VSpeak

IMAGE TO TEXT TO SPEECH CONVERTER

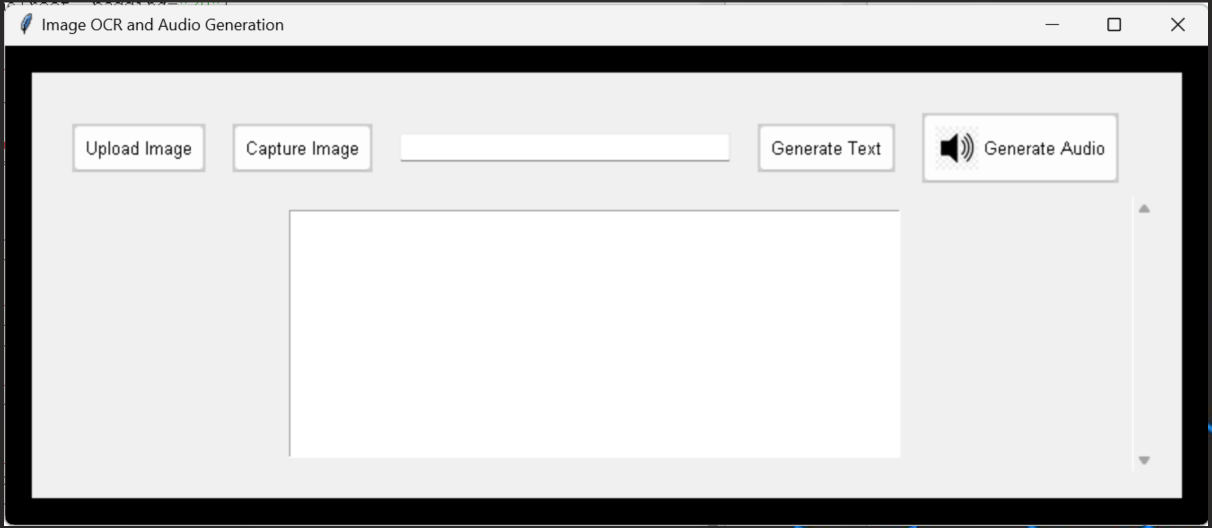
Sakshi Biyani (22BLC1385)
Vidushi Agnihotri (22BLC1387)

PROBLEM STATEMENT

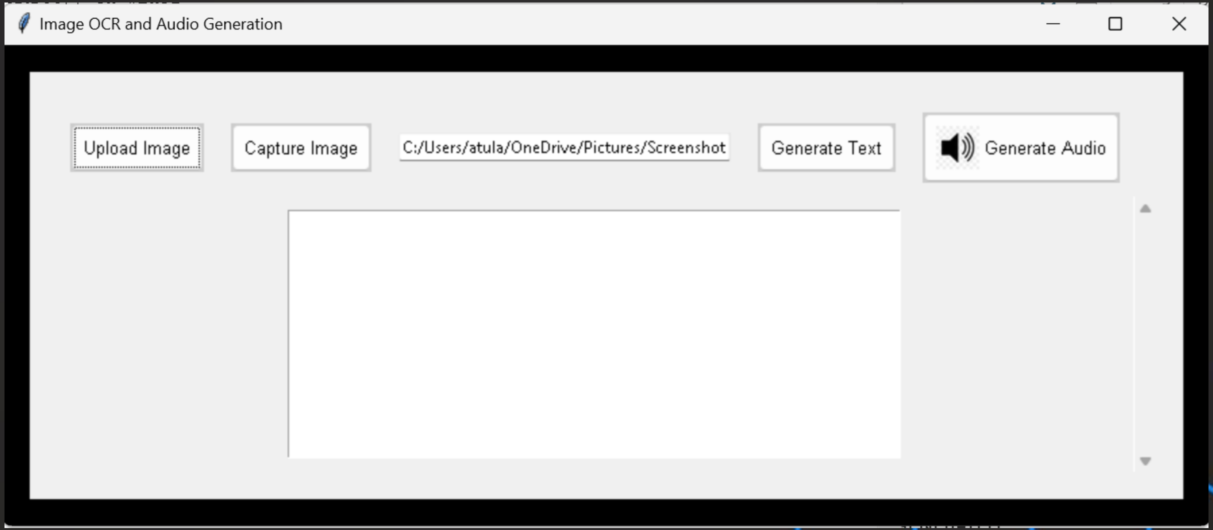
Create a system for accurately converting various types of images to machine-readable text. The solution should efficiently handle factors like image quality, text orientation, and language diversity.The goal is to produce accurate and contextually meaningful text output, with potential applications in digitizing documents, aiding visually impaired individuals, and enabling efficient text extraction and audio visualization for image analysis.

WEB APP UI/UX

SCREENSHOT OF GENERATED GUI:



SCREENSHOT AFTER THE IMAGE IS UPLOADED:



C:/Users/atula/OneDrive/Pictures/Screenshot

SCREENSHOT AFTER THE TEXT HAS BEEN GENERATED:

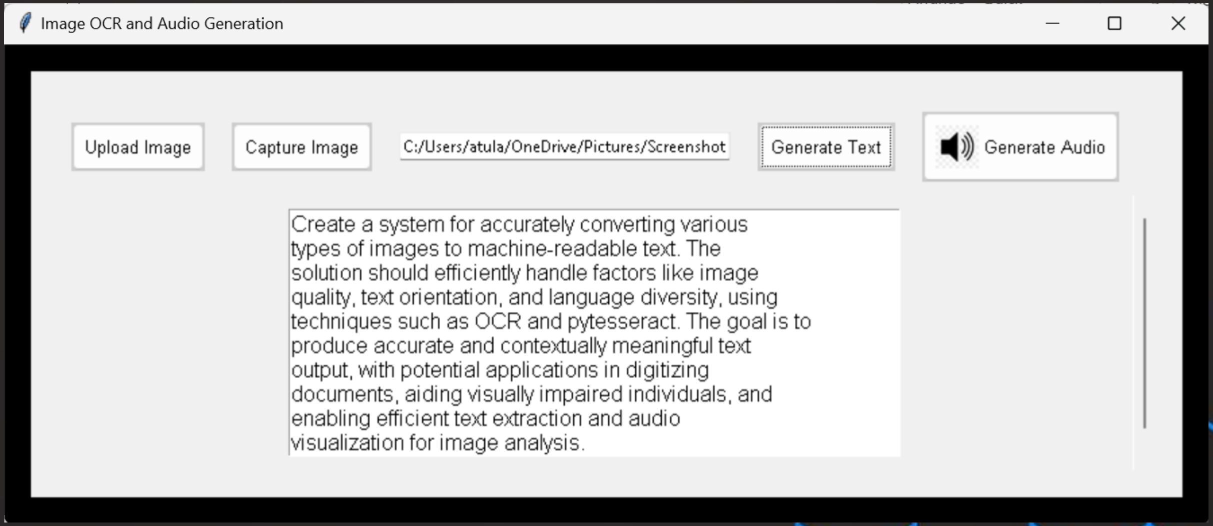


Image OCR and Audio Generation
Upload Image
Capture Image
C:/Users/atula/OneDrive/Pictures/Screenshot
Generate Text
Generate Audio

Create a system for accurately converting various types of images to machine-readable text. The solution should efficiently handle factors like image quality, text orientation, and language diversity, using techniques such as OCR and pytesseract. The goal is to produce accurate and contextually meaningful text output, with potential applications in digitizing documents, aiding visually impaired individuals, and enabling efficient text extraction and audio visualization for image analysis.

BASIC OVERVIEW OF OUR CODE FOR WEB APPLICATION

We have made a Python application built with the Tkinter library, enabling users to convert images to text and generate audio from the extracted text

Import Statements:
tkinter: Import the Tkinter library for building the GUI.
ttk, filedialog, messagebox: Additional Tkinter modules for specific functionalities like themed widgets, file dialogs, and message boxes.
PIL.Image, PIL.ImageEnhance, ImageTk: Imports from the Python Imaging Library (PIL) for image processing and displaying images in Tkinter.
pytesseract: Import the pytesseract library for Optical Character Recognition (OCR).
gtts: Import the gTTS library for text-to-speech conversion.
os, subprocess, cv2: Standard Python libraries for OS operations, subprocess management, and OpenCV for capturing images.

Tesseract Path Configuration:
Set the path to the Tesseract OCR executable.

Image Processing Functions:
enhance_image: Enhance image contrast.
image_to_text: Convert image to text using OCR.
Text-to-Speech Functions:
text_to_audio: Convert text to audio using gTTS.
play_audio: Play generated audio.

GUI Functions:
open_image: Open file dialog to select an image.
capture_image: Capture image from webcam.
generate_text: Generate text from the selected/captured image.
generate_audio: Generate audio from the extracted text.

Main Window Setup:
Create a Tkinter root window.
Set the window title and background color.
Configure a style for buttons.
Load and resize an audio icon.

Frame and Widgets:
Create a frame within the root window.
Add buttons for uploading/capturing images and generating text/audio.
Add an entry widget for displaying the image path.
Add a text box for displaying OCR results.
Add a scrollbar for the text box.
Run the GUI:
Start the Tkinter event loop to display the GUI and handle user interactions.

THANK YOU