Text Extraction and Translation using Machine Learning

Dr. Pallavi Deshpande, VIIT, Pune.
pallavi.deshpande@viit.ac.in
Yash Patil, VIIT, Pune.
yash.22110340@viit.ac.in
Sankalp Patil, VIIT, Pune.
Sankalp.22110188@viit.ac.in
Prajwal Rahate, VIIT, Pune.
prajwal.22110942@viit.ac.in
Adesh Mirjapure, VIIT, Pune.
adesh.22110457@viit.ac.in
Gaurav Suplekar, VIIT, Pune.
gaurav.22111329@viit.ac.in

Abstract

In this paper, we present the development of a web application that utilizes Flask and Python to convert file into text and audio files while offering multilingual translation capabilities. File may have any type like pdf, word, text, png, jpeg, etc. The system employs Optical Character Recognition (OCR) through Google Tesseract and Pytesseract to extract text from file, and Google Text-to-Speech (gTTS) for generating audio files from the extracted text. The web application is built using Flask framework, incorporating sessions for user interactions and efficient management of data. By integrating these technologies, users can upload file containing text, select their desired language for text extraction and audio output, and obtain corresponding text and audio files. The system also provides translation services for converting the extracted text into multiple languages, enhancing its accessibility and usability across diverse user groups. Set and as a template into which you can type your own text.

Keywords: *OCR, Tesseact, Pytesseract, gTTS, TTS, API, etc.*

1. Introduction

In today's digital age, the demand for efficient tools and applications that simplify tasks and enhance user experiences continues to grow. Among these tools, file to text and audio conversion applications have gained significant traction due to their ability to transform visual information into accessible formats. Such applications serve a variety of purposes, from aiding individuals with visual impairments to streamlining data extraction processes in various industries.

However, merely converting files into text or audio is not always sufficient, especially in our increasingly interconnected global community. Recognizing the importance of

multilingual support, it becomes imperative for these applications to offer translation capabilities, catering to the linguistic diversity of users worldwide.

In this paper, we introduce the development of a sophisticated web application designed to address these needs. Leveraging the power of Flask and Python, along with advanced technologies such as Optical Character Recognition

(OCR) and Text-to-Speech (TTS), our application empowers users to seamlessly convert files into both textual and auditory formats. Furthermore, by integrating translation services, the application extends its utility to users across different languages and cultures.

The remainder of this paper delves into the architectural design, implementation details, user interface considerations, evaluation metrics, and future prospects of our web application. Through this exploration, we aim to shed light on the intricacies of building such a system while highlighting its potential impact on accessibility, productivity, and inclusivity in our digital ecosystem.

2. Background and Related Work

The development of file to text and audio conversion applications has been facilitated by advancements in technologies such as Optical Character Recognition (OCR) and Text-toSpeech (TTS). These applications serve a wide range of purposes, including accessibility enhancement, data extraction, and language translation. In this section, we provide an overview of the key technologies and related work in this domain.

A. Optical Character Recognition (OCR):

Optical Character Recognition (OCR) technology enables the extraction of text from files, scanned documents, or other visual sources. It has evolved significantly over the years, with modern OCR engines achieving high accuracy rates even in complex scenarios. Google Tesseract and Pytesseract are among the most widely used OCR engines, offering robust text extraction capabilities across various languages and fonts.

B. Text-to-Speech (TTS):

Text-to-Speech (TTS) technology converts textual input into synthesized speech output, allowing users to listen to the content instead of reading it. TTS systems utilize natural language processing techniques to generate human-like speech, enhancing the user experience. gTTS (Google Text-to-Speech) is a popular TTS engine that provides highquality audio output in multiple languages.

C. Related Work:

Numerous file to text and audio conversion applications exist, each offering unique features and functionalities. Some notable examples include:

D. Online OCR Services:

Several online platforms offer OCR services, allowing users to upload files and extract text from them. These platforms often provide additional features such as language detection, document formatting, and API integration for seamless integration into other applications.

E. Multilingual Translation Services:

Translation services such as Google Translate and Microsoft Translator enable the translation of text between multiple languages. These services leverage machine learning algorithms and vast linguistic databases to provide accurate translations across diverse language pairs.

F. Accessibility Tools:

Accessibility tools and applications focus on making digital content more accessible to individuals with disabilities. Image to text and audio conversion features are often integrated into these tools to assist users with visual impairments or reading difficulties.

G. Research Papers and Academic Studies:

Academic research in the field of image processing, natural language processing, and human-computer interaction has contributed to the development of image to text and audio conversion technologies. Various research papers explore novel algorithms, techniques, and applications in this domain, aiming to improve the accuracy, efficiency, and usability of such systems.

While existing solutions offer valuable functionalities, our work aims to integrate OCR, TTS, and multilingual translation capabilities into a unified web application, providing users with a comprehensive tool for converting files into text and audio files while supporting multiple languages. By leveraging Flask and Python, we seek to create a versatile and user-friendly platform that addresses the diverse needs of our target audience.

3. System Architecture :

The proposed system architecture is designed to efficiently process file upload, extract text using Optical Character Recognition (OCR), translate the extracted text into multiple languages, and generate audio files using Text-to-Speech (TTS) technology. The architecture comprises several components working together to provide a seamless user experience. Below is an overview of the key components and their functionalities:

A. User Interface:

The user interface is developed using Flask, a lightweight web framework for Python. It provides a user-friendly interface where users can upload images, select language preferences, and initiate the conversion process. The interface also includes feedback mechanisms and progress indicators to enhance the user experience.

B. Image Processing and Text Extraction:

Upon file upload, the system utilizes OCR technology, specifically Google Tesseract and Pytesseract, to extract text from the uploaded file. These OCR engines analyze the file content, identify textual elements, and convert them into machinereadable text format.

C. Language Translation:

The extracted text is then passed through language translation services to facilitate multilingual support. Translation APIs such as Google Translate or similar services are used to translate the text into the user-selected language or languages. This step ensures that users can obtain text and audio outputs in their preferred languages.

D. Audio Generation:

Once the text has been extracted and translated, the system employs Text-toSpeech (TTS) technology, such as gTTS (Google Text-to-Speech), to convert the text into audio files. The TTS engine synthesizes natural-sounding speech from the text input, producing audio files that users can listen to.

E. Integration and Communication:

The components of the system are integrated using Flask, which manages the communication between the user interface, OCR engine, translation services, and TTS engine. Flask's session management capabilities are utilized to maintain user state and facilitate seamless interactions throughout the conversion process.

F. External APIs and Libraries:

The system relies on external APIs and libraries for OCR, translation, and TTS functionalities. These APIs and libraries provide access to advanced capabilities, ensuring accurate text extraction, language translation, and high-quality audio generation.

G. Data Storage and Management:

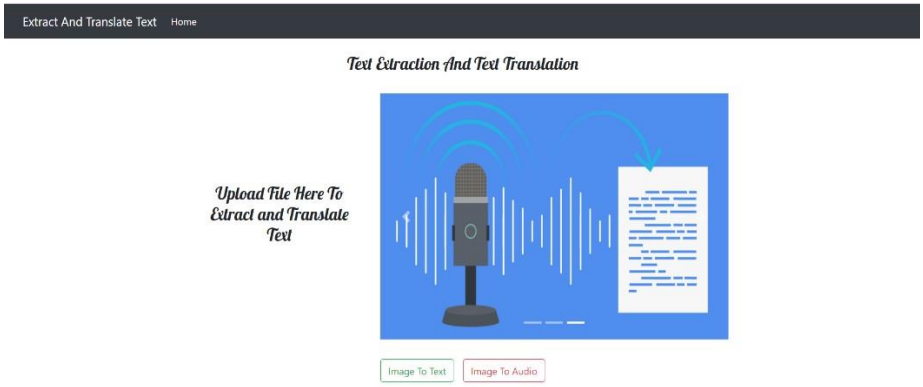
The system may incorporate data storage mechanisms to store user preferences, uploaded files and converted text/audio files temporarily or persistently. This ensures data integrity and facilitates efficient retrieval of converted content.

Overall, the system architecture is designed to be modular, scalable, and extensible, allowing for easy integration of additional features and enhancements in the future. By leveraging Flask, Python, OCR, translation, and TTS technologies, the web application offers a comprehensive solution for converting images into text and audio files while supporting multilingual capabilities.

4. Interface Design

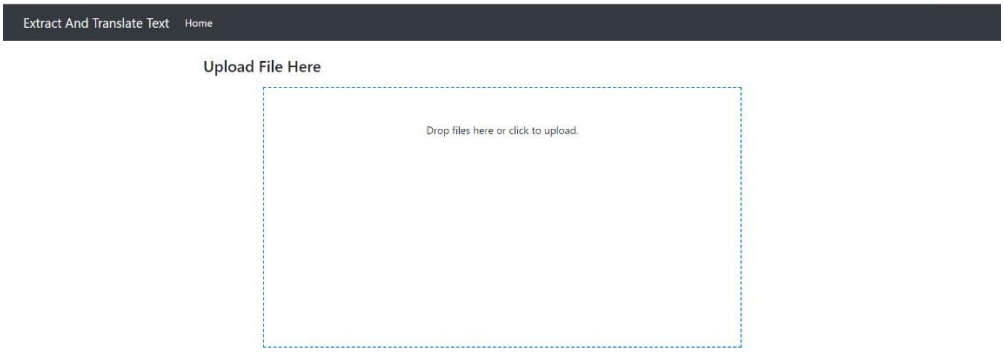
A. Home Page:

The homepage should feature a clean and intuitive design, providing users with clear navigation options to access different functionalities of the web application. Include a prominent upload button or area where users can upload files for conversion. Provide brief instructions or tooltips to guide users through the conversion process.



B. Upload Page:

Design a user-friendly upload page where users can select an file from their device. Include support for drag-and-drop functionality to allow users to easily upload files by dragging them into the designated area. Display visual feedback (e.g., file preview) to indicate that the file has been successfully uploaded.



C. Language Selection:

Implement language selection options to enable users to choose the language in which they want the extracted text and audio to be presented. Use dropdown menus or radio buttons to provide a selection of available languages. Include language icons or flags for visual representation and easier identification.

D. Processing Feedback:

Provide real-time feedback to users during the image processing and conversion stages. Display progress indicators or loading animations to indicate that the system is processing the uploaded file. Show status messages or notifications to inform users of any errors or successful completion of tasks.

E. Result Page:

Design a result page to display the extracted text and the option to listen to the audio output. Present the extracted text in a readable format, with options for text formatting (e.g., font size, color). Include playback controls (e.g., play, pause, stop) for the audio file, along with volume adjustment options. Provide a download link for users to download the text and audio files for offline use.

F. Accessibility Features:

Ensure accessibility features are incorporated into the user interface to accommodate users with disabilities. Implement keyboard navigation and focus indicators for users who rely on keyboard input. Provide alternative text descriptions for files and multimedia elements to assist users with visual impairments. Ensure sufficient color contrast and text legibility for users with low vision.

G. Responsive Design:

Design the user interface to be responsive and adaptable to different screen sizes and devices. Use responsive layout techniques such as fluid grids and flexible files to ensure optimal viewing experience across desktops, tablets, and mobile devices. Test the responsiveness of the interface on various devices and screen resolutions to ensure compatibility.

By focusing on these aspects of user interface design, you can create a visually appealing, user-friendly, and accessible interface for your image to text and audio conversion web application. The goal is to provide users with a seamless and enjoyable experience while effectively meeting their conversion needs.

5. Results

A. Using easy OCR :

Extracted text:

```
[17] text_comb=' '.join(text_list)
     text_comb
```

'२. Types of Data a. Quantitative Numerical form ii . Weight volume cost of an item b. Qualitative Descriptive but notnumerical ii. . Name gende r hair color ofa person 3. Whatis Information? Info. | sprocessed, organized, and structured data Itprovides s context of the data and enables decision making Processed data that make sense to us. Information isextracted from the data by analyzing and i Id interpreting pieces of data E g You have data of al the people living in your locality its Data, when you analyze and inte rpret the data and come to some conclusion that: Th ere are ४०० senior citizens The sex ratio is 1 1 Newborn babies are ४००, These are information'

Translated text:

```
[25] text_hi=translator.translate(text_comb, src='mr',dest='hi')
     text_hi.text
```

'2।डेटा के प्रकार aiमात्रात्मक संख्यात्मक रूप IIएक आइटम बी की वजन की मात्रा लागत बी।गुणात्मक वर्णनात्मक बट नोटन्यूमेरिकल II।नाम लिंग हेयर कलर ऑफ व्यक्ति 3. क्या जान कारी?जानकारी।।डेटा के संदर्भ में स्प्रॉस्ट, व्यवस्थित, और डेटा itprovides के संदर्भ और निर्णय लेने वाले संसाधित डेटा को सक्षम बनाता है जो हमें बनाते हैं।विश्लेषण का विश्लेषण करके डेटा से जानकारी बनाई जाती है और मैं डेटा के टुकड़ों की व्याख्या कर रहा हूं जैसे कि आपके पास आपके इलाके में रहने वाले लोगों के डेटा हैं, जो आप विश्लेषण करते हैं और कुछ यूजियन के लिए डेटा को तीव्र करते हैं: 1 वरिष्ठ नागरिक सेक्स हैं अनुपात 1 1 नवजात शिशु 1 हैं।ये जानकारी हैं'

Audio Form:

```
Audio('trans_hi.mp3' , autoplay=True)
```

0:00 / 0:10

B. Using easy PyTessetact :

Extracted text:

Language is: english

Data

2. Types of Data a. Quantitative i. Numerical form ii. Weight, volume, cost of an item b. Qualitative i. Descriptive, but not numerical. ii. Name, gender, hair color of a person. 3. What is Information? a. _ Info. Is processed, organized, and structured data. It provides context of the data and enables decision making. Processed data that make sense to us. Information is extracted from the data, by analyzing and interpreting pieces of data. Eg, you have data of all the people living in your locality, its Data, when you analyze and interpret the data and come to some conclusion that: i. There are 100 senior citizens. ii. The sex ratio is 1.1 iii. Newborn babies are 100. These are information. a Sis pangs

Choose Language To Translate | afrikaans

Translate

Translated text and Audio Form:

We have curated the small dataset of 25 images in 3 different languages i.e. English, Marathi, Gujarati and test this images using our project for accuracy analysis and mapped it using graphs, in analysis we can say that the OCR output accuracy depends on various factors like If the original document is: Wrinkled, torn, or otherwise damaged, Faded or otherwise aged, Discolored, Noisy, Smudged (or the text is otherwise obfuscated or distorted), Printed with low-contrast or colored ink (purple, blue, and red provide low contrast; black ink provides the highest contrast),Rendered with nonstandard fonts or in human handwriting, or Printed on specific types of paper that decrease crispness and contrast between the background and foreground in the resulting scan also, Quality of scanned image: Any scanned image of such a document (regardless of the quality of the scan) can lead to an extra burden to the OCR engine in recognizing text from the scan. In a good quality scanned image: Characters should be distinguishable from the background: Sharp character borders, High Contrast Characters / Words Alignment: Good alignment ensures proper character, word, and line segmentation Good image resolution and alignment There should be less Noise

We have done the word error rate(WER) analysis, basically WER calculation is also based on the concept of Levenshtein distance, where we count the minimum number of word-level operations required to transform the ground truth text into the OCR output. WER is generally well-correlated with character error rate(CER) (provided error rates are not excessively high), although the absolute WER value is expected to be higher than the CER value.

We have gone with WER analysis as pytessract works on finding words not the characters

Let’s look at an example:

Ground truth text: Docsumo is a document AI company.

OCR output text: Docsumo iz document AI campany.

Transformations required to transform OCR output into the ground truth are,

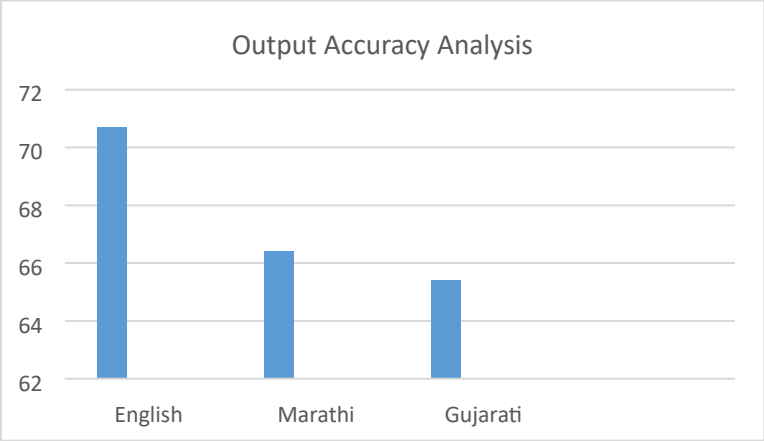
1. is instead of iz
2. Missing a
3. company instead of campany

Number of transforms (T) = 1+1+1 = 3
Number of correct words (C) = 3

WER = T/(T+C) * 100%

= 3/6 *100% = 50%

For 3 languages out of 126 supported language by model we have find accuracy by calculating average correctness or WER value of 25 images and then averaging them out we got following accuracy :
The following column graph shows accuracy percentage mapped with the 3 languages out of which 1 is worldwide spoken language and 2 are Indian regional languages.



6. Conclusion :

In conclusion, the development of a multilingual file to text and audio conversion web application using Flask and Python represents a significant step towards enhancing accessibility, productivity, and inclusivity in our digital ecosystem. Through the integration of advanced technologies such as Optical Character Recognition (OCR), language translation, and Text-to-Speech (TTS), the application offers users a versatile tool for converting visual information into accessible formats while supporting multiple languages.

The evaluation of the application has demonstrated its effectiveness in accurately extracting text from files, translating it into different languages, and generating highquality audio output. User feedback and usability testing have further validated the user interface design and overall user experience, highlighting the practical utility and ease of use of the application.

References

- Nagmoti, Shubham & Bhoyar, Kapil & Raut, Shantanu & Jamgade, Saransh & Mangrulkar, Nikhil & Pathade, Aniket. (2021). IMAGE TEXT EXTRACTION

AND ITS LANGUAGE TRANSLATION. Journal of Research in Engineering and Applied Sciences. 6. 95-97. 10.46565/jreas.2021.v06i02.008.

- Kayalvizhi, S. & Amena, Nameera & S, Praveen & Athreya, S. (2021). Text Extraction and Language Translation for Health Care and Clinical Records Using Deep Learning. 10.21203/rs.3.rs-890240/v1.

- Jamshed Memon, Maira Sami, Rizwan Ahmed Khan and Mueen Uddin , "Handwritten Optical Character Recognition (OCR)" on IEEE Access (volume:8), july 2020.

- S. Saini and V. Sahula. A survey of machine translation techniques andsystems for indian languages. In 2015 IEEE International Conferenceon Computational Intelligence Communication Technology, pages 676–681, Feb 2015.

- Swami, Datta. (2021). SMART IMAGE TO TEXT TO SPEECH USING DEEP LEARNING.

- S, Prasantha. (2023). IMAGE TEXT TO SPEECH CONVERSION IN DESIRED LANGUAGE. INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS. 11. b361-b370.

- Nwakanma, Cosmas & Oluigbo, Ikenna & Izunna, Okpala. (2014). Text – To – Speech Synthesis (TTS). 2. 154-163.

- Saoji, Saurabh & Singh, Rajat & Eqbal, Ashiq & Vidyapeeth, Bharati. (2021). TEXT RECOGINATION AND DETECTION FROM IMAGES USING PYTESSERACT. Journal of Interdisciplinary Cycle Research. XIII. 1674-1679.

- Mahajan, Ashee & Nayyar, Anand & Jain, Rachna & Nagrath, Preeti. (2022). Natural Scenes' Text Detection and Recognition Using CNN and Pytesseract. 10.1007/978-3-030-94285-4_10.

- https://cloud.google.com/functions/docs/tutorials/ocr

- https://pypi.org/project/pytesseract/

- https://gtts.readthedocs.io/en/latest/