# Implementation of a Document-to-Speech System

## Introduction

This document outlines the development of a system to convert documents containing text and images into speech for visually impaired individuals. It also highlights existing applications, their limitations, and how our project addresses these issues. Additionally, UML and sequence diagrams are provided for technical clarity.

## Existing Applications and Challenges

### Examples of Existing Applications

• Supersense

- Features: Reads documents, identifies objects, and scans barcodes.

- Issues: Limited naturalness in speech output and occasional OCR inaccuracies.

• KNFB Reader

- Features: Converts text to speech or Braille.

- Issues: High cost, limited multilingual support, and dependency on perfect lighting for OCR.

• Speechify

- Features: Converts digital/physical text into high-quality audio.

- Issues: Robotic-sounding voices and limited emotional expression in TTS.

• Murf AI

- Features: Advanced neural TTS with natural-sounding voices.

- Issues: Limited support for low-resource languages and complex accents.

### Challenges Faced by Existing Systems

• **Robotic Speech**: Lack of human-like intonation and emotion.

• **OCR Accuracy**: Struggles with handwritten or low-quality scanned documents.

• **Limited Multilingual Support**: Inadequate handling of regional languages or dialects.

• **Usability Issues:** Complex interfaces that are not user-friendly for visually impaired users.

• **Data Privacy Concerns**: Risk of sensitive data exposure during processing.

## Proposed Enhancements

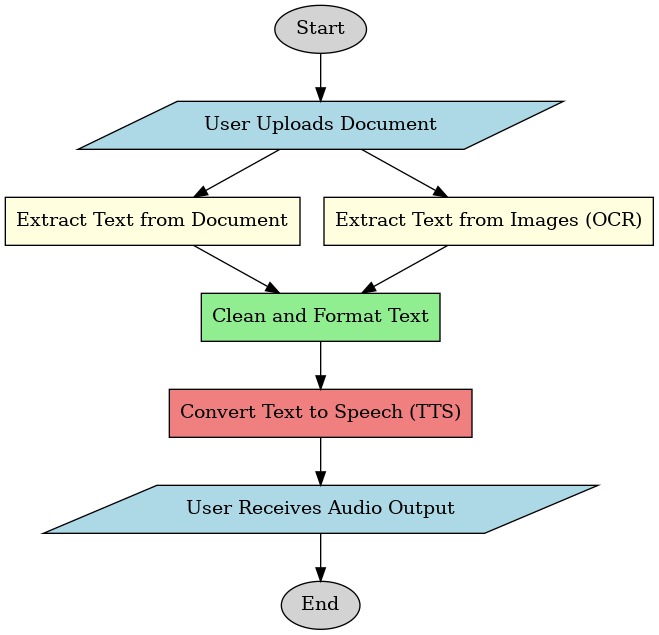
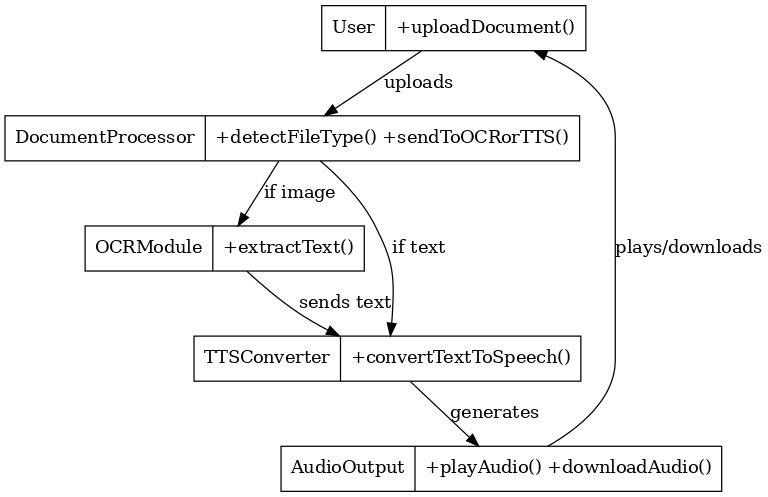
• Fine-tuning GPT-2 and hugging face text to speech speech5\_tts model to improve text processing accuracy and naturalness in speech synthesis.

• Enhancing OCR preprocessing to handle low-quality images and handwritten text effectively.

• Supporting multiple languages with region-specific models.

• Designing a simple, intuitive interface for ease of use.

• Ensuring robust data privacy through local processing and encryption.

## System Architecture

The system follows a structured workflow to process input documents and generate speech output. The workflow includes image preprocessing, OCR extraction, text refinement using GPT-2, and speech synthesis.

### Workflow

• Input document (image or PDF) is captured or uploaded.

• Preprocessing enhances image quality for OCR.

• OCR extracts text from the document.

• GPT-2 processes the extracted text for contextual accuracy.

• A fine tune TTS model converts the refined text into speech.

• Audio is delivered to the user.

## Technical Details

### 1. Image Preprocessing

Techniques: Grayscale conversion, noise reduction, binarization, and cropping.

### 2. OCR Module

Tool: Tesseract OCR (fine-tuned with custom datasets). Features: Handles diverse fonts, layouts, and handwritten text.

### 3. Text Processing

Model: Fine-tuned GPT-2 for contextual understanding and improved sentence structuring.

### 4. Speech Synthesis

Engine: Hugging face speech5 tts model for converting text to speech.

## Implementation Timeline

|  |  |  |
| --- | --- | --- |
| Milestone | Timeline | Deliverables |
| Project Planning | Feb 14 - Feb 21 | Requirements finalized |
| Data Preparation | Feb 22 - Mar 7 | Curated datasets for OCR & TTS |
| Model Development | Mar 8 - Mar 31 | Integrated OCR-TTS pipeline |
| GPT-2 Fine-Tuning | Apr 1 - Apr 15 | Enhanced accuracy in text processing |
| System Integration & UI | Apr 16 - Apr 25 | Complete system with UI |
| Testing & Validation | Apr 26 - May 1 | Final testing & performance evaluation |

**Work so far!**

We have divided the work into creating two models – image to text and then text to speech. When a document is loaded by the user in form of pdf or image, it can have text along with some images , euqations supporting those texts. Our app takes inspiration from a 2 pass SIC XE assembler, which in the first pass converts the image into text at the position where the image is encountered, and then in the second pass the entire text is loaded to be converted into speech.  
  
**Text to Speech Synthesis**

We have been working on converting text to speech for weeks now, however we got a clear direction after having a talk with you regarding what the project actually demands. At first we tried to read some research papers about the topic and also tried to implement it. After looking at the current landscape, we thought fine tuning a model would give better results. After trying our hands on a few models , we decided on the speech to text model (speech5\_tts model) from the hugging face. We have decided upon the dataset as of the moment, but we are open to changes. We have successfully completed the data preprocessing steps and would soon move on to the training stage.  
  


The first code snippet loads the dataset and maps the text and audio so that they can be preprocessed later.  
  
The second code snippet is the pre processing code where the data is cleaned. Data cleaning involves audio trimming, padding, text cleaning.