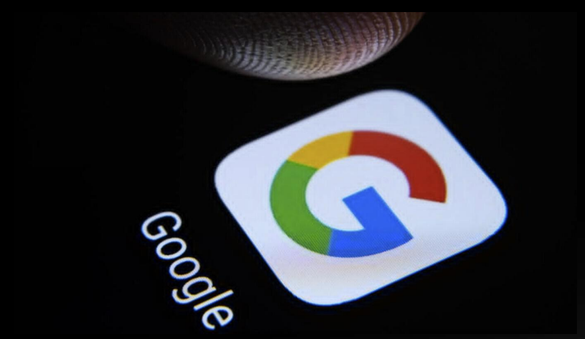


Echo Eyes

ABSTRACT

With the boom of online education following the COVID-19 pandemic, many virtual classes and course materials are now being shared via screen or as part of video content. While this mode of education is beneficial for many, it creates an accessibility gap for **visually impaired individuals**. Existing software, primarily designed to read text components, fall short in **reading text within images or video content**. Our mission is to bridge this gap, ensuring quality online education is accessible to all.

PROBLEM DESCRIPTION

In the wake of the online education surge, a large portion of content delivered in online classes is showcased through **screen-sharing or as embedded text within videos**. This format, while mainstream, inadvertently **sidelines visually impaired learners**. Current assistive technologies like JAWS and NVDA excel at reading direct text components, but falter when addressing **text presented within images or videos**. This discrepancy highlights a pressing need for improved accessibility tools tailored to the current online teaching methodologies.

PROJECT DESCRIPTION

Echo Eyes is a cutting-edge GUI (Graphical User Interface) tool designed to ensure that online education becomes universally inclusive, especially for **Visually Impaired Individuals**. Leveraging **OCR and text-to-speech technologies**, EchoEYES empowers users to go through their online materials in video/image format by defining the screen region, extract text, and then, have it read aloud. Future enhancements will include web-based hosting, customizable audio speed, Translation, and advanced image descriptions. Our mission with Echo Eyes is to bridge the accessibility gap in online education, making it truly inclusive for all learners.

OBJECTIVES

Identification of Limitations

Recognize and document the drawbacks of current tools for text extraction in screen-shared and video-based courses.

Design & Implementation

Create innovative software for detecting and extracting text from multimedia content.

Accessibility Transformation

Convert extracted text into an audible format for enhanced learning for visually impaired users.

Develop a User-Friendly Application

Develop an intuitive, user-centric software requiring minimal technical expertise, ensuring a seamless learning experience for visually impaired individuals.

Use Cases

1 Real-time Screen-sharing Lectures for Visually Impaired Students

Scenario

An online class is in session where the instructor utilizes screen sharing to present various materials including slides, diagrams, and real-time code demonstrations. Traditional assistive technologies can't decipher the content within the screen-shared visuals.

Use Case

As the lecture progresses, the GUI application continuously scans the content being screen-shared. Upon detecting textual information, it employs OCR to transmute this into readable text. The integrated TTS module then vocalizes the content, enabling visually impaired attendees to grasp the lecture content equivalently to their peers.

2

Accessing Pre-recorded Video Course Material

Scenario

A popular online course on ancient civilizations primarily employs videos for instruction. These videos often contain on-screen text annotations and graphics to enhance comprehension.

Use Case

Users can upload the video content to the GUI application. The software meticulously processes the video, analyzing each frame to extract textual information using advanced machine learning algorithms. All extracted text is then collated into an accessible format. As the video plays, the application vocalizes the extracted text synchronously, ensuring all crucial information is conveyed.

3 Interactive Webinars and Workshops

Scenario

Webinars and workshops, essential for continuous professional development, often feature real-time polling, Q&A segments, and interactive whiteboard sessions.

Use Case

Throughout the webinar, presenters frequently share visual components like graphs, real-time polls, and whiteboard annotations. The GUI application actively monitors these sessions, alerting users when new textual content surfaces on screen. After extracting this content, it's read aloud to attendees. When interactive segments like real-time polls are presented, the questions and options are vocalized, ensuring all participants can actively engage.

Our Solution and Approach

OCR-based Solution

Research & Optimization

Delve into identifying top-performing OCR engines and optimize them using diverse datasets from online lectures. This ensures the system's adaptability across varied content layouts.

Integration & Voice Output

Combine the OCR with a Text-to-Speech module, prioritizing natural voice output and catering to a range of language preferences.

Feedback Mechanism

Implement an intuitive feedback loop for users, enabling the reporting of inaccuracies and facilitating continuous improvement.

Implementation

1

Install Libraries

To begin the implementation, ensure that you have the necessary libraries installed. These include `tesseract`, which is used for optical character recognition (OCR), `pyttsx3` for text-to-speech conversion, `pyautogui` for screen capture, and `wxPython` for creating a graphical user interface if needed.

2

Define a Region Based on Cursor Position

Start by detecting the initial cursor position using `pyautogui.position()`. To focus on a specific area, define a region of interest (ROI) centered around the cursor. This ROI will be used for capturing the content of interest.

3

Perform OCR On Region

Capture a screenshot of the defined ROI using `pyautogui.screenshot()`. Crop the screenshot to isolate the ROI and convert it to grayscale (`"L"`) to enhance OCR accuracy. Utilize `pytesseract.image_to_string()` to perform OCR on the grayscale image, extracting textual information.

4

Extract Texts From The Source

After OCR, the extracted text will be in string format. You can further process this text as needed, such as removing unwanted characters or formatting.

5

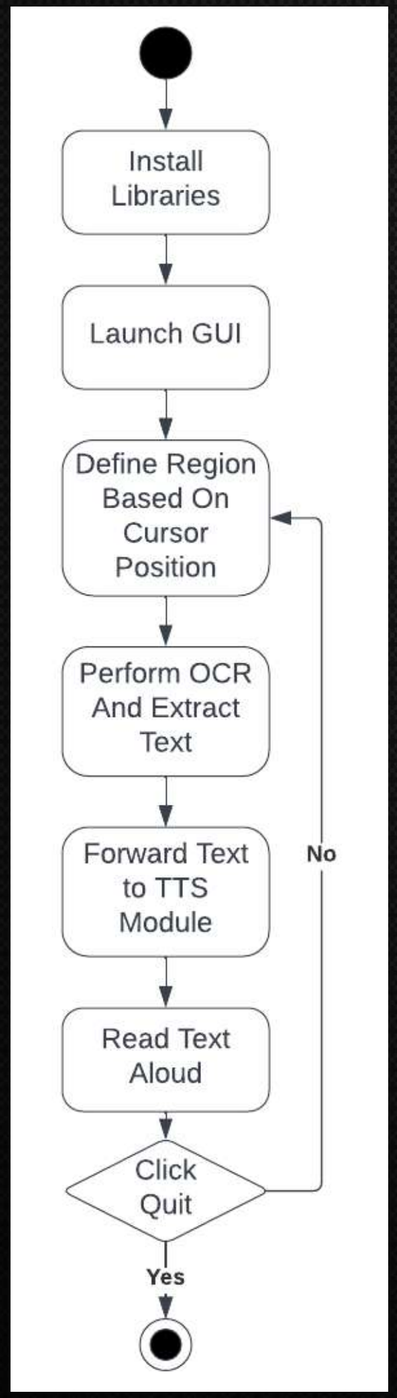
Forward The Text Output To Text To Speech

Initialize the `pyttsx3` library to set up a text-to-speech engine. Customize settings such as speech rate and voice options based on user preferences.

6

Read Text Aloud

Finally, use the configured text-to-speech engine to read the extracted text aloud. This step makes the content accessible to users with visual impairments. Ensure that you manage text updates and decide when to trigger the text-to-speech functionality, such as when changes in the OCR results are detected.



Implementation UML

Future Enhancements

1 User Feedback and Iterative Improvement

Implement a feedback system within the tool, allowing users to provide input on its effectiveness and usability. Use this feedback to make iterative improvements and address specific needs of partially blind users.

2 Enhanced Voice Commands

Continue to develop voice command functionality, enabling users to navigate the tool and interact with course content through natural language commands.

3 Gesture-Based Navigation (Future Enhancement)

We will be developing gesture-based navigation features for fully blind users, enabling them to interact intuitively with our accessibility tool and course content through touch gestures. This enhancement will provide immediate audio feedback, making online education more accessible and engaging for our users.

Thank You