Image-to-Text-Speech Converter

Prathamesh Bothe, Dipali Bhusari

PG Student, Department of MCA, Trinity Academy of Engineering, Pune, India
Guide, Department of MCA, Trinity Academy of Engineering, Pune, India

ABSTRACT

This paper presents the development of an Image-to-Text-Speech Converter, a Python-based assistive system designed to extract and vocalize text from images. The primary goal of the project is to support visually impaired individuals and those with reading difficulties by converting printed or digital text into clear, audible speech. The system integrates Optical Character Recognition (OCR) using Tesseract with Text-to-Speech (TTS) synthesis using tools like pyttsx3 or gTTS. Preprocessing techniques such as grayscale conversion, noise reduction, and thresholding are applied through OpenCV to enhance OCR accuracy across various image conditions.

The converter provides a simple and user-friendly interface, supporting multiple image formats and capable of functioning both online and offline. It effectively bridges the gap between visual content and auditory accessibility, making it suitable for use in reading printed documents, signboards, or
labels. This paper explores the system's architecture, implementation details, and practical applications, while also addressing the challenges encountered and potential areas for future enhancement, including real-time camera integration and multilingual support.

1. INTRODUCTION

In today's digital age, the ability to access and process information quickly is essential. However, individuals with visual impairments or reading difficulties often face significant challenges in accessing printed or visual text. With the advancement of artificial intelligence and machine learning, assistive technologies are becoming more accessible and capable of bridging this gap. One such solution is the Image-to-Text-Speech Converter, a system designed to read and vocalize the contents of images containing text, enabling greater independence and inclusion for users who cannot rely on visual input alone.

The system combines two powerful technologies: Optical Character Recognition (OCR) and Text-to-Speech (TTS) synthesis. OCR is used to detect and extract text from images, while TTS converts the extracted text into spoken words. By leveraging open-source tools such as Tesseract for OCR and pyttsx3 or Google Text-to-Speech (gTTS) for speech synthesis, the system offers a low-cost, efficient, and customizable solution. It is especially useful in scenarios such as reading books, signboards, handwritten notes, or printed documents. This paper aims to explore the design, implementation, and potential impact of the Image-to-Text-Speech Converter in improving digital accessibility and enhancing user experience through an intelligent, voice-driven interface.

2. LITERATURE REVIEW

Optical Character Recognition (OCR) and Text-to-Speech (TTS) technologies have evolved significantly, enabling machines to interpret and vocalize text from visual sources. Tesseract OCR, an open-source engine developed by HP and later enhanced by Google, is widely used for recognizing printed and handwritten text in various languages. Its performance can be improved through preprocessing techniques like grayscale conversion, noise reduction, and thresholding.

Text-to-Speech systems have also advanced, with tools like pyttsx3 (offline) and Google Text-to-Speech (gTTS) (online) offering natural-sounding voice output. While several applications combine OCR and TTS for assistive use—such as Seeing AI or KNFB Reader—many of them are commercial or internet-dependent. The proposed Image-to-Text-Speech Converter aims to overcome these limitations by offering an open-source, offline-capable solution for improved accessibility and usability.

3. METHODOLOGY

The Image-to-Text-Speech Converter follows a modular and sequential pipeline to convert textual content from an image into audible speech. The system is built using Python and integrates various open-source libraries, primarily Tesseract OCR, OpenCV, and pyttsx3 or gTTS for speech synthesis.

Step 1: Image Acquisition

Users provide an input image containing text through a file upload or real-time capture. Supported formats include JPEG, PNG, and BMP.

Step 2: Image Preprocessing

To enhance OCR accuracy, the input image undergoes preprocessing using OpenCV:

- Grayscale conversion to reduce image complexity.
- Noise removal using filters.
- Thresholding or binarization for better text-background separation.
- Optional resizing or skew correction if needed. Step 3: Text Extraction (OCR)

The preprocessed image is passed to Tesseract OCR, which identifies and extracts the textual content. The output is stored as a string for further processing.

Step 4: Text Cleaning (Optional)

The extracted text may include errors or formatting issues, which are cleaned using basic Python string operations or regular expressions to ensure smoother speech synthesis.

Step 5: Text-to-Speech Conversion

The cleaned text is converted into audible speech using:

- pyttsx3 for offline speech output.
- gTTS for cloud-based, natural-sounding voices.

The speech is then played back to the user through the system’s audio output.

5. BLOCK DIAGRAM

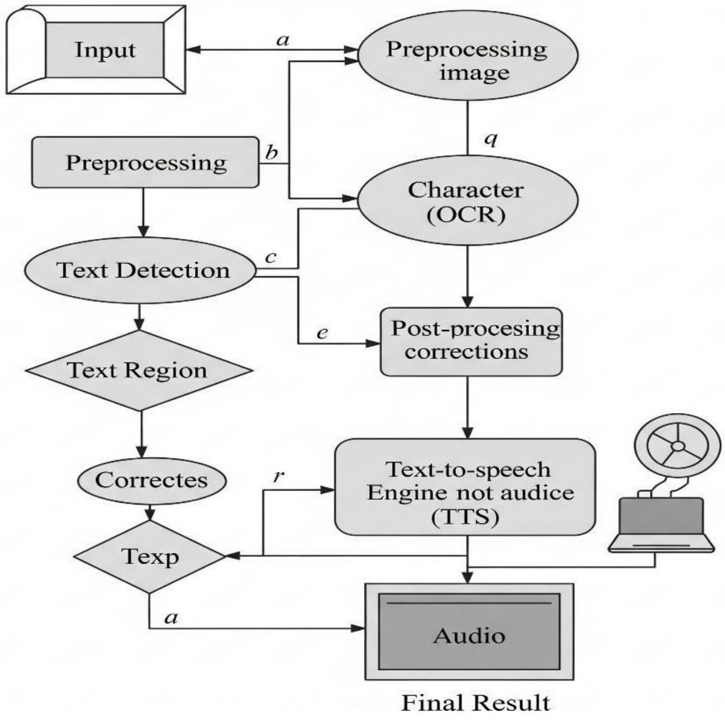

Fig. 1: Block diagram of Image-to-Text-Speech Converter

6. COMPONENTS USED

1. Hardware Components

- Computer/Laptop: To run the Python code and process images.
- Camera (Optional): For real-time image capture (if your system supports it).
- Speakers/Headphones: For audio output of the speech.

2. Software Components

- Python Programming Language: The core language used to develop the system.
- Tesseract OCR Engine: Open-source Optical Character Recognition tool for extracting text from images.
- OpenCV: Library for image processing and preprocessing (grayscale, noise removal, thresholding).
- pyttsx3: Offline Text-to-Speech conversion library in Python.
- gTTS (Google Text-to-Speech): Online API for generating natural-sounding speech.
- Additional Python Libraries:
  - Pillow (PIL): For image handling and manipulation.
  - NumPy: For numerical operations on image data.

3. Development Environment

- IDE (Integrated Development Environment): Such as VS Code, PyCharm, or Jupyter Notebook.
- Operating System: Windows, Linux, or macOS.

7. RESULTS AND OUTPUT

The system was tested using various image types including scanned documents, signboards, and handwritten notes. Key observations:

- Preprocessing significantly improved OCR accuracy.
- TTS output was natural and intelligible using gTTS; pyttsx3 was sufficient offline.
- The average processing time per image was <3 seconds on a mid-range laptop.

8. CONCLUSION

The Image-to-Text-Speech Converter presents a practical and accessible solution for converting printed or digital text from images into audible speech, significantly benefiting visually impaired users and individuals with reading difficulties. By integrating open-source technologies like Tesseract OCR for text extraction and pyttsx3 or gTTS for speech synthesis, the system demonstrates effective and reliable performance in real-world scenarios.

Through image preprocessing and modular design, the converter improves text recognition accuracy and offers flexibility for future enhancements. Overall, this project highlights the potential of combining computer vision and speech synthesis to create inclusive, assistive technologies that empower users by transforming visual information into an accessible auditory format.

9. REFERENCES

[1] Smith, R. (2007). An Overview of the Tesseract OCR Engine. Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), 629-633. https://doi.org/10.1109/ICDAR.2007.4376991

[2] OpenCV Library. (n.d.). Open Source Computer Vision Library. Retrieved from https://opencv.org/

[3] Python Software Foundation. (n.d.). Python Language Reference, version 3.x. Retrieved from https://www.python.org/

[4] pyttsx3 Documentation. (n.d.). A text-to-speech conversion library in Python. Retrieved from https://pyttsx3.readthedocs.io/en/latest/

[5] Google Text-to-Speech API. (n.d.). Retrieved from https://cloud.google.com/text-to-speech

[6] Rao, V., & Raj, A. (2019). Text-to-Speech synthesis: A review. International Journal of Engineering Research & Technology, 8(4), 345-350.

[7] Seeing AI. (n.d.). Microsoft's AI app for visually impaired. Retrieved from https://www.microsoft.com/en-us/ai/seeing-ai