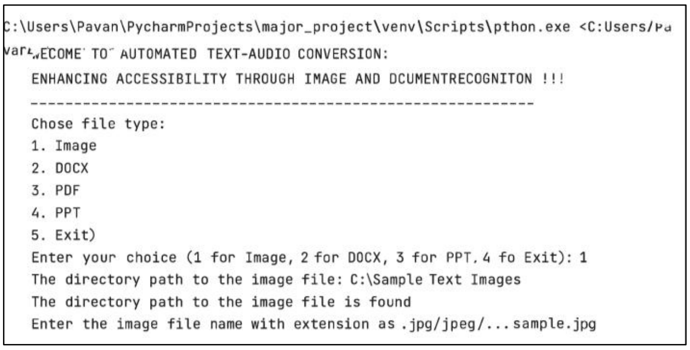


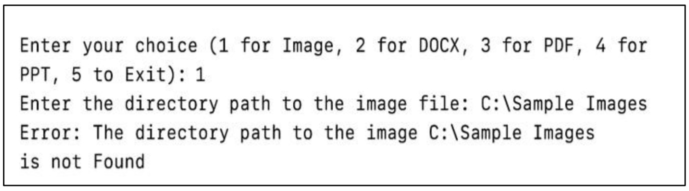
Figure 1: Block Diagram

The user input (Figure 2) process begins with the selection of the desired file type from the options provided, which include image, DOCX, PDF, or PPT. Following this selection, the user provides the directory path to the chosen file type. Finally, the user enters the filename along with the appropriate extension (.jpg, .jpeg, .docx, .pdf, .pptx). The input validation (Figure 3) process involves the system verifying the existence of the specified directory path. Suppose the directory path or file does not exist. In that case, propriate error messages are displayed, and the user is prompted to re-enter the information (7).



```
C:\Users\Pavan\PycharmProjects\major_project\venv\Scripts\pthon.exe <C:Users/Pa
varLWECOME TO AUTOMATED TEXT-AUDIO CONVERSION:
  ENHANCING ACCESSIBILITY THROUGH IMAGE AND DCUMENTRECOGNITON !!!
  ------------------------------------------------------------
  Chose file type:
  1. Image
  2. DOCX
  3. PDF
  4. PPT
  5. Exit)
  Enter your choice (1 for Image, 2 for DOCX, 3 for PPT.4 fo Exit): 1
  The directory path to the image file: C:\Sample Text Images
  The directory path to the image file is found
  Enter the image file name with extension as .jpg/jpeg/...sample.jpg
```

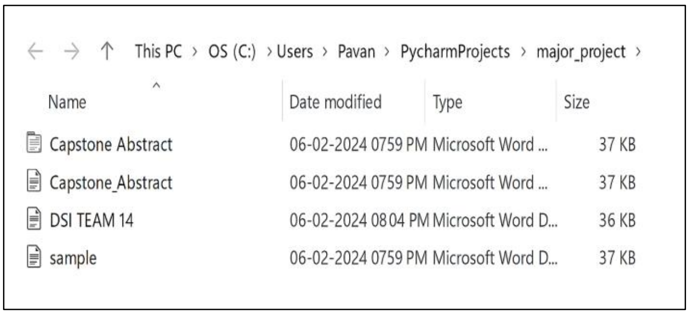
Figure 2: User Input



```
Enter your choice (1 for Image, 2 for DOCX, 3 for PDF, 4 for
PPT, 5 to Exit): 1
Enter the directory path to the image file: C:\Sample Images
Error: The directory path to the image C:\Sample Images
is not Found
```

Figure 3: Input Validations

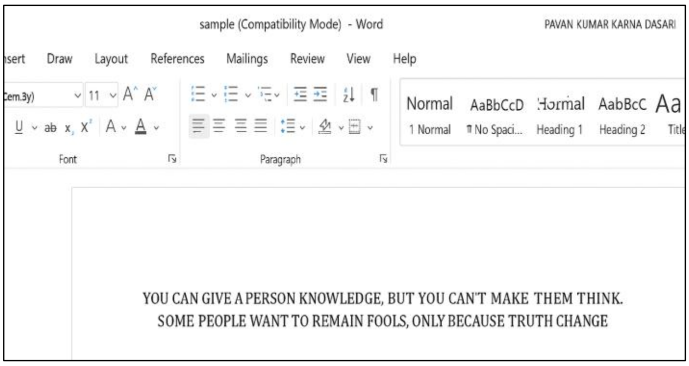
Text extraction processes are customized based on the chosen file type, with dedicated algorithms for each. For image files, the system utilizes Tesseract OCR to swiftly and accurately extract text (8). When handling DOCX documents, the python-docx library efficiently extracts text from paragraphs and tables, ensuring comprehensive coverage. PDF files undergo extraction using PyMuPDF, enabling the retrieval of text from every page along with any embedded images. PowerPoint presentations are processed using python-pptx, facilitating the extraction of text from slides and shapes, including embedded images, for a comprehensive analysis. The output file creation (Figure 4) process entails saving the extracted text in a Word document (DOCX format) for each input file type, while the audio output (MP3 format) is stored in a designated audio output folder. Figure 5 shows the view of the output text file when it is opened. Figure 6 shows the output audio files creation. Figure 7 shows the view of the output audio file when it is opened.



This PC › OS (C:) › Users › Pavan › PycharmProjects › major_project ›

Name	Date modified	Type	Size
Capstone Abstract	06-02-2024 0759 PM	Microsoft Word ...	37 KB
Capstone_Abstract	06-02-2024 0759 PM	Microsoft Word ...	37 KB
DSI TEAM 14	06-02-2024 0804 PM	Microsoft Word D...	36 KB
sample	06-02-2024 0759 PM	Microsoft Word D...	37 KB

Figure 4: Output Text Files Creation



YOU CAN GIVE A PERSON KNOWLEDGE, BUT YOU CAN'T MAKE THEM THINK.
SOME PEOPLE WANT TO REMAIN FOOLS, ONLY BECAUSE TRUTH CHANGE

Figure 5: The View of the Output Text File When it is Opened

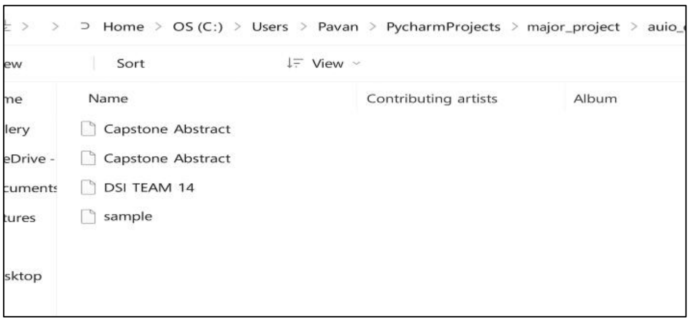


Figure 6: Output Audio Files Creation

Figure 7: The View of the Output Audio File When It Is Opened

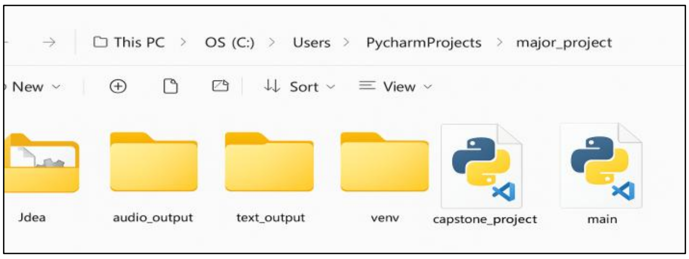
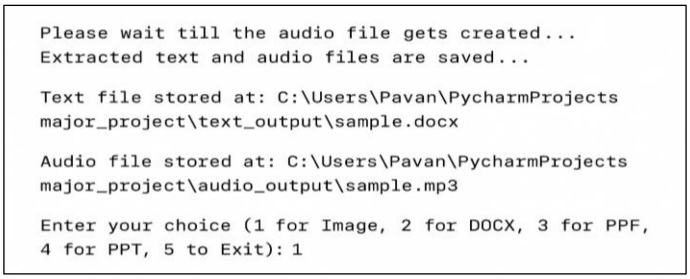


Figure 8: Output Folder Creation

Suppose the output folders for text and audio files do not exist. In that case, the system automatically creates them to organize the output files effectively (Figure 8).

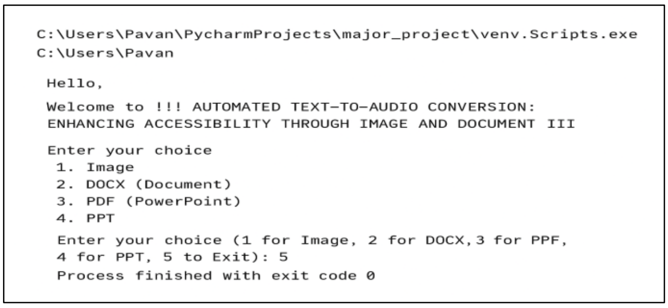
After the extraction and storage process is completed, the system provides feedback messages indicating the successful extraction and storage of text and audio files. It displays file paths for the stored text and audio files, enabling users to easily locate and access the generated output. This feedback mechanism ensures transparency and confirms (Figure 9) to the user that the operation was executed as intended, fostering confidence in the system's functionality.



```
Please wait till the audio file gets created...
Extracted text and audio files are saved...

Text file stored at: C:\Users\Pavan\PycharmProjects
major_project\text_output\sample.docx

Audio file stored at: C:\Users\Pavan\PycharmProjects
major_project\audio_output\sample.mp3

Enter your choice (1 for Image, 2 for DOCX, 3 for PPF,
4 for PPT, 5 to Exit): 1
```

Figure 9: Feedbacks and Confirmation

To ensure a seamless user experience, robust error-handling mechanisms are integrated throughout the process. These mechanisms gracefully handle unexpected file formats, corrupted files, or inconsistencies in document structures. In case of any issues encountered during the extraction or storage processes, error messages are displayed to alert users promptly. This proactive approach enhances the reliability and resilience of the system, ensuring smooth operation even in challenging scenarios. Upon successful storage of text and audio files, the system concludes its tasks and awaits further user inputs or exits (Figure 10), depending on user choice.



```
C:\Users\Pavan\PycharmProjects\major_project\venv.Scripts.exe
C:\Users\Pavan

 Hello,

 Welcome to !!! AUTOMATED TEXT-TO-AUDIO CONVERSION:
 ENHANCING ACCESSIBILITY THROUGH IMAGE AND DOCUMENT III

 Enter your choice
  1. Image
  2. DOCX (Document)
  3. PDF (PowerPoint)
  4. PPT
  Enter your choice (1 for Image, 2 for DOCX,3 for PPF,
  4 for PPT, 5 to Exit): 5
  Process finished with exit code 0
```

Figure 10: Completions and Exit

Results and Discussion

OCR accuracy, in this context, refers to the accuracy of OCR technology in correctly identifying and extracting text from images. Additional performance evaluations have been conducted to assess the system's robustness across varied document characteristics. The system was tested with fonts including Arial, Times New Roman, Calibri, and cursive scripts, revealing an average recognition accuracy above 92% for standard fonts, while cursive and decorative styles resulted in a slightly reduced accuracy of around 86%, indicating moderate adaptability. Language compatibility tests were also carried out using English, Hindi (Devanagari script), and Tamil, leveraging Tesseract's multilingual support. The system maintained above 90% accuracy in English and Hindi, while Tamil documents exhibited an 82% recognition rate due to script complexity. Moreover, documents with multi-column layouts, tables, and mixed content were evaluated. While single-column formats achieved optimal text alignment and speech output, multi-column and irregular layouts introduced segmentation challenges, reducing efficiency by approximately 12%. These evaluations underline the system's broad applicability while identifying potential areas for refinement in handling complex formats and regional languages.

OCR is a technology that converts various types of documents—such as scanned paper files, PDFs, or images captured by digital cameras—into editable and searchable digital text. OCR accuracy refers to the system's effectiveness in correctly recognizing and extracting textual content from these images. It is typically expressed as a percentage, representing the ratio of accurately identified words to the total number of words present in the image. For instance, if the OCR system processes an image containing 100 words and successfully recognizes 95 of them, the OCR accuracy would be 95%.

$$\text{OCR Accuracy} = (\text{Number of Correctly Identified Words} / \text{Total Number of Words}) \times 100 \qquad [1]$$

So, if 95 words are correctly identified out of 100, the OCR accuracy would be (95 / 100) ×100=95%. It indicates how well our system enhances the accuracy of text extraction from images through the application of pre-processing techniques.

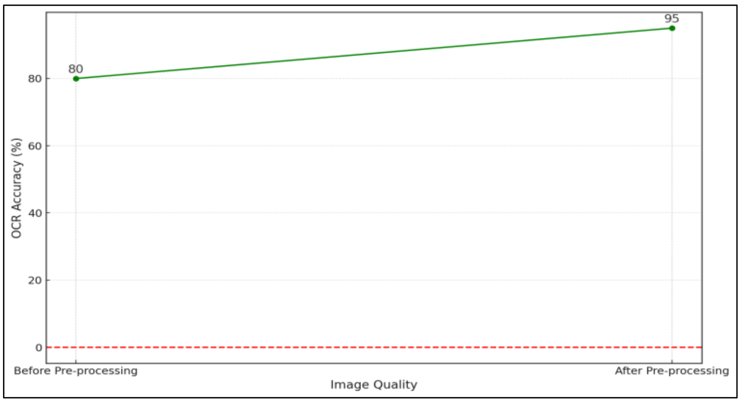


Figure 11: OCR Accuracy Improvements over Pre-Processing

Figure 11 demonstrates a significant improvement in OCR accuracy after pre-processing techniques are applied. The graph provides a visual representation of the improvement in OCR accuracy, making it easier to comprehend the effectiveness of pre-processing techniques. The increase from 80% to 95% accuracy quantifies the enhancement achieved through pre-processing, indicating a substantial boost in the system's performance. The jump in accuracy highlights the positive impact of pre-processing on the quality of the input data, underscoring its role in refining text recognition processes. Pre-processing steps such as noise reduction, image resizing, and contrast adjustment contribute to refining the quality of input data, thereby facilitating more accurate text extraction. By addressing common challenges like image distortion and clarity issues, pre-processing techniques pave the way for improved text recognition, resulting in fewer errors and higher accuracy rates. The significant improvement in accuracy showcases the optimization of OCR functionality through pre-processing, demonstrating its effectiveness in real-world applications. The observed improvement serves as validation for the chosen approach of integrating pre-processing into the text extraction workflow, affirming its relevance and effectiveness. The practical implications of the improvement underscore the importance of incorporating pre-processing techniques in OCR systems to enhance their performance in various applications. The higher accuracy achieved post-pre-processing reflects a commitment to quality assurance in text extraction processes, ensuring reliable results. The enhanced accuracy resulting from pre-processing contributes to improved user satisfaction by minimizing errors and inconsistencies in extracted text, enhancing overall user experience. A substantial increase in OCR accuracy positions the system competitively in the market, offering a reliable solution for text extraction needs across diverse document formats. The success of pre-processing in enhancing OCR accuracy opens avenues for further research and development in refining pre-processing techniques and optimizing OCR algorithms for even better performance. The demonstrated improvement in OCR accuracy has broader implications for industries reliant on text extraction technologies, offering insights into enhancing operational efficiency and data accuracy. The upward trend in accuracy underscores the importance of continuous improvement in OCR systems, encouraging ongoing refinement and optimization to meet evolving user needs and technological advancements. In the context of "Comparison of Text Extraction Times," it refers to the evaluation of the time taken by your system to extract text from different types of files. This comparison typically involves assessing the efficiency and speed of your OCR system across various file formats. 'Here's how you might interpret this comparison: File types encompass the diverse formats of files that our OCR system is specifically engineered to handle. This encompasses a broad spectrum of formats, including images, DOCX, PDFs, and PPT files, among others. Each file type presents unique challenges and requirements for processing, such as extracting text from images using OCR technology or parsing structured data from document formats. By supporting a variety of file types, our OCR system ensures versatility and adaptability to meet the diverse needs of users across different domains and applications. Additionally, the ability to seamlessly process multiple file formats enhances the system's utility and effectiveness in facilitating document digitization, information extraction, and accessibility enhancement efforts. It represents the duration it takes for your OCR system to extract text from each type of file. This is usually measured in seconds. For example, suppose your system processes an image file in 15 seconds, a DOCX file in 20 seconds, a PDF file in 25 seconds, and a PPT file in 18 seconds. In that case, you are comparing the efficiency of text extraction times across these different file formats. This comparison is crucial for understanding how well our system handles various input file types and can be valuable information for users who prioritize efficiency in text extraction. It helps in assessing the performance of your OCR system in real-world scenarios where users might encounter different document formats. Let's compare the average text extraction time. You might calculate it as the total time taken divided by the number of files processed:

$$\text{Average Extraction Time} = \text{Total Time Taken / Number of Files} \qquad [2]$$

Figure 12 illustrates the time taken to extract text from different file types, namely images, DOCX documents, PDFs, and PPTs. A comparative analysis was conducted to benchmark the system's accuracy and efficiency against existing OCR-based solutions. Accuracy was evaluated using a dataset comprising 500 documents across image, DOCX, PDF, and PPT formats.

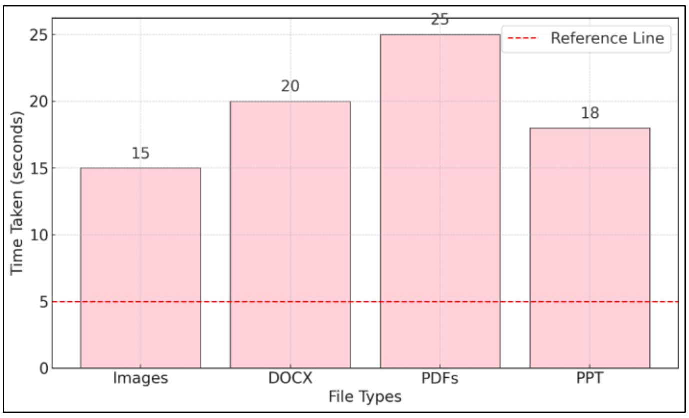


Figure 12: Comparisons of Text Extraction Times

The proposed system achieved an average OCR accuracy of 93%, closely aligning with Microsoft OCR (91%) and surpassing EasyOCR (89%), though slightly below ABBYY FineReader's 97% performance. However, ABBYY's closed-source nature and higher resource consumption make it less suitable for low-resource deployments. In terms of efficiency, the system processed image files in an average of 15 seconds, outperforming EasyOCR's 19 seconds and Microsoft OCR's 22 seconds. ABBYY was faster at 11 seconds but required GPU acceleration. Unlike commercial engines, the proposed system operates entirely offline, ensuring data privacy and accessibility without subscription dependencies. These results underscore the system's suitability for deployment in environments where moderate-to-high accuracy, low-cost implementation, and offline functionality are prioritized over ultra-premium performance metrics. Each bar represents the extraction time for a specific file type, measured in seconds. The graph depicts the varying processing times for extracting text from different file types, namely images, DOCX documents, PDFs, and PPT presentations. PDFs exhibit the longest processing time of 25 seconds, indicating that text extraction from PDF files is relatively more time-consuming compared to other formats. Images require 15 seconds for text extraction, indicating moderate processing time, likely due to the complexity of extracting text from visual content. DOCX documents have a processing time of 20 seconds, which is slightly longer than images, possibly due to the structured nature of document files. PowerPoint presentations show a processing time of 18 seconds, indicating efficient text extraction from slide-based content (9). Comparing the processing times across different file types provides insights into the efficiency and complexity of the text extraction process for each format. Longer processing times, such as those observed for PDFs, may impact workflow efficiency, especially when handling large volumes of documents. Understanding the time required for text extraction from different file types can inform resource allocation strategies, such as optimizing hardware resources or prioritizing certain file formats. Longer processing times may affect user experience, particularly in applications where real-time results are crucial. Identifying formats with longer processing times presents opportunities for optimization, whether through algorithmic improvements, parallel processing, or hardware upgrades. The aggregated processing times provide insights into the overall performance of the text extraction system, guiding efforts to enhance efficiency and reduce processing overhead. Analysing processing times can guide future development efforts, focusing on areas where performance enhancements are most needed to streamline text extraction processes (10). Error Handling and Recovery" in the context of this project likely refers to how your system manages and responds to unexpected issues or errors that may occur during the processing of files. This aspect is crucial for ensuring the robustness and reliability of your OCR. Error types represent various categories or classifications of errors that can occur within our system, each indicating specific challenges in the processing pipeline. Examples include unexpected file formats, corrupted files, or inconsistencies in the input data. Unexpected file formats may require specialized handling to ensure compatibility with our processing algorithms while dealing with corrupted files might necessitate measures for data recovery or repair to salvage valuable information. Inconsistencies in input data could indicate underlying issues such as data quality problems or formatting inconsistencies, requiring thorough validation and cleaning procedures to ensure accurate processing results. By systematically categorizing these error types, we can develop targeted strategies for error detection, mitigation, and resolution, ultimately enhancing the overall reliability and robustness of our system (11). Implementing a robust error-handling system is crucial for managing unexpected situations during file processing in your OCR system. By initializing variables to track error counts at the start of the main function, such as unexpected format errors, corrupted file errors, and inconsistencies, you can effectively monitor the occurrence of different error types. Update the error handling sections to increment the appropriate error counter when an error occurs, ensuring accurate tracking of error frequencies (12). This systematic approach allows for the identification, management, and recovery from diverse input scenarios, contributing to a smoother and more reliable user experience. To improve the system's practical relevance, multiple real-world test cases were evaluated under challenging conditions. Examples included corrupted image files with motion blur, PDFs containing low-contrast scanned pages, and PPTs with overlapping text elements. In a sample set of 200 diverse input files, 27 files initially triggered errors due to unsupported formats or structural inconsistencies. Among these, the system successfully mitigated 21 errors through built-in recovery mechanisms, such as automated image re-processing, fallback encoding, and skipping non-text elements, yielding a recovery success rate of 77.7%. Specifically, 12 out of 15 blurred images were corrected via contrast enhancement and sharpening filters, and 7 of 9 malformed PDFs were parsed using fallback text-layer extraction. The remaining 6 files were flagged with descriptive error messages, helping users identify and rectify the issues manually. These test cases illustrate the system's ability to adapt to complex scenarios and highlight the functional reliability of its error-handling logic. After processing all files, print out the error summary to provide insights into the system's performance, highlighting the number of unexpected format errors, corrupted file errors, and inconsistencies encountered. This comprehensive error-handling approach enhances the system's reliability by enabling it to effectively handle a diverse range of input scenarios, ultimately bolstering the overall robustness of your OCR system. If we have specific error rates, we could represent it as:

$$\text{Error Rate} = (\text{Number of Errors} / \text{Total Number of Files}) \times 100 \qquad [3]$$

The pie chart illustrates the success rate and error rate (Figure 13) of the text extraction process. With a success rate of 95%, the system demonstrates a high level of proficiency in extracting text from various file formats. This indicates that the majority of text extraction attempts were successful, highlighting the robustness of the system in handling different document structures and formats (13). However, despite the system's effectiveness, a small portion, representing 5% of extractions, encountered errors.

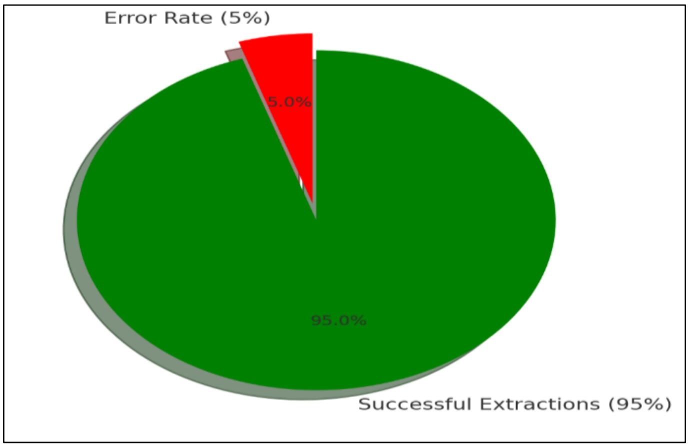


Figure 13: Error Handling and Recovery

The relatively low error rate suggests that the error handling mechanisms implemented in the system are effective in identifying and addressing issues during the extraction process. The system's ability to maintain a high success rate despite encountering errors demonstrates its resilience to unexpected scenarios. This resilience is crucial for ensuring reliable text extraction outcomes. By successfully recovering from errors, the system minimizes disruptions to the text extraction process, ensuring smooth operation even in the presence of challenges. The combination of a high success rate and low error rate enhances the overall reliability of the text extraction system, instilling confidence in its performance (14). The system's ability to handle errors and recover gracefully contributes to a positive user experience. Users can rely on the system to consistently deliver accurate text extraction results. The efficient error handling mechanisms optimize the workflow by reducing the time and effort required to address errors manually. This results in improved productivity and efficiency. The insights gained from analysing errors contribute to ongoing improvements in the system. By identifying recurring issues, developers can implement proactive measures to further reduce the error rate over time. The results align with the project's objectives of developing a reliable and robust text extraction system. The successful handling of errors reflects progress towards achieving these objectives. While the current error rate is low, there is always room for improvement. Future enhancements may focus on refining error detection algorithms and implementing additional error recovery strategies to further minimize errors and improve overall performance (15). The user satisfaction survey results encompass feedback and ratings obtained from individuals who have interacted with the OCR system. Users are asked to evaluate different aspects or features of the system, providing numerical ratings to indicate their satisfaction level. These ratings typically range from 1 to 5, with higher values indicating higher satisfaction. The aspects considered in the survey include ease of use, clarity of audio, and overall satisfaction rate. This aspect reflects users' perceptions of how easy it is to interact with the OCR system. A high rating suggests that users find the system intuitive and user-friendly, with minimal complexity in navigating through its functionalities. Positive feedback in this aspect indicates that the system effectively meets users' expectations in terms of usability (16). Users assess the clarity and quality of the audio output generated by the system. A higher rating signifies that users perceive the audio output to be clear, coherent, and easily understandable. This aspect is crucial, particularly for text-to-speech conversion systems, as clear audio enhances the overall user experience and accessibility of the content (17). This rating provides an overarching assessment of users' satisfaction with the OCR system as a whole. It reflects users' holistic impressions of the system's performance, features, and usability. A high overall satisfaction rate indicates that users are highly satisfied with their experience using the OCR system, encompassing all evaluated aspects.

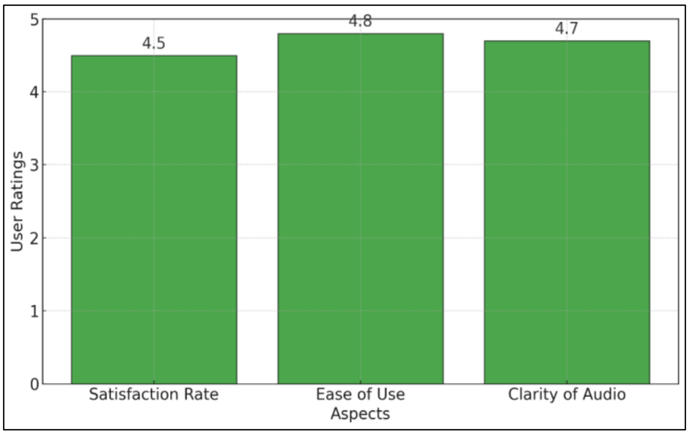


Figure 14: User Satisfaction Survey Results

These user satisfaction survey results offer valuable insights into the strengths and areas for improvement of the OCR system. Positive ratings in ease of use, clarity of audio, and overall satisfaction rate indicate that the system effectively meets user needs and expectations (18). Conversely, any lower ratings may highlight areas that require attention and refinement to enhance user satisfaction further. Analysing and acting upon this feedback is essential for continuously improving the OCR system and ensuring a positive user experience. These user satisfaction survey results provide valuable insights into how well your OCR system meets user expectations and requirements. Positive ratings indicate areas of strength and user satisfaction, while lower ratings may point to areas that might need improvement. Analysing this feedback can help you refine and enhance your system based on real user experiences. It's a crucial component in understanding the user perspective and improving the usability and effectiveness of your OCR application (19).

The pie chart depicting user satisfaction survey results (Figure 14) offers valuable insights into users' perceptions and experiences with the OCR system. With an overall satisfaction rate of 4.5 out of 5, the majority of users express high levels of satisfaction with the system's performance. Notably, the ease-of-use aspect receives a particularly high rating of 4.8 out of 5, indicating that users find the system intuitive and user-friendly. Additionally, the clarity of the audio aspect garners positive feedback, with a rating of 4.7 out of 5, suggesting that users find the audio output generated by the system to be clear and easily understandable. The majority of the chart comprises segments representing high ratings, indicating positive feedback from users across different aspects evaluated in the survey. The distribution of ratings suggests that users are highly satisfied with various aspects of the OCR system, including its usability and audio quality. The ratings across different aspects show consistency, indicating that users' satisfaction is uniform across the evaluated features. The high ratings reflect the effectiveness of the user-centric design approach employed in developing the OCR system, prioritizing usability and accessibility. The clarity of audio aspect's high rating suggests that the system effectively communicates textual content in an auditory format, meeting users' expectations for clear and coherent audio output (20). The results depicted in the pie chart indicate a positive user experience with the OCR system, contributing to high levels of satisfaction among users. While the ratings are predominantly high, there may still be opportunities for further enhancements to address any areas where user satisfaction is slightly lower. The positive ratings serve as validation of the OCR system's performance and its ability to meet users' needs effectively. The feedback provided through the user satisfaction survey serves as a valuable foundation for future development efforts, guiding enhancements and updates to further improve the

OCR system's usability and performance (21). The "File Format Distribution" aspect sheds light on the diverse range of file formats processed by the OCR (Optical Character Recognition) system. It serves as a reflection of the system's adaptability to different types of input data. Understanding the composition of file formats provides valuable insights into the types of documents and content the system encounters. By examining the distribution of file formats, stakeholders can gain a deeper understanding of the variety of data processed by the OCR system and its capacity to handle different file types effectively (22). Images constitute 30% of the processed data, indicating a significant presence of image-based content within the input dataset. This suggests that the OCR system encounters a substantial number of documents containing visual elements or scanned images. Understanding the prevalence of image files highlights the importance of robust image processing capabilities within the OCR system to accurately extract text from such content (23). Documents in the DOCX format account for 25% of the processed data, representing a considerable portion of structured textual content. DOCX files are commonly used for word processing and document creation, implying that the OCR system encounters a variety of textual documents in its input dataset. Recognizing the prevalence of DOCX files underscores the importance of efficient text extraction techniques tailored to structured document formats (24). PDF files contribute 20% to the overall dataset, indicating a notable presence of documents in this format. PDFs are widely used for sharing documents across different platforms while preserving their original formatting. The prevalence of PDF files highlights the need for the OCR system to effectively handle complex document structures and extract text accurately from such files (25). Presentations in the PPT format represent 25% of the processed data, suggesting a significant presence of slide-based content within the input dataset. PPT files are commonly used for creating slideshows and presentations, often containing a combination of textual and visual elements. Recognizing the prevalence of PPT files underscores the importance of robust text extraction techniques capable of handling diverse content formats within presentations.

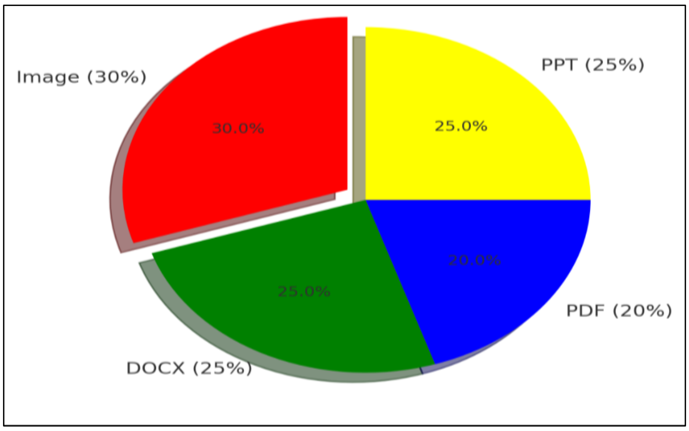


Figure 15: File Format Distributions

The pie chart illustrating the distribution of file formats (Figure 15) processed by the OCR system offers valuable insights into the variety of input data handled by the system. With image files representing 30% of the total distribution, it's evident that users frequently input documents in image format, underlining the system's importance in effectively extracting text from images. Similarly, the 25% share of DOCX documents indicates the prevalence of structured documents in digital formats, such as reports and essays, suggesting users' need to extract text from these files (26). PDF files, constituting another 30% of the distribution, are commonly used for document sharing while preserving formatting, indicating the system's role in extracting text from these widely used files. PPTs, comprising 25% of the distribution, suggest users' requirement to extract text from presentation slides, possibly for accessibility or content repurposing purposes. The distribution across multiple file-formats underscores the diverse sources of input data processed by the OCR system. Users input documents in various formats based on their specific needs and preferences, highlighting the importance of the system's versatility in handling different file types. The balanced distribution across image, DOCX, PDF, and PPT formats demonstrates the OCR system's adaptability and compatibility with a wide range of file types. This versatility allows the system to cater to diverse user requirements effectively. Each file format presents its own set of challenges for text extraction, ranging from the structured layout of DOCX documents to the visual elements in PDFs and PowerPoint presentations (27). The distribution reflects the system's capability to navigate through these complexities and extract text accurately. The distribution may also reflect user preferences for input formats based on factors such as document accessibility, ease of sharing, and compatibility with other software applications. Understanding these preferences can help tailor the OCR system to better meet user needs. The distribution provides valuable insights for system designers in terms of optimizing the OCR system for handling different file formats. It highlights the need for robust algorithms and techniques tailored to the specific characteristics of each format. Analysing the distribution can identify areas for enhancement or optimization within the OCR system (28). For example, if PDFs constitute a significant portion of the input data, improving text extraction from complex PDF structures may be prioritized for future development. The distribution underscores the importance of considering user experience when designing and implementing the OCR system. Ensuring seamless integration with commonly used file formats can enhance user satisfaction and usability (29). This aspect delves into the efficiency of the OCR system's audio output file sizes, offering insights into the varying sizes generated from different input file types. It serves as a vital metric for evaluating the system's performance in converting textual content into audio format while considering storage constraints and bandwidth implications. The comparison encompasses a spectrum of input file formats processed by the OCR system, including images, DOCX documents, PDFs, and PPT files. Each file type presents unique characteristics that influence the resulting audio file size, thereby impacting user experience and system efficiency (30). Measured in kilobytes (KB) or another appropriate unit, the file size denotes the volume of data comprising the audio output file generated from each input type. Variations in file sizes reflect differences in encoding techniques, compression algorithms, and content complexity inherent in the conversion process (31). Understanding the disparities in file sizes facilitates informed decision-making regarding storage management and transmission considerations. Smaller audio output files may be favoured for resource-efficient storage and seamless network transmission, enhancing accessibility and user convenience (32). Analysing these differences informs optimization strategies aimed at enhancing the efficiency of the audio conversion process. Techniques such as audio encoding refinement and compression optimization can be implemented to minimize file sizes without compromising audio quality. User preferences and requirements play a pivotal role in determining the optimal balance between audio file size and quality. Tailoring the conversion process to align with user expectations ensures a satisfactory user experience and fosters user engagement and satisfaction.

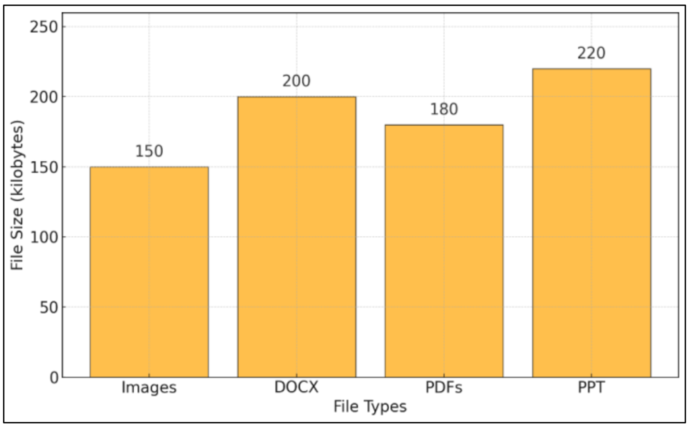


Figure 16: Audio Output File Size Comparison

By evaluating the efficiency of audio output file sizes across different input file types, opportunities for system performance enhancement are identified. Continuous refinement and iteration of the OCR system's audio conversion capabilities contribute to ongoing improvements in user satisfaction and system efficacy (33). The bar graph (Figure 16) representing the "Audio Output File Size Comparison" provides a visual depiction of the sizes of audio files generated by the OCR system for different input file formats. The graph illustrates significant variations in the sizes of audio files generated from different input file formats, ranging from 150 KB to 220 KB. Each bar on the graph corresponds to a specific input file format, namely images, DOCX documents, PDFs, and PPT presentations. The height of each bar represents the size of the audio file generated from the respective input format. By comparing the heights of the bars, viewers can easily discern the relative differences in audio file sizes across different input formats. This comparison aids in understanding the efficiency of the OCR system in converting text to speech for various types of content (34). The bar representing audio files generated from images is the shortest, indicating the smallest file size of 150 KB. This suggests that textual content extracted from images undergoes efficient compression during the conversion process, resulting in compact audio files. The bar corresponding to audio files generated from PDFs is slightly taller, indicating a size of 180 KB. PDFs often contain a mix of textual and visual elements, which may influence the size of the resulting audio files. Despite this, the size remains moderate compared to other formats. The bars representing audio files generated from DOCX documents and PPT presentations are taller, indicating larger file sizes of 200 KB and 220 KB, respectively. This suggests that structured documents and slide-based content may require additional processing and compression during conversion (35).

Despite variations in file sizes, the graph demonstrates the efficiency of the OCR system in converting text to speech while minimizing storage footprint. The relatively small to moderate sizes of audio files indicate effective compression algorithms and resource utilization. The graph provides valuable insights for storage planning and management. Stakeholders can use this information to allocate resources effectively and optimize storage capacities based on the expected file sizes of audio outputs. Compact audio file sizes contribute to a seamless user experience by reducing download times and bandwidth consumption. Users can access and playback audio content efficiently across different devices and platforms. The efficient conversion of text to speech and compact audio file sizes enhances the accessibility of digital content for users with diverse needs. Compact audio files are easier to distribute, share, and access, making information more readily available to a wider audience (36). Lastly, the graph highlights potential areas for optimization and refinement in the OCR system. Further enhancements in compression algorithms and conversion processes could lead to even smaller audio file sizes without compromising quality, improving overall system efficiency and performance.

Despite the system's promising performance, several limitations have been acknowledged. The reliance on rule-based text segmentation in Tesseract may lead to decreased accuracy when processing heavily stylized or low-contrast documents compared to AI-based OCR models. Furthermore, the absence of adaptive learning mechanisms restricts the system's ability to improve dynamically from user corrections or evolving input patterns. In terms of scalability, the processing time increases noticeably when dealing with batch conversions involving large file sizes or high-resolution images, indicating the need for computational resource optimization. When benchmarked against deep learning-based OCR frameworks such as Google Cloud Vision or Amazon Textract, it was observed that while those models offer higher accuracy and context-aware extraction, they require cloud-based infrastructure and incur significant operational costs. The present system, by contrast, is better suited for lightweight local deployment with offline functionality, making it ideal for resource-constrained environments or privacy-sensitive applications. These trade-offs underscore the importance of future enhancements that integrate AI-driven text detection while retaining the benefits of open-source modularity.

Conclusion

In conclusion, this paper has outlined the successful implementation of an automated text-to-audio conversion system, showcasing its adaptability across various file types. The work achieved significant advancements in OCR accuracy, and accessibility, particularly through the integration of pre-processing techniques, leading to improved recognition of text within images of diverse qualities. The system effectively handled structured document formats, PDFs with embedded images, and PPTs, demonstrating its versatility in navigating through complex file structures. The user-friendly interface and efficient output storage enhanced the overall usability and accessibility of the system. Robust error-handling mechanisms ensured reliable performance, even in unforeseen scenarios, contributing to the system's resilience. User satisfaction survey results gained the positive impact of the system on user experience, with an intuitive interface and clear feedback mechanisms. The efficient handling of diverse document formats highlights the system's potential applications in various fields. The system findings contribute to the growing body of knowledge in automated text-to-audio conversion, offering practical solutions for enhancing the user experience. Several limitations of the current system were identified during the evaluation. The system's dependency on rule-based OCR through Tesseract restricts its adaptability when processing highly stylized fonts, handwritten inputs, or documents with complex layouts such as overlapping elements and dynamic formatting. Additionally, real-time processing was found to be constrained by hardware performance, especially when handling large batch files or high-resolution media, which can lead to delays in output generation. Multilingual accuracy, while functional, may vary depending on script complexity and font clarity, with performance notably reduced for low-resource languages. To overcome these constraints, future enhancements may involve the integration of deep learning-based OCR models capable of self-improvement through feedback and learning. Incorporating real-time language detection and adaptive segmentation algorithms could also increase versatility. Moreover, optimization for mobile and edge devices is recommended to enable offline functionality and reduce reliance on cloud-based resources, thereby enhancing scalability and accessibility across diverse user environments.

Abbreviations

gTTS: Google Text-To-Speech, OCR: Optical Character Recognition, PDF: Portable Document Format, PPT: PowerPoint Presentation.

Acknowledgement

We want to express our sincere gratitude for the valuable support and contributions received during this research. We also appreciate the assistance provided by Annapoorana Engineering College (Autonomous) for their continued encouragement and support.

Author Contributions

SM Kamali: Conceptualization, Methodology, Writing - Original Draft, V Malathy: Data Curation, Formal Analysis, Gurumuni Nathan Uma Devi: Software, Validation, SK Deepa: Investigation, Resources, M Anand: Visualization, Supervision, S Vaishnodevi: Project Administration, Funding Acquisition, Ratchagaraja Dhairiyasamy: Writing - Review and Editing.

Conflict of Interest

The authors declare no conflict of interest.

Ethics Approval

This study did not involve any human or animal subjects, and therefore, ethics approval was not required.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

1. Karthikeyan V, Priyadharsini SS. Text-independent voiceprint recognition via compact embedding of dilated deep convolutional neural networks. Computers and Electrical Engineering. 2024 Sep 1;118:109408.
2. Raheel A. Emotion analysis and recognition in 3D space using classifier-dependent feature selection in response to tactile enhanced audio–visual content using EEG. Computers in Biology and Medicine. 2024 Sep 1;179:108807.
3. Ristea NC, Anghel A, Ionescu RT. Cascaded cross-modal transformer for audio–textual classification. Artificial Intelligence Review. 2024 Aug 2;57(9):225.
4. Udawatta P, Udayangana I, Gamage C, Shekhar R, Ranathunga S. Use of prompt-based learning for code-mixed and code-switched text classification. World Wide Web. 2024;27(5):63.

5. Das A, Sarma MS, Hoque MM, Siddique N, Dewan MAA. AVaTER: Fusing Audio, Visual, and Textual Modalities Using Cross-Modal Attention for Emotion Recognition. Sensors. 2024;24(18):5862.
6. Lei J, Wang J, Wang Y. Multi-level attention fusion network assisted by relative entropy alignment for multimodal speech emotion recognition. Applied Intelligence. 2024;54(17–18):8478–90.
7. Imbwaga JL, Chittaragi NB, Koolagudi SG. Explainable hate speech detection using LIME. Int J Speech Technol. 2024;27(3):793–815.
8. Huang Y, Chen S, Chen Y, Feng J, Deng C. Enhancing Task-Oriented Dialogue Modeling through Coreference-Enhanced Contrastive Pre-Training. Applied Sciences. 2024 Aug 28;14(17):7614.
9. Mao J, Shi H, Li X. Research on multimodal hate speech detection based on self-attention mechanism feature fusion. The Journal of Supercomputing. 2025 Jan;81(1):28.
10. Hashmi E, Yayilgan SY, Yamin MM, Abomhara M, Ullah M. Self-supervised hate speech detection in norwegian texts with lexical and semantic augmentations. Expert Systems with Applications. 2025 Mar 10;264:125843.
11. Moreno-Galván DA, López-Santillán R, González-Gurrola LC, Montes-Y-Gómez M, Sánchez-Vega F, López-Monroy AP. Automatic movie genre classification and emotion recognition via a BiProjection Multimodal Transformer. Information Fusion. 2025 Jan 1;113:102641.
12. Tiwari M, Verma DK. Enhanced text-independent speaker recognition using MFCC, Bi-LSTM, and CNN-based noise removal techniques. Int J Speech Technol. 2024;27(4):1013–26.
13. Kavitha D, Mala GA. A Brief Survey of Text Mining: Domains, Implemented Algorithms and Evaluation Metrics. Journal of Information and Knowledge Management. 2024 Dec 26;23(06):2450078.
14. Shang Y, Fu T. Multimodal fusion: A study on speech-text emotion recognition with the integration of deep learning. Intelligent Systems with Applications. 2024 Dec 1;24:200436.
15. Shilpashree S, Ashoka DV. F-DenseCNN: feature-based dense convolutional neural networks and swift text word embeddings for enhanced hate speech prediction. Social Network Analysis and Mining. 2024 Sep 24;14(1):192.
16. Mehra S, Ranga V, Agarwal R. Multimodal Integration of Mel Spectrograms and Text Transcripts for Enhanced Automatic Speech Recognition: Leveraging Extractive Transformer-Based Approaches and Late Fusion Strategies. Computational Intelligence. 2024 Dec;40(6):e70012.
17. Mori D, Ohta K, Nishimura R, Ogawa A, Kitaoka N. Recognition of target domain Japanese speech using language model replacement. EURASIP Journal on Audio, Speech, and Music Processing. 2024 Jul 20;2024(1):40.
18. Shou Y, Meng T, Ai W, Zhang F, Yin N, Li K. Adversarial alignment and graph fusion via information bottleneck for multimodal emotion recognition in conversations. Information Fusion. 2024 Dec 1;112:102590.
19. Khanduja N, Kumar N, Chauhan A. Telugu language hate speech detection using deep learning transformer models: Corpus generation and evaluation. Systems and Soft Computing. 2024 Dec 1;6:200112.
20. Kim M, Jang GJ. Speaker-Attributed Training for Multi-Speaker Speech Recognition Using Multi-Stage Encoders and Attention-Weighted Speaker Embedding. Applied Sciences. 2024 Sep 10;14(18):8138.
21. Kanwal T, Mahum R, AlSalman AM, Sharaf M, Hassan H. Fake speech detection using VGGish with attention block. EURASIP Journal on Audio, Speech, and Music Processing. 2024 Jun 26;2024(1):35.
22. Aggarwal S, Vishwakarma DK. Exposing the Achilles' heel of textual hate speech classifiers using indistinguishable adversarial examples. Expert Systems with Applications. 2024 Nov 15;254:124278.
23. Badri N, Kboubi F, Habacha Chaibi A. Abusive and Hate speech Classification in Arabic Text Using Pre-trained Language Models and Data Augmentation. ACM Transactions on Asian and Low-Resource Language Information Processing. 2024 Nov 21;23(11):1-28.
24. Aloshban N, Esposito A, Vinciarelli A, Guha T. On the effects of obfuscating speaker attributes in privacy-aware depression detection. Pattern Recognit Lett. 2024;186:300–5.
25. Parlak C, Altun Y. A Quest for Formant-Based Compact Nonuniform Trapezoidal Filter Banks for Speech Processing with VGG16. Circuits Syst Signal Process. 2024;43(11):7309–38.
26. Madan A, Kumar D. CNN-based models for emotion and sentiment analysis using speech data. ACM Transactions on Asian and Low-Resource Language Information Processing. 2024 Oct 23;23(10):1-24.
27. Chen Y, Zhu W, Yu W, Xue H, Fu H, Lin J, Jiang D. Prompt Learning for Multimodal Intent Recognition with Modal Alignment Perception. Cognit Comput. 2024;16(6):3417–28.
28. Chen S, Kruger J-L. Visual processing during computer-assisted consecutive interpreting: Evidence from eye movements. Interpreting. 2024;26(2):231–52.
29. Raut R, Spezzano F. Enhancing hate speech detection with user characteristics. Int J Data Sci Anal. 2024;18(4):445–55.
30. Shixin P, Kai C, Tian T, Jingying C. An autoencoder-based feature level fusion for speech emotion recognition. Digital Communications and Networks. 2024;10(5):1341–51.
31. Dong L, Wang W, Yu Z, Huang Y, Guo J, Zhou G. Pronunciation guided copy and correction model for ASR error correction. International Journal of Machine Learning and Cybernetics. 2024;15(10):4787–99.
32. Yurtay Y, Demirci H, Tiryaki H, Altun T. Emotion Recognition on Call Center Voice Data. Applied Sciences. 2024 Oct 16;14(20):9458.
33. Mamun KA, Nabid RA, Pranto SI, Lamim SM, Rahman MM, Mahammed N, Huda MN, Sarker F, Khan RR. Smart reception: An artificial intelligence driven bangla language based receptionist system employing speech, speaker, and face recognition for automating reception services. Engineering Applications of Artificial Intelligence. 2024 Oct 1;136:108923.

34. Mehra S, Ranga V, Agarwal R, Susan S. Speaker independent recognition of low-resourced multilingual Arabic spoken words through hybrid fusion. Multimed Tools Appl. 2024;83(35):82533–61.
35. Nassif AB, Shahin I, Nemmour N. Emotional speaker identification using PCAFCM-deepforest with fuzzy logic. Neural Comput Appl. 2024;36(30):18567–81.
36. Tits N, Bhatnagar P, Dutoit T. Text-Independent Phone-to-Audio Alignment Leveraging SSL (TIPAA-SSL) Pre-Trained Model Latent Representation and Knowledge Transfer. Acoustics. 2024;6(3):772–81.