ISSN 2277-2685
IJESR/June. 2025/ Vol-15/Issue-3s/611-622
P.Shravya *et. al.*, / International Journal of Engineering & Science Research

Scene to Text Conversion and Pronunciation for Visually Impaired People

P.Suresh Kumar, P.Shravya, G.Sravani, R.Swetha

[1] Associate Professor, Department Of ECE, Bhoj Reddy Engineering College For Women, India.

[2,3,4]B. Tech Students, Department Of ECE, Bhoj Reddy Engineering College For Women, India.

ABSTRACT

*Access to written information is a critical challenge for visually impaired individuals. This project, **"Scene to Text Conversion and Pronunciation for Visually Impaired People,"** aims to address this challenge by developing a syste capable of identifying text in captured images and converting it into audible output. The system utilizes the **Maximally Stable Extremal Regions (MSER) algorithm** to detect and extract text regions from various scenes effectively. By integrating Optical Character Recognition (OCR), the extracted text is converted into a digital format for further processing. The converted text is then transformed into speech using a text-to-speech (TTS) engine, providing users with real-time audio output. The combination of MSER and OCR ensures precise and efficient text detection and recognition, even in complex environments with varying lighting and background conditions. This technology is designed to empower visually impaired individuals by making printed and digital text content more accessible.*

The project is implemented using MATLAB software, which provides a robust platform for executing the algorithms and processing images efficiently. By enabling seamless conversion of visual text to speech, this system contributes to dependence and daily l

1-INTRODUCTION

Optical Character Recognition (OCR) is one of the most important fields in the pattern recognition domain, capable of recognizing handwritten characters, irregularly spaced characters, and machine-printed characters. Generally, a character recognition system consists of five major tasks (as shown in Fig. 1): pre-processing, segmentation, feature extraction, classification, and recognition. Pre-processing includes thresholding and determining the size and aspect ratio of images, which are then normalized. Thresholding is applied to remove noise and separate the background from the foreground. Image segmentation follows as a critical step, clustering the pixels of the image to extract text-relevant regions. Subsequently, feature extraction techniques are applied to retrieve all useful characteristics of the text and reduce recognition errors. While a variety of feature extraction methods exist, selecting the most suitable one is essential for achieving high accuracy in character recognition.

The performance of recognition heavily depends on the classifier used for the specific application. Various classifiers such as Support Vector Machines (SVMs), Artificial Neural Networks (ANNs), and Hidden Markov Models (HMMs) have been explored to enhance OCR performance. Replicating the human reading process through machines has long been a key research area in pattern recognition and machine learning. Despite significant advancements, achieving robust OCR performance on degraded text—such as broken, merged

characters or noisy inputs—remains a challenging task without human intervention. Advanced models that use a combination of ANNs and HMMs have been developed to address these challenges and have also contributed to cognitive psychology by examining human word shape perception and letter position importance during visual word recognition.

The motivation for developing OCR systems stems from their wide range of applications in sectors like banking, healthcare, publishing, and government. For example, OCR enables instant cheque processing in banks, automates data entry from paper-based medical records, and facilitates the digitization of books for searchable online archives. In industries and offices, OCR technology simplifies document management, automates address reading in mail sorting, and supports financial tracking through invoice imaging tools. Essentially, OCR transforms two-dimensional text images—whether printed or handwritten—into machine- readable formats, typically using a pipeline process involving pre-processing, layout analysis, character recognition, and language modeling. Pre-processing may include binarization, noise removal, and skew correction, while layout analysis determines reading order and text structure. Language modeling improves recognition accuracy by leveraging domain-specific vocabulary.

Despite decades of advancement and the availability of commercial OCR systems, OCR outputs often still include errors, especially with degraded or low-resolution inputs. One of the major challenges remains character segmentation, particularly in aged, scanned, photocopied, or faxed documents. Degradations can lead to touching characters, broken symbols, or smudged text due to ink or image artifacts.

In phase 2 of our project, a basic OCR-based system was developed using MATLAB software, the MSER (Maximally Stable Extremal Regions) algorithm for text region detection, and standard OCR methods for character recognition. This system successfully converted scene text into speech for the visually impaired but was limited to recognizing characters only in the Times New Roman font style. While this phase demonstrated the foundational capabilities of scene-to-text conversion and pronunciation, its font-specific nature restricted practical usability in real-world scenarios.

To address this limitation, phase 2 of the project introduced significant improvements. The enhanced system is now capable of recognizing multiple font styles, making it far more robust and adaptable. This phase maintains the MSER-based text detection while upgrading the OCR pipeline to support font-independent recognition. The updated system handles various print styles more effectively, allowing it to process text from diverse sources such as signage, printed forms, product labels, and advertisements—thereby improving its utility for visually impaired users. Moreover, the feature extraction and classification techniques have been fine- tuned to reduce recognition errors and improve accuracy even under noisy or degraded conditions.

With these enhancements, the system advances towards the broader goal of building a font-independent, accurate, and user-friendly text-to-speech assistive technology. By combining the solid foundational structure from Phase 1 with the comprehensive recognition capability in Phase 2, the project aims to bring a reliable, real-time reading aid closer to practical deployment for the visually impaired community.

2-LITERATURE SURVEY

One of the significant research contributions in the area of feature extraction for character recognition involves a comparative study of four different

classifiers applied on Malaysian license plate characters using enhanced geometrical topological feature analysis. These classifiers include Bayesian networks, artificial immune recognition systems, support vector machines (SVM), and neural networks. The goal of the study was to determine the most efficient classification technique for accurate character recognition. Among them, the SVM yielded the highest accuracy, though it required more processing time per character compared to neural networks. This indicates a trade-off between accuracy and processing speed, which is crucial when designing real-time recognition systems like the one proposed in our project.

In **Phase 1** of our project, character recognition was limited to printed text in **Times New Roman** font using the OCR approach. Although OCR provided decent accuracy for clean, printed texts, its limitations were evident when the system was applied to varied fonts or noisy, real-world images. In **Phase 2**, to overcome these challenges and ensure font- independence, our system incorporates advanced feature extraction techniques and supports recognition of diverse font styles and text structures. Inspired by methodologies used in historical document analysis, our system improves accuracy by incorporating spatial features such as.

A related technique applied for historical Greek documents utilized an SVM-based hierarchical classification approach with high granularity, which effectively distinguished similar-looking characters. The results showed exceptional recognition accuracy — 94.51% for handwritten and 97.71% for printed texts — proving the reliability of SVM in handling complex, low-quality, or aged documents. Drawing from this, our project in Phase 2 is now equipped to process both clean and degraded text, even from scanned or low-resolution images.Furthermore, texture analysis using statistical and signal processing methods is another area explored for character recognition. These methods convert image textures into new formats for feature extraction. However, they tend to obscure human-interpretable features and are vulnerable to noise introduced during the conversion process. Moreover, researchers like Ding et al. noted that the recognition accuracy of Gabor filters — commonly used in texture analysis — decreases as the number of target classes increases. Similarly, Tuceryan, Jain, and Petrou emphasized the high computational cost associated with signal processing techniques.

Our project addresses these limitations by integrating selective preprocessing techniques, such as binarization, skew correction, and noise removal, combined with robust OCR and MSER algorithms. As we transitioned from Phase 1 to Phase 2, the system evolved from recognizing a single font type to accurately processing multilingual and stylistically varied texts, offering higher recognition rates and broader usability for the visually impaired community.

The Gray Level Co-occurrence Matrix (GLCM) method, proposed by Haralick et al., is one of the foundational techniques used for global texture-based feature extraction in image analysis. GLCM operates by analyzing how often pairs of pixel intensities occur in an image, resulting in a matrix that statistically represents textural patterns. From this matrix, up to 36 unique features can be derived, including contrast, homogeneity, energy, and entropy. These features are widely used in character recognition, medical imaging, and document analysis due to their robustness in texture differentiation. However, GLCM is limited when applied to complex scripts or calligraphy styles, especially where directional context and edge details are critical.

One of the techniques that has been applied in this

paper, which is named EDMS, discussed a feature extraction method for optical font recognition that is an Arabic calligraphy script image proposed by Bilal Bataineh et al. The result of the proposed method had been compared with the Gray Level Co-occurrence Matrix (GLCM) method..

The second matrix, which is 3 × 3, is considered an edge direction matrix and contains the relationship representation of each pixel. By measuring the occurrence of each value in EDM2, the most important pixel relationships were identified.

By using these two matrices, various features can be extracted such as Homogeneity, Contrast, Angular Second Moment (ASM), Entropy, Energy, and Correlation with 0°, 45°, 90°, and 135° angles, which are 28 features. Three different classifiers were used for the purpose of classification.

3-SOFTWARE REQUIREMENTS

In this chapter we will discuss about software requirements for scene to text conversion and pronunciation for visually impaired people.

Software requirements

Operating system	:	Windows 10
Coding language	:	MATLAB language
Tool	:	MATLAB 2016A

MATLAB, short for Matrix Laboratory, is a high-performance software platform specifically designed for numerical computing, algorithm development, and data visualization. Developed by MathWorks, MATLAB is widely used across engineering, scientific research, and academic fields for its powerful computational capabilities and user-friendly environment. Its matrix-centric architecture makes it particularly suitable for operations involving linear algebra, image processing, and signal analysis, which are essential for this project

One of MATLAB's standout features is its extensive collection of built-in toolboxes. These toolboxes provide specialized functions and tools that extend the software's capabilities for domain-specific tasks. For instance, the Image Processing Toolbox is essential for tasks such as object detection, feature extraction, and morphological operations, while the Deep Learning Toolbox simplifies the implementation and training of deep learning models like Convolutional Neural Networks (CNNs). These toolboxes are critical to the success of this project, as they support efficient text detection and character recognition workflows.

MATLAB's support for the MSER (Maximally Stable Extremal Regions) algorithm is pivotal for identifying text regions in images. This algorithm is renowned for its accuracy and efficiency in locating text in complex backgrounds, making it highly suitable for our project. MATLAB also seamlessly integrates OCR (Optical Character Recognition) functionality, allowing for the direct extraction of textual information from images.

Additionally, its ability to integrate pre-trained deep learning models ensures accurate and robust character recognition, even in challenging scenarios. MATLAB offers an interactive environment that enables real-time debugging and testing, which significantly accelerates the development process. Its visual interface and tools for plotting allow developers to observe intermediate results and adjust

parameters dynamically.

For projects involving image analysis and recognition, MATLAB provides rich visualization tools, enabling clear representation of detected features, processed regions, and classification results.

The platform also supports multi-language integration, allowing the inclusion of external libraries or custom functions written in C, C++, Python, or Java. This flexibility is useful for extending the project's functionality or integrating additional features if needed.

To run MATLAB effectively, certain hardware and system requirements must be met. The software is compatible with Windows, macOS, and Linux operating systems. A minimum of 8 GB RAM and a multi-core processor are recommended for smooth performance, especially for tasks involving deep learning and image processing. If the project involves computationally intensive processes, like training CNN models, a GPU with CUDA support can be utilized to significantly enhance processing speed. MATLAB's built-in tools automatically optimize computations for available hardware, ensuring efficient resource utilization.

MATLAB stands out for its versatility in addressing diverse computational challenges, ranging from simple data analysis to advanced algorithm development. It provides a unified environment for handling various tasks, including image preprocessing, feature extraction, and neural network training, all of which are essential for this project. MATLAB's scalability ensures that it can handle projects of varying complexity, from small-scale prototypes to large- scale systems. Moreover, its ability to support scripting, functions, and modular programming promotes clean and reusable code, which is beneficial for collaborative development and future enhancements.

MATLAB's flexibility extends beyond the core toolboxes, offering support for creating custom functions and algorithms, which allows for tailored solutions to specific problems. For instance, the platform enables the development of specialized image processing pipelines to handle unique challenges in text detection and recognition, such as varying font styles, noise, and distortions in the input images. Its vast ecosystem of libraries and functions empowers users to integrate new techniques and approaches as the field of computer vision and deep learning evolves, ensuring that projects remain up-to-date with the latest advancements. Furthermore, MATLAB's built-in parallel computing capabilities allow users to distribute

enhancing the system's efficiency. This capability is particularly useful for large-scale data analysis, model training, and real-time applications, where performance and speed are critical.

Another major advantage of using MATLAB is its seamless integration with hardware, which is beneficial for projects that require real-time interaction or deployment on embedded

systems. MATLAB supports various hardware platforms and microcontrollers, including Arduino and Raspberry Pi, which enables rapid prototyping and testing of algorithms in real- world environments. This is crucial for projects involving image recognition and text-to-speech systems, as it allows the developed models to be tested on actual hardware, ensuring that the system works as intended under real-world conditions. The integration with cloud platforms also allows for the deployment of MATLAB applications in distributed environments, facilitating scalability and remote monitoring of projects. Moreover, the ability to generate standalone applications from MATLAB code, along with the option to compile code into executable files, enhances the software's accessibility and portability

across different platforms. This ensures that solutions can be shared with stakeholders or integrated into larger systems without requiring users to have MATLAB installed.

4-SCENE TO TEXT CONVERSION AND PRONUNCIATION

Scene-to-text conversion is a transformative technology that plays a pivotal role in assistive systems for visually impaired individuals. The fundamental objective of this process is to bridge the communication gap between the visual world and users who are unable to perceive it through sight. By interpreting visual scenes and converting relevant information—especially textual content—into a machine-readable and audible format, such systems empower users to better understand their surroundings and make informed decisions.

The core principle of scene-to-text systems lies in their ability to extract and recognize textual elements from images and translate them into spoken language using text-to-speech (TTS) technology. This enables users to receive real-time auditory feedback about their environment, aiding them in tasks such as reading signs, understanding labels, identifying documents, and navigating public spaces. As a result, these systems enhance the autonomy and quality of life for visually impaired individuals.

To achieve effective scene interpretation, advanced computer vision and pattern recognition techniques are required. Among them, Optical Character Recognition (OCR) and Maximal Stable Extremal Regions (MSER) are key technologies. OCR is the process by which text within digital images is detected and converted into an editable or readable format. It is capable of recognizing printed or handwritten characters and translating them into machine-encoded text. MSER, meanwhile, is an algorithm used in image processing to detect regions that are stable over a range of intensity thresholds—typically where text is found—making it ideal for localizing characters or word blocks in complex scenes.

Existing system

Existing systems for scene-to-text conversion intended for visually impaired individuals have contributed significantly to the advancement of assistive technology. These systems typically utilize classical image processing methods such as thresholding and template matching to detect and recognize text from images. Thresholding is a widely used technique that simplifies image data by converting grayscale or color images into binary format. It separates the foreground (often text) from the background based on a fixed or adaptive intensity threshold. Although this method can be effective in structured and consistent lighting environments, it tends to perform poorly in real-world situations where lighting varies, shadows

Proposed system

At its core, the system utilizes a combination of advanced image processing and recognition algorithms, primarily Optical Character Recognition (OCR) and Maximal Stable Extremal Regions (MSER). OCR is employed to extract textual content from images, while MSER helps in identifying high-probability text regions within complex scenes. This combination allows the system to detect and isolate text more reliably, even when it is embedded in cluttered or visually noisy backgrounds. In Phase 2, the OCR engine is further trained and configured to recognize multiple font styles beyond Times New Roman, including serif, sans-serif, cursive, bold, italic, and stylized fonts that are commonly encountered in real-world scenes such as posters, signboards, product labels, and newspapers.To enhance the recognition process, the system incorporates several preprocessing steps aimed at

improving image clarity and optimizing text extraction. These include grayscale conversion, noise reduction, adaptive thresholding, morphological operations (such as dilation and erosion), contrast enhancement, and skew correction. These techniques help to standardize the input and emphasize text regions, resulting in higher recognition accuracy. The MSER algorithm plays a critical role in localizing text components regardless of their orientation or alignment, making the system more robust to skewed or rotated text.

Methodology

In Phase 2, the methodology evolves significantly to support more complex real-world scenarios where the system must accurately recognize and pronounce text that appears in various fonts, orientations, and environmental contexts. This stage emphasizes improved generalization, adaptability, processing efficiency, and user interaction. Each step of the original pipeline is refined and extended to meet these objectives.

1. Data Collection (Diverse and Augmented Dataset)

- To develop a robust system, a comprehensive and diverse dataset is compiled:
- Images are collected from real-world environments, such as classrooms, offices, street signs, supermarket labels, medicine packages, and public transport boards.
- Include multiple text styles: serif, sans-serif, cursive, bold, italics, all-caps, digital fonts, and printed handwriting.
- Ground truth annotations are created for training and evaluation, including bounding boxes and corresponding text labels.

5-RESULTS

Fig.1 input image with arial bold font style.

Fig.2 input image with sans-serif font style.

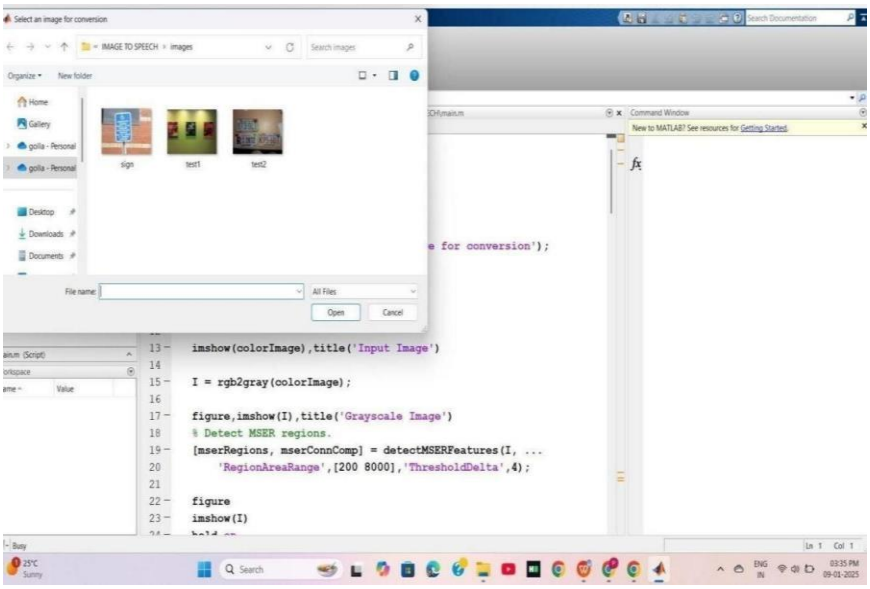

Fig.3 selection of input image

In above screen we can select the image to detect the text and processing.

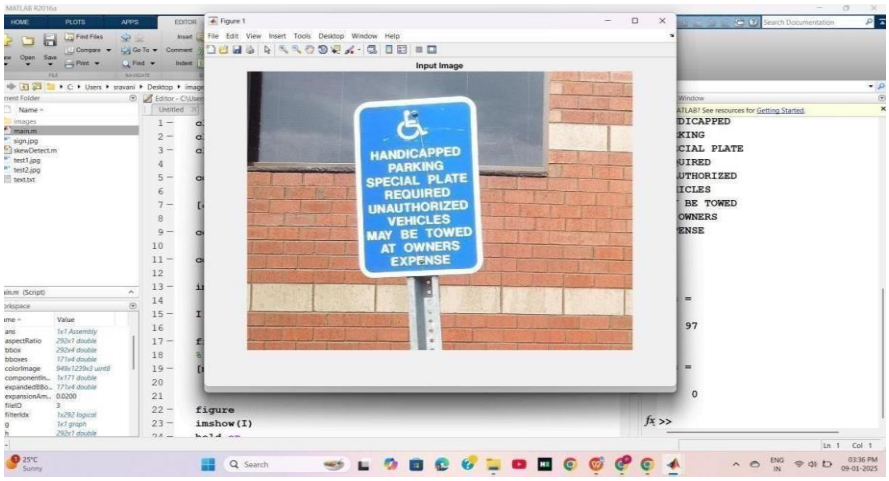

Fig.4 selected image

In above screen it is showing the selected image which is used for further processing

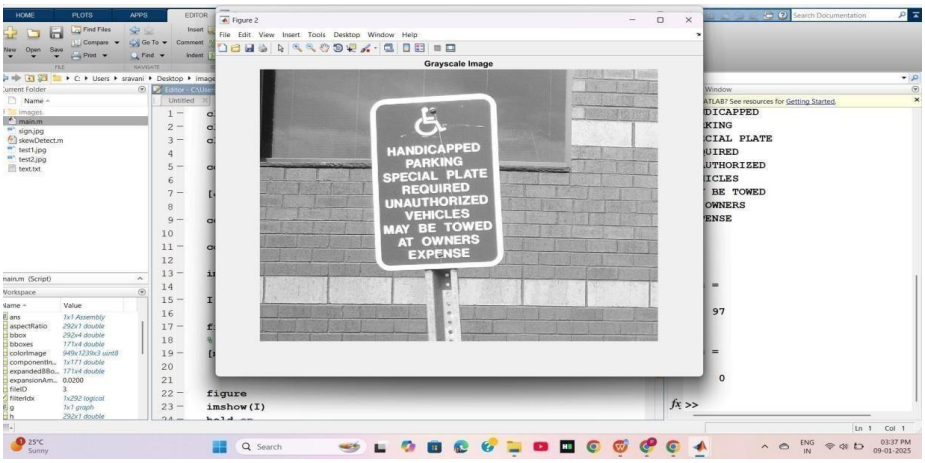

Fig.5 gray scale image

In above screen can see the image which is converted to gray scale image from rgb .This gray scale image has less complexity than rgb image

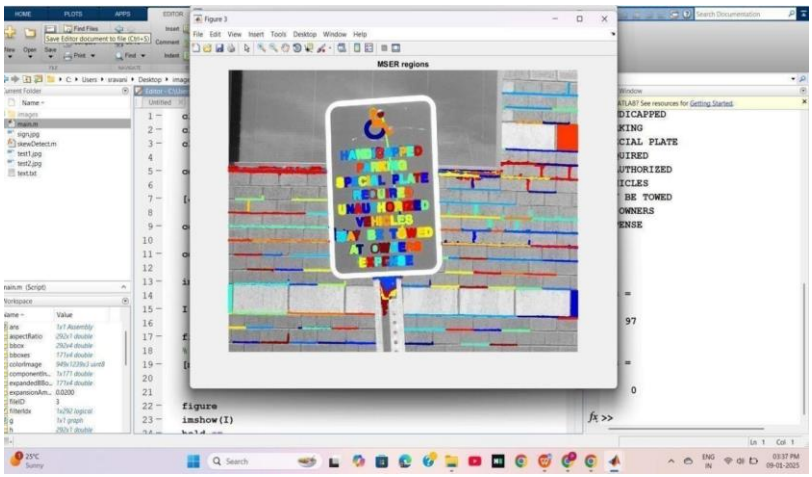

Fig.6 mser algorithm is applied

In above screen can see that MSER algorithm is applied to extract the text regions from the gray scale image.

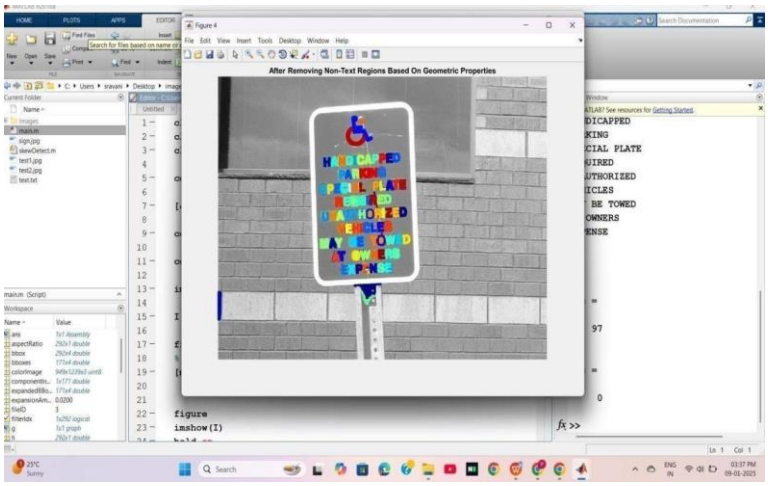

Fig.7 non-text region is removed

In above screen the unwanted text region is removed and only the required text is recognized for further procecssing.

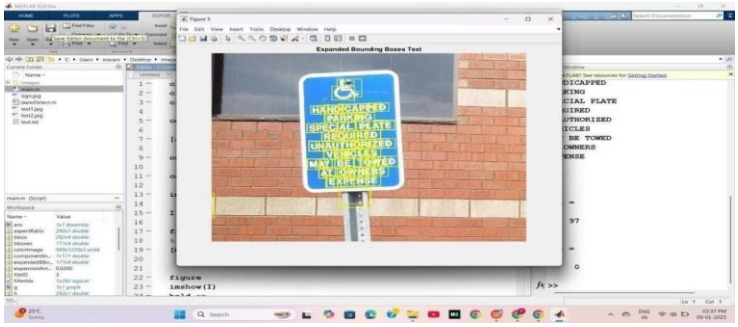

Fig.8 OCR is applied

In the avove screen for the text region the OCR algorithm is applied to conver the text into readable text.

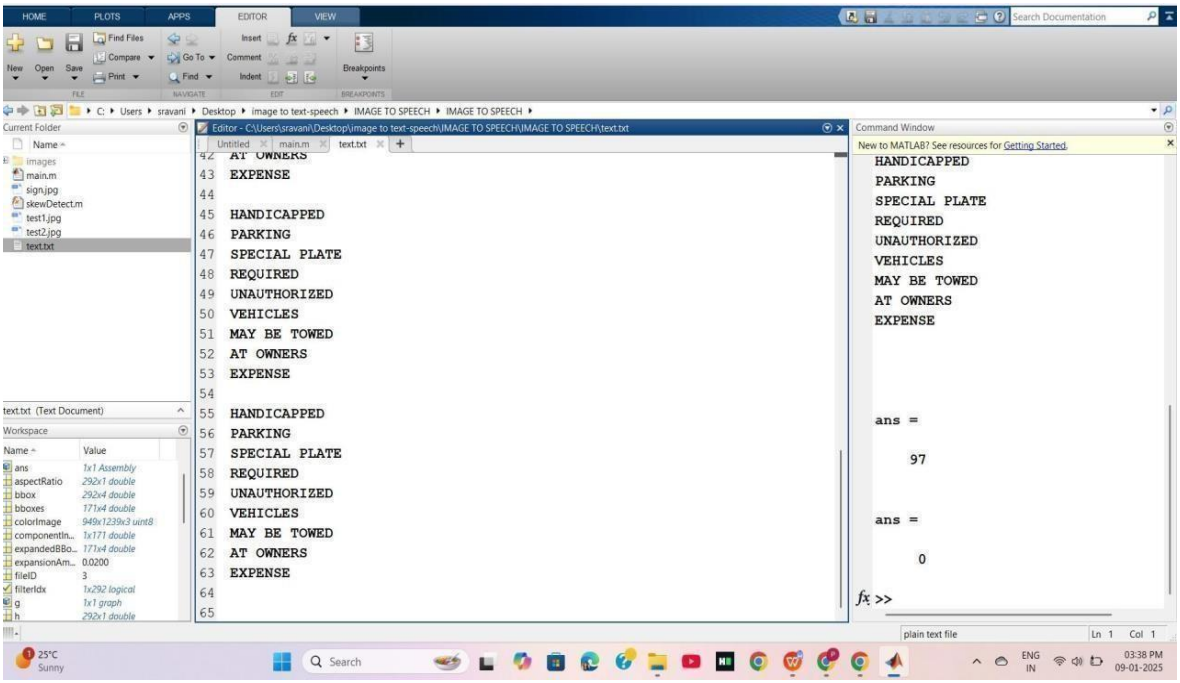

Fig.9 text file

6-CONCLUSION

The second phase of the scene-to-text conversion and pronunciation system marks a significant advancement in creating a more intelligent and accessible tool for visually impaired individuals. Building upon the foundational features of Phase 1, this enhanced version integrates more refined and adaptive techniques in both image processing and speech synthesis. Improvements in OCR accuracy and MSER segmentation now allow the system to handle more complex backgrounds, angled or curved text, and lower-quality images with greater reliability. Preprocessing methods have been optimized to adapt dynamically to varied lighting conditions, ensuring better clarity and consistency during text detection. Additionally, Phase 2 introduces preliminary support for handwritten text and a broader range of fonts and styles, further increasing the system's versatility in real-world scenarios.

The text-to-speech component has also been significantly improved, offering more natural and human-like speech output. Users are now able to personalize speech characteristics such as pitch, speed, and tone, making the system more comfortable and user-friendly in different environments. MATLAB's expanded functionality has been leveraged to introduce more robust error handling, reducing OCR-related inaccuracies by validating output against contextual language models.

While the system now covers a wider array of applications, certain limitations still exist, such as difficulties with very small text, highly stylized fonts, and highly cluttered backgrounds. However, these challenges offer a roadmap for future improvements. The groundwork laid in Phase 2 also sets the stage for multi-language support, enhanced mobile integration, and compatibility with wearable assistive devices, moving closer to a real-time, hands-free solution. Altogether, this phase significantly strengthens the system's practicality

and impact, empowering visually impaired users with greater independence, confidence, and the ability to interact more effectively with their environment.

REFERENCES

1. Abuhaiba, I.S.I.; Datta S.; and Holt, M.J.J. (1995), "Line Extraction and Stroke
2. Ordering of Text Pages", Proceedings of the 3rd International Conference on Document Analysis and Recognition, pp. 390-393.Amin, A.(2000), "Recognition of printed Arabic text based on global features and decision tree learning techniques", Pattern Recognition, Vol. 33, pp. 13091323.
3. Arica, N. and Vural, F. T. Y. (2001), "An overview of character recognition focused on offline handwriting", IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 31(2), pp. 216-233.
4. Arora, S.; Bhattacharjee, D.; Nasipuri, M.; Basu, D. K.; and Kundu, M. (2009), "Application of Statistical features in Handwritten Devanagari Character Recognition", International Journal of Recent Trends in Engg., Vol. 2(2), pp.
5. Arora, S.; Bhaattacharjee, D.; Nasipuri, M.; Malik, L.; Kundu, M.; and Basu, D. K.(2010), "Performance Comparison of SVM and ANN for Handwritten Devanagari Character Recognition", International Journal of Computer Science Issues, Vol. 7, Issue 3 (6), pp. 18- 26.
6. Arora, S.; Bhattacharjee, D.; Nasipuri, M.; Basu, D. K.; Kundu, M.; and Malik, L. (2009), "Study of different features on handwritten Devnagari characters", Proceedings of the international conference on Emerging Trends Engg.
7. Technol., pp. 929–933.
8. Bag, S.; Harit, G. (2013), "A survey on optical character recognition for Bangla and Devanagari scripts", Sadhana, Vol. 38, pp. 133-168.
9. Bansal, V. (1999), "Integrating Knowledge Sources in Devanagari Text Recognition", Ph. D. thesis, IIT Kanpur, India.
10. Bansal, V. and Sinha, R.M.K. (2000), "Integrating knowledge sources in Devanagari text recognition," IEEE Transactions- System Man Cybernetics. A: Syst. Hum., Vol. 30 (4), pp. 500–505.
11. Bansal, V. and Sinha, R. M. K. (2002), "Segmentation of touching and fused Devanagari characters", Pattern Recognition, Vol. 35(4), pp. 875-893.
12. Abuhaiba, I.S.I.; Datta S.; and Holt, M.J.J. (1995), "Line Extraction and StrokeOrdering of Text Pages", Proceedings of the 3rd International Conference on Document
13. Amin, A.(2000), "Recognition of printed Arabic text based on global features and decision tree learning techniques", Pattern Recognition, Vol. 33, pp. 13091323.
14. Arica, N. and Vural, F. T. Y. (2001), "An overview of character recognition focused on offline handwriting", IEEE Transactions on Systems, Man, and Cybernetics – Part C: Applications and Reviews, Vol. 31(2), pp. 216-233.
15. Arora, S.; Bhattacharjee, D.; Nasipuri, M.; Basu, D. K.; and Kundu, M. (2009), "Application of Statistical features in Handwritten Devanagari Character Recognition", International Journal of Recent Trends in

Engg., Vol. 2(2), pp.

16. 40–42.

17. Arora, S.; Bhaattacharjee, D.; Nasipuri, M.; Malik, L.; Kundu, M.; and Basu, D. K.(2010), "Performance Comparison of SVM and ANN for Handwritten Devanagari Character Recognition", International Journal of Computer Science Issues, Vol. 7, Issue 3 (6), pp. 18-26.

18. Arora, S.; Bhattacharjee, D.; Nasipuri, M.; Basu, D. K.; Kundu, M.; and Malik, L. (2009), "Study of different features on handwritten Devnagari characters", Proceedings of the international conference on Emerging Trends Engg.

19. Technol., pp. 929–933.

20. Bag, S.; Harit, G. (2013), "A survey on optical character recognition for Bangla and Devanagari scripts", Sadhana, Vol. 38, pp. 133-168.

21. Bansal, V. (1999), "Integrating Knowledge Sources in Devanagari Text Recognition", Ph.

22. D. thesis, IIT Kanpur, India.

23. Bansal, V. and Sinha, R.M.K. (2000), "Integrating knowledge sources in Devanagari text recognition," IEEE Transactions- System Man Cybernetics. A: Syst. Hum., Vol. 30 (4), pp. 500–505.

24. Bansal, V. and Sinha, R. M. K. (2002), "Segmentation of touching and fused Devanagari characters", Pattern Recognition, Vol. 35(4), pp. 875-893.