International Journal of Research in Engineering and Science (IJRES)
ISSN (Online): 2320-9364, ISSN (Print): 2320-9356
www.ijres.org Volume 11 Issue 5 ǁ May 2023 ǁ PP. 768-774

Iris: Glasses for the visually impaired

[1]Shaikh Mohd Ashfaque, [2]Aartem Singh, [3]Mawrah Khan, [4]Laik Shaikh, [5]Yash Devadiga

*[*1]Department of Computer Engineering, Rizvi College of Engineering, Mumbai, Maharashtra, India*
[2]Department of Computer Engineering, Rizvi College of Engineering, Mumbai, Maharashtra, India
[3]Department of Computer Engineering, Rizvi College of Engineering, Mumbai, Maharashtra, India
[4]Department of Computer Engineering, Rizvi College of Engineering, Mumbai, Maharashtra, India
[5]Department of Computer Engineering, Rizvi College of Engineering, Mumbai, Maharashtra, India
Corresponding Author: mhdashfaque@eng.rizvi.edu.in

Abstract
As the world progresses in each industry, the difficulties for disabled people remain constant. People with disabilities, such as low or no vision, find it difficult to perform simple tasks like identifying objects in their surroundings, reading text documents, and recognizing people around them. Tactile tiles can make commercial spaces more accessible for the blind, but unfortunately, most places do not implement this. For blind individuals who might want to visit such locations, this poses a significant challenge. It becomes extremely difficult for them to lead an independent life, as they depend on family, friends, and caretakers to help them with their daily routines. To assist such individuals, Iris was created, a solution that makes living an independent life much easier in a cost-effective manner. The project revolves around image capturing and performing various operations on the images for text recognition, object detection, and face detection. People who are blind or visually impaired can scan printed text using optical character recognition (OCR) equipment provided by Iris, which can subsequently speak the text aloud. An ESP32 camera module is mounted on top of a pair of glasses. The camera module captures an image based on the gesture input provided by the user in the Iris mobile application developed using Flutter. The image goes through various image-cleaning processes, such as filtration, contrast, and autocorrect, using OpenCV2. For object detection, the input image from the ESP32 is sent to the server. On the server, using the cvlib library, the object is detected, and the output is sent to the mobile app. Iris is also capable of face detection; the input image from the ESP32 is sent to the server. On the server, encoding of the image is done; if the face is already registered in the user's database, it will return the name of the identified person. However, if the face is not registered, it will ask the user whether to add a new face or not. If yes, it will save that person's image along with the name recorded by the blind user. The final output from this process is read out to the user via headphones, Bluetooth speakers, and other audio sources as per the user's choice. The setup of the ESP32 module, image pre-processing, text extraction from an image, detecting objects and faces from the input image, Firebase database as well as the Django server setup, is achieved. Iris aspires to make reading, object detection, and face detection easier for the visually impaired. This paper further covers the methodology of Iris, a review of existing systems, the current implementation, and future work, as well as the results and conclusions based on the project.

Keywords: *ESP32 camera module, Object detection, Face detection, Optical Character Recognition (OCR)*

Date of Submission: 14-05-2023 Date of acceptance: 26-05-2023

I. INTRODUCTION

The world we live in is full of opportunities, but those opportunities can be inaccessible to people with disabilities. For people with low or no vision, everyday tasks such as recognizing objects, reading text documents and recognizing people can be difficult. This makes it difficult for them to live independently and often requires them to rely on the help of others. Fortunately, technology has the potential to change that. With advances in computer vision and machine learning, it is now possible to develop assistive technologies that can significantly improve the independence and productivity of the visually impaired. For this, we propose Iris, smart glasses that can detect objects using TensorFlow, perform text recognition using Tesseract and facial recognition using HOGward. The Iris project revolves around capturing images and performing various operations on the image for text recognition, object and face recognition. The ESP 32 module is installed on the glasses, which takes pictures based on the gesture input given by the user in the Iris mobile application. The captured image is transmitted to the server set up using Django. The final output of this process is read to the user via headphones, Bluetooth speakers or other audio sources of the user's choice. Using deep learning and computer vision algorithms, Iris can accurately detect objects in real time, recognize text from images, and recognize faces in

different settings. These features provide users with valuable information and improve their independence, security and productivity.

The Iris project is driven by an Android app built into Flutter that works with gestures and makes it easily accessible for the visually impaired. All outputs are voice activated so users can easily navigate and control the device independently. With a user-friendly interface, voice-based system and advanced features, Iris can change the way the visually impaired interact with the world. Iris aims to make life easier for the visually impaired, and as the technology progresses, it can change the lives of millions of visually impaired and facilitate their integration into society.

II. LITERATURE REVIEW

In the research paper titled **"Smart Glasses for Blind People" by Hawra Al Said, Lina Alkhatib, Aqeela Aloraidh, and Shoaa Alhaidar published in the Prince Mohammad Bin Fahd University journal in Spring 2018/2019**, the authors propose smart glasses to assist blind individuals in reading and translating text. The glasses utilize image scanning and convert the text into audio, which can be listened to through connected headphones. The project's technologies include OpenCV, Optical Character Recognition (OCR) with Tesseract, Efficient and Accurate Scene Text Detector (EAST), Text to Speech technology (TTS), and the Google Translation API. However, the project's drawbacks include limited language support (English only), a specific capture distance range, and heavy and impractical hardware. As our contribution, we introduce Iris as a solution to address these issues. Iris offers smaller and more user-friendly hardware, making it easier to carry. The computational tasks are offloaded to the server, reducing the reliance on bulky glasses. Iris also incorporates gesture-based interaction, providing additional features like object detection and facial recognition.

Another paper titled **"Smart Glasses for Blind" by Dr. Sankit Kassa, Manisha Inchanalkar, Amisha Kamble, and Mayuri Inchanalkar, published in the IJIRT (International Journal of Innovative Research in Technology) in May 2021**, presents a project designed to assist visually impaired individuals in reading text documents and detecting nearby objects. The system offers a minimal setup and allows users to listen to scanned text through headsets or speakers. The main tools used are Optical Character Recognition (OCR), Putty, and Text-to-Speech (TTS). However, this project faces drawbacks such as overheating issues with the Raspberry Pi 4 used for computing, a heavy camera attachment, and lower accuracy due to a non-HD camera. To address these limitations, Iris is proposed as a lightweight solution. Iris provides lightweight hardware attached to the glasses for capturing pictures, while the processing is done on the user's phone. This eliminates the need for extra gear and offers gesture-controlled features for an improved user experience.

III. METHODOLOGY

Each functionality of Iris follows a certain methodology:

2.1 OCR

Initially, the image is obtained from the ESP 32 camera module and uploaded to Firebase. Subsequently, the Iris mobile application sends an OCR request to the server, initiating the image processing stage using OpenCV library in Python. Various image processing techniques are employed to enhance the accuracy of OCR results, including Gaussian blur, thresholding, contour detection, and morphological operations. The pytesseract library is utilized for text extraction. The resulting text is then transmitted back to the Iris app, where it is converted into audio using a text-to-speech library.

To elaborate on the algorithm, the following steps are involved: First, the image address is obtained from the request form, and the image is opened and converted into a NumPy array. Skew correction is performed to align the image, and the corrected angle is saved. The image is then converted to grayscale and undergoes Gaussian blur and thresholding. A rectangular kernel is applied for dilation, followed by Canny edge detection. Contours are detected in the edge image and sorted based on their x-coordinate. For each contour, a bounding rectangle is determined, and if it satisfies specific conditions, the image is cropped accordingly. The contrast of the cropped image is enhanced through normalization, followed by conversion to grayscale. Erosion and dilation operations are applied to the grayscale image using a rectangular kernel. Adaptive thresholding is performed, and morphological opening is carried out using a defined kernel. Finally, text is extracted from the processed image using the pytesseract library, and the resulting text is returned.

These detailed steps elucidate the procedure employed by the Iris system for analyzing images and extracting text, facilitating its utilization in practical applications.

2.2 Object Detection

The methodology for detecting common objects in an image involves the use of Python and libraries such as OpenCV, Flask, and NumPy. The process begins by creating a Flask app and defining a route to handle client POST requests. The image URL is retrieved from the form data using the 'request.form.get' function. Subsequently, the image is opened using the urllib request and converted into a NumPy array with the 'asarray' function. The NumPy array is then decoded using OpenCV's 'imdecode' function and stored as 'myImg'.

A function named 'objDetect' is defined to handle the detection process, taking the image URL as input. Within this function, OpenCV's 'detect_common_objects' function is used to identify common objects in the image. Only the labels of the detected objects are returned from the function and stored in the 'objects' variable. Finally, the 'objects' variable is returned as a JSON response to the client. This methodology provides a straightforward approach for detecting common objects in images and offers a valuable tool for image processing and analysis.

2.3 Face Detection

The image is captured from the ESP 32 camera module and uploaded to Firebase. The Iris mobile application sends a facial recognition request to the server, which reads the image using OpenCV. The Python face-recognition library is utilized to detect faces in the image. If the identified person is registered in the blind user database, their audio URL is returned to the Iris application, where it is played using the Flutter audio player library. However, if the detected person is not registered, the Iris application receives a response indicating "New face found." The blind user can decide whether to register the person by swiping up on the application. If the user chooses to register, they are prompted to say the person's name, and a long press triggers sound recording. After recording, the sound is played back to the user. The server is queried again with the final data comprising the image and recorded sound, which are stored in the Firebase database.

The algorithm described in section 3.3.3 focuses on face registration and detection. For face registration, the Firebase app is checked and initialized if necessary. The algorithm retrieves the image address and name from the request form and proceeds to download existing encodings and name lists. The new image is loaded, converted to RGB, and encoded. The algorithm appends the encoding and the new name to the known encodings and name list, respectively. The updated encodings and name list are saved and uploaded to the storage. A message is returned to indicate the successful addition of the new face.

Regarding face detection, the algorithm retrieves the image address and loads the existing encodings and name list. The new image is loaded, converted to RGB, and encoded. The algorithm compares the new image encoding with the known encodings and calculates face distances. If a match is found, the algorithm returns the name of the best matching face. Otherwise, it returns a message indicating the detection of a new face.

2.4 ESP32 and Mobile Application interface

The project utilizes a mobile application developed using Flutter, which enables blind users to interact with the system through swiping gestures. By swiping on the mobile screen, requests are sent to the ESP32 microcontroller board to capture images using the integrated camera module. The ESP32 then uploads the captured images to Firebase and sends the corresponding image URL back to the mobile app. This seamless communication between the mobile app and ESP32 enables efficient image transfer for further processing.

Once the image URL is received on the mobile app, it is forwarded to a Flask server for processing. The Flask server performs the necessary image processing tasks, such as facial recognition or object detection, depending on the specific request. The processed results are then sent back to the mobile app, where they are presented as voice output. This voice output provides blind users with real-time information and feedback based on the processed image data. By utilizing the swiping gestures on the mobile app, blind users can conveniently request image capture, receive processed results, and access the system's functionalities in an accessible and user-friendly manner.

IV. RESULTS

We have the ESP32 camera with WIFI module.

Figure 1: Glasses mounted with ESP32 Camera

The Wi-Fi ssid and password can be uploaded to the camera by connecting to the ESP32's access point. The camera is coded to click pictures as soon as it is requested to by the mobile application. These pictures are saved on Firebase cloud storage by the ESP32. The ESP32 then returns the firebase URL to the mobile application which, in turn, forwards it to the server for processing it accordingly.

This is the Mobile application that works using swipes and voice assist to help the visually impaired:

Figure 2: Iris Mobile Application

Following outputs have been achieved:

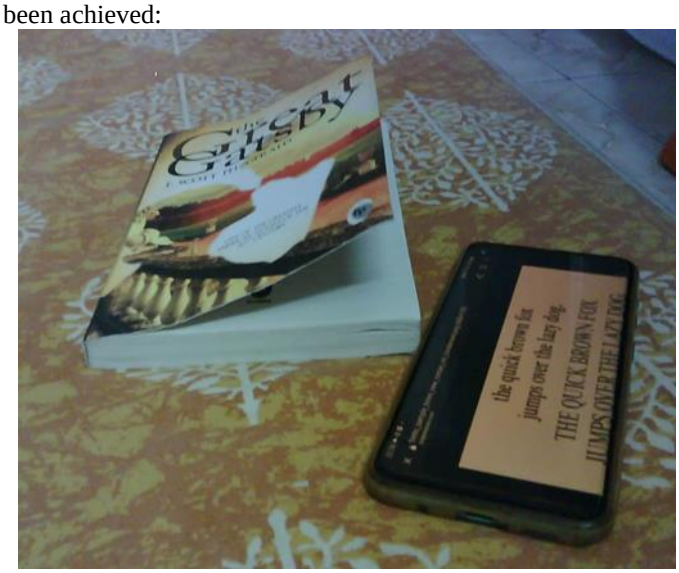

Figure 3: Image taken bu ESP32 Cam for Object Detection

```
I/flutter ( 2124): running object detection!
I/flutter ( 2124): [cell phone, book]
D/TTS     ( 2124): Invalid pitch 0.4 value -
Range is from 0.5 to 2.0
I/flutter ( 2124): join cell phone, book
D/TTS     ( 2124): Utterance ID has started:
dc7d55f0-5714-4fc1-a912-10b85fcf6f03
D/TTS     ( 2124): Utterance ID has complete
d: dc7d55f0-5714-4fc1-a912-10b85fcf6f03
I/flutter ( 2124): speaking completed
```

Figure 4: Output of Object Detection

Figure 5: Image taken by ESP32 Cam for OCR

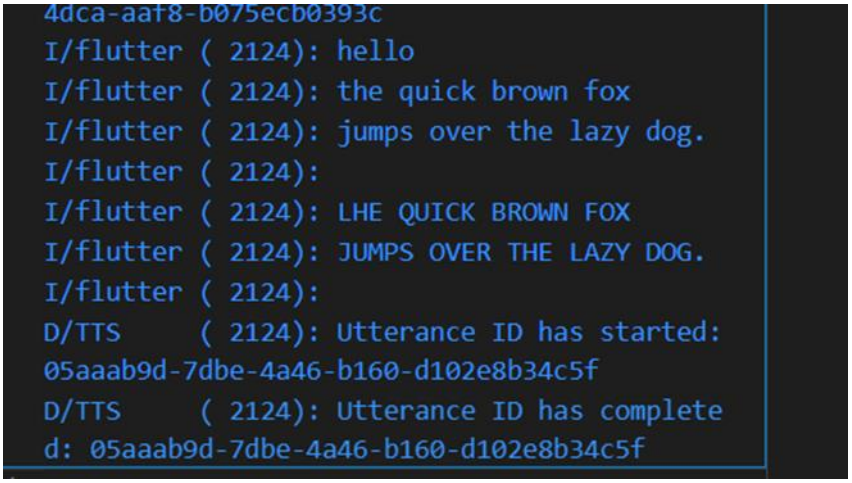

Figure 6: Output of OCR

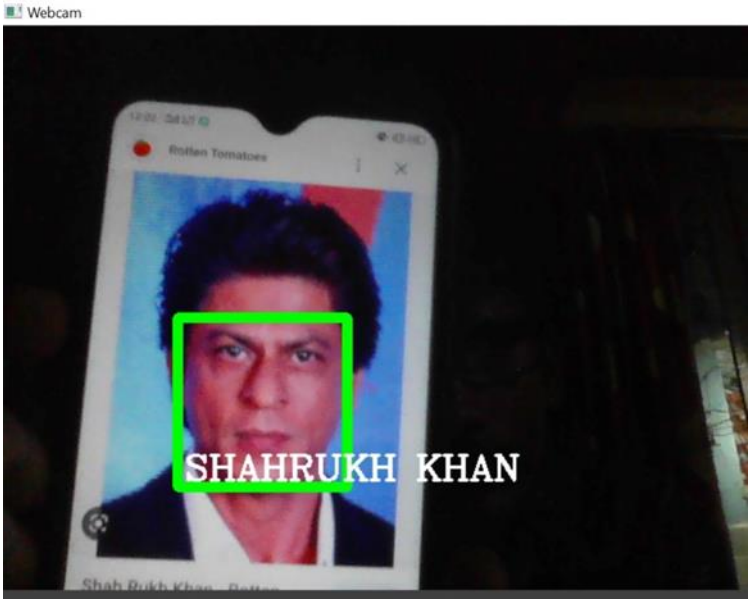

Figure 7: Output of Face Detection

V. CONCLUSION

In conclusion, the Iris smart glasses offer an innovative solution for visually impaired individuals, providing cutting-edge features controlled by an Android application built in Flutter that works on gestures. With its voice-based output system, Iris allows users to easily navigate its advanced capabilities, including object detection, OCR, and face detection powered by TensorFlow, Tesseract, and HOGward technologies.

By leveraging deep learning and computer vision algorithms, Iris can accurately detect and identify objects, recognize text from images and videos, and identify faces in real-time, all while being easily controlled using gestures through the Android application.

Overall, Iris smart glasses represent a significant breakthrough in assistive technology, offering visually impaired individuals an unprecedented level of autonomy and accessibility. With its user-friendly interface, voice-based output system, and advanced capabilities, Iris has the potential to revolutionize the way visually impaired individuals interact with the world, providing them with greater independence, safety, and productivity. As technology continues to evolve, Iris and other similar products have the potential to continue to transform the lives of visually impaired individuals and provide them with equal opportunities in society.

REFERENCES

[1]. Al Said, Hawra, Lina Alkhatib, Aqeela Aloraidh, and Shoaa Alhaidar. "Smart Glasses for Blind People." Smart Glasses for Blind People. Accessed October 23, 2022. https://www.pmu.edu.sa/attachments/academics/ppt/udp/cces/smart-glasses-for-blind-people-report.pdf

[2]. Kassa, Dr. Sankit, Manisha Inchallankar, Amisha Kamble, and Mayuri Inchallankar. "Smart Glasses for Blind People - PDF Free Download." Smart Glasses for Blind People - PDF Free Download. Accessed October 23, 2022. https://docplayer.net/221564200-Smart-glasses-for-blind-people.html

[3]. Challenges blind people face when living life. "Challenges Blind People Face When Living Life," October 20, 2022. https://www.letsenvision.com/blog/challenges-blind-people-face-when-living-life

[4]. Ishfar, Shakleen. "Document Detection in Python. Building a Simple Document Detector… | by Shakleen Ishfar | Intelligentmachines | Medium." Medium, June 6, 2020. https://medium.com/intelligentmachines/document-detection-in-python-2f9ffd26bf65