SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I9P114

Volume 10 Issue 9, 154-159, September 2023

Original Article

IoT Based Smart Glasses with Facial Recognition for People with Visual Impairments

Swapna Choudhary[1], Nitin Dhote[2], Ashwini A. Deshpande[3], Ansh Sambhariya[4], Poorvi K. Joshi[5]

[1]*Electronics and Telecommunication Engineering, G. H. Raisoni College of Engineering, Nagpur, Maharashtra, India.*
[2]*Electrical Engineering Department, St. Vincent Palloti College of Engineering, Nagpur, Maharashtra, India.*
[3] *Electronics and Telecommunication Engineering Department, Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India.*
[4,5]*Electronics Engineering Department, Shri Ramdeobaba College of Engineering and Management, Nagpur, Maharashtra, India.*

[5]*Corresponding Author : joshipk@rknec.edu*

Received: 18 July 2023 Revised: 22 August 2023 Accepted: 15 September 2023 Published: 30 September 2023

***Abstract** - The paper presents a low-cost intelligent smart glasses system that uses a Raspberry Pi 4 board, a camera module, and an ultrasonic sensor to capture images and audio from the environment and process them using computer vision and natural language processing techniques. A working prototype of intelligent eyewear will enable people with vision impairments to recognize others in their sight and learn about potential dangers. This will be made possible by face recognition technology and distance detection capabilities. A built-in sensor that emits ultrasonic waves in the direction the user is moving while scanning a maximum of 5 to 6 meters away at an angle of 30 degrees. This new system could resolve some of the most significant issues that blind people still face. Finally, a message is sent to blind people, informing them of the person in front of them using the sounds associated with each individual in the database. The proposed system consists of a depth camera to gather depth data from the environment, an ultrasonic rangefinder made up of an ultrasonic sensor, and an embedded CPU board serving as the central processing module. The embedded CPU handles tasks like depth image processing, data fusion, AR rendering, and guiding sound synthesis. AR glasses are used to show the visual enhancement information, and an earphone is used to play the guiding sound. This proposed system is designed to assist people with visual impairments by enabling them to recognize people in front of them and detect obstacles. The technology utilized in this system includes facial recognition and distance detection capabilities, which are made possible by using Raspberry Pi and Pi cameras. The system also incorporates an ultrasonic sensor and a Raspberry Pi powered by a 5V power supply to detect and avoid obstacles.*

***Keywords** - Smart glasses, Raspberry Pi 4, Facial recognition, Ultrasonic sensor, Vision impairments, Computer vision, Natural language processing.*

1. Introduction

Individuals with vision impairment face various challenges in their daily lives, including difficulties with low vision that hinder their full integration into society. Such individuals often feel embarrassed when they cannot perform or complete tasks due to their vision impairment.

As people are intelligent, technology is crucial in bridging the gap between society and those with vision impairment. While new technologies are constantly being introduced to make life more convenient and comfortable for healthy individuals and those with disabilities, many modern assistive devices fall short of consumer expectations regarding cost and level of aid provided. To address this issue, this project proposes a new style of smart glasses designed to assist people who are blind in performing various daily tasks. These glasses utilize wearable technology, including a camera mounted on the spectacles worn by the user, to perform necessary recognition tasks. A dataset of people from common scenarios is constructed to enable facial recognition. This project aims to increase opportunities for individuals with vision loss to realize their full potential by implementing a real-time face recognition system and obstacle sensing using an ultrasonic sensor and incorporating these smart glasses for blind people.

2. Literature Survey

"Smart Glasses" aid blind and visually impaired people by identifying objects through a headset attached to the glasses. The goal is to create an intelligent system that recognizes nearby objects and sends actions to the brain. “NODE MCU” mimics the human brain by analyzing images. Voice messages are sent to the earphones to assist in finding things quickly. The device uses sensors to identify objects and emit ultrasonic waves to scan up to 5-6 meters at a 30° angle. This technology could help to resolve pressing issues faced by the visually impaired, such as identifying individuals through associated sounds in a database [1].

To advance object recognition, a compiled dataset of 328k images of everyday scenes with 91 object categories. Per-instance segmentations were used for accurate object localization. The dataset was created with the help of crowd workers using user interfaces for category recognition, instance spotting, and instance segmentation. A thorough statistical analysis and present baseline performance using a Deformable Parts Model for segmentation detection and bounding box results, surpassing PASCAL, ImageNet, and SUN [2].

The initiative aims to create an autonomous thinking workshop for the visually impaired using “picture to sound” conversion. MATLAB processes the image displayed to the user, then compared to a microcontroller database for auditory output via earbuds. Colour data is evaluated to determine the object’s colour, and the user is notified of the production through headphones [3].

3. Proposed System

The proposed system comprises an ultrasonic rangefinder composed of an ultrasonic sensor and a Raspberry PI 4 embedded CPU board functioning as the primary processing module. The Raspberry PI 4 performs depth image processing, data fusion, AR rendering, and guided sound synthesis tasks. Additionally, the system includes a pair of glasses to display visual enhancement information and an earphone to play the guiding sound.

4. Block Diagram

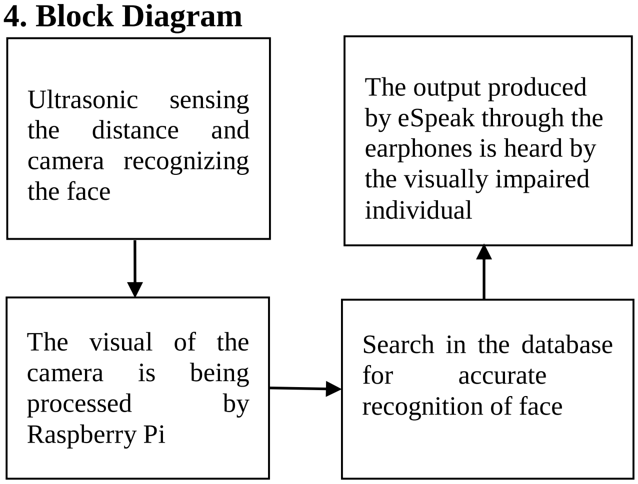

Fig. 1 Block diagram

5. Block Diagram Implementation

An ultrasonic sensor is a device that detects things and calculates their distance using high-frequency sound waves. Robotics, automation and industrial applications that need non-contact measurement frequently use these sensors.

The HC-SR04 sensor is the most popular kind of ultrasonic sensor. An ultrasonic signal is sent using the TRIG pin, and the signal returned by the item is picked up by the ECHO pin [4]. The following components are needed to integrate an ultrasonic sensor with a Raspberry Pi 4. a Raspberry Pi 4, an HC-SR04 ultrasonic sensor, and the computer language Python [5, 6].

The following steps describe incorporating an ultrasonic sensor with a Raspberry Pi 4: Step 1: Join the HC-SR04 sensor’s VCC and GND pins to the Raspberry Pi 4’s 5V and GND pins. Make the connections with jumper wires and a breadboard. Step 2: Join the Raspberry Pi 4’s GPIO 23 to the TRIG pin of the HC-SR04 sensor. Step 3: Join the Raspberry Pi 4’s GPIO 24 to the ECHO pin of the HC-SR04 sensor. Step 4: Launch the Raspberry Pi 4’s Python IDE and write a new Python script. Step 5: Import the time and RPi.

GPIO libraries that are required for the ultrasonic sensor. Step 6: Using the GPIO setup function, define the GPIO pins for the ultrasonic sensor’s TRIG and ECHO pins. Step 7: Create a process that delivers an ultrasonic signal and measures the time for the call to bounce back from the item. Step 8: Use the equation distance = sound speed x time / 2 to determine how far away the target is. Step 9: Use a while loop to calculate the distance to the item continuously. Step 10: Execute the Python application and observe the distance measurements on the Raspberry Pi 4.

This sensor is helpful for various applications because it allows you to estimate the distance to an object without touching it. A tiny, high-quality camera specifically made for the Raspberry Pi 4 is the Raspberry Pi 4 Camera Module. It enables Raspberry Pi 4 to take pictures and movies, making it an excellent tool for applications like time-lapse photography, security systems and more.

The Camera Serial Interface (CSI) bus, a high-speed serial bus that delivers visual data from the camera to the Raspberry Pi 4, links the camera module to the Pi 4. The camera module contains a 5-megapixel sensor that can record video at 1080p at 30 frames per second and still photographs with a 2592 x 1944 pixels resolution.

The following parts are needed to integrate the Raspberry Pi 4 Camera Module with the Raspberry Pi 4: Raspberry Pi 4, the Camera Module for the Raspberry Pi 4, a ribbon cable for a suitable camera, an appropriate power supply for the Raspberry Pi 4, a monitor or display to see the output from the camera, and the Raspberry Pi 4 can be controlled by a keyboard and mouse. To integrate the Raspberry Pi 4 camera module with the Raspberry Pi 4, these steps are required: Step 1: Turn off the Raspberry Pi 4 and unplug it from the power source. Step 2: Connect the camera ribbon wire to the Raspberry Pi 4's Camera Serial Interface (CSI) port. Ensure the cable is securely attached. Step 3: Attach the camera module for the Raspberry Pi 4 to the opposite end of the ribbon cable.

Take extra care not to harm the fragile camera module. Step 4: Turn on the Raspberry Pi 4 and log into the operating system as the fourth step. Step 5: Open the terminal and run the following command to enable the camera module: sudo raspi-configuration. Step 6: Enter after choosing "Interfacing Settings" with the arrow keys. At step 7, select "Camera" and hit Enter. Step 8: Click "Yes" to turn on the camera, then hit Enter. Step 9: Reboot the Raspberry Pi 4 by typing the following command: sudo reboot. Step 10: Open the terminal again and run the following command to take a still photo: raspistill -o image.jpg Step 11: Open the file manager to locate the saved image file. Step 12: To capture video, run the following command: raspivid -o video.h264 -t 10000. This command will capture 10 seconds of video and save it to the "video.h264" file.

Text can be converted into spoken words using the free and open-source voice synthesizer software called eSpeak. The eSpeak library can be used to add text-to-speech functionality to a variety of apps, making them more accessible to people who have visual impairments or prefer to listen to content rather than read it. This paper uses eSpeak, which has various customizable features, to convert text to speech. ESpeak can change the language, voice, pitch, and other speaking techniques [7, 8].

OpenCV (Open Source Computer Vision) is a C++, Python, and other programming language-based software library for computer vision and machine learning. It is intended to assist programmers in developing real-time computer vision programs that can examine still photos and moving pictures for various purposes, including image segmentation, object detection and facial recognition. Image processing, video processing, machine learning methods [9, 10], pattern recognition, and 3D reconstruction are a few of OpenCV's standout characteristics.

Additionally, it has a broad range of capabilities that make it appropriate for use in a variety of applications, such as robots, autonomous cars, medical imaging, security and surveillance systems, augmented reality, and more [11-14]. Python's NumPy library is used to perform numerical computations. This open-source project provides a high-performance multidimensional array object and tools for using these arrays. Scientific computing, data analysis, and machine learning heavily use NumPy. One of the main advantages of using NumPy is that it offers a quick and easy approach to handling big data arrays. When working with enormous datasets, it's critical to have fast, memory-efficient arrays like those provided by NumPy. Moreover, a large selection of mathematical operators and functions are available in NumPy that can be used to manipulate arrays. Another major feature of NumPy is its ability to integrate with other libraries and languages.

For instance, scientists and engineers who use Python and MATLAB will find it handy since NumPy arrays are easily translated to and from MATLAB arrays. The Fortran libraries, frequently used in scientific computing, can also be accessed through NumPy [5]. The well-known open-source Dlib C++ library offers machine-learning tools and algorithms for computer vision and image processing tasks. Dlib's capability to do facial recognition and landmark detection is one of its primary advantages. It has a trained face detection system that can accurately identify faces in pictures and videos, even in poor lighting and when there are occlusions. After faces are found, Dlib may perform landmark detection to find important facial features, including the corners of the mouth and eyes, the tip of the nose, and the chin.

Tasks like face alignment, face recognition, and emotion detection can be done using this information. To identify faces in pictures and videos, the Face Recognition library employs a deep learning model trained on a sizable dataset of faces. The library can also perform face detection and alignment for accurate face detection and recognition, even in poor illumination and with occlusions. Face Recognition offers various face-related tasks besides face recognition, including face encoding, face clustering, and face identification.

Face encoding entails taking characteristics from a face image and comparing them to see if other faces share them. Face identification identifies a person from their face, whereas face clustering includes grouping similar faces. Moreover, Face Recognition offers a straightforward API for working with other libraries and programs. Together with prominent deep learning frameworks like TensorFlow and PyTorch, it supports widely used image and video formats.

The simplicity of Face Recognition is one of its main benefits. Its interface is straightforward and user-friendly, making it easier for programmers to incorporate facial recognition into their apps. A variety of beneficial examples and tutorials that show how to use the library for typical facial recognition tasks are also included in the library [15-20]. A built-in Python library called the time offers several functions for handling time-related activities. Because it is a standard library, Python installations automatically include it and do not need to be set up separately. Retrieving the current time and date is one of the time library's key purposes. The time () method returns the current time as the number of seconds since the Unix epoch (January 1, 1970) can compute and compare things throughout time using this number. Moreover, the time library provides utilities for converting between several time formats. The local time () function returns the current time in a more human-readable format, with separate fields for the year, month, day, hour, minute, and second. To format a time string in a certain format, such as “YYYY-MM-DD HH:MM: SS,” use the strftime () method.

The time library’s capacity to gauge the length of code execution is another crucial aspect. The time () method can be used before and after a block of code, and the elapsed time can be determined by comparing the two times. This can be useful for benchmarking code and discovering performance bottlenecks. The time library also includes sleep and delay routines for code execution. The sleep () function can suspend a program’s execution for a predetermined period. This can help imitate real-time behaviour or provide delays between operations. The time library also has tools for working with calendars and dates. While the calendar module offers functions for working with dates and calendars, such as producing a calendar for a given month and year, the mktime () function can convert a time tuple to a Unix timestamp.

6. Methodology

The device’s technology and operation use facial recognition and distance measurement to help visually impaired people identify people they already know. The Raspberry Pi 4 camera and ultrasonic sensors run the gadget. The ultrasonic sensors determine the distance between the object and the person, while the camera takes a picture of the person in front of the thing.

This information is then processed by the Raspberry Pi 4 using particular codes written into it [6, 21]. These codes are intended to recognize features and provide precise distance measurements. After determining the face and distance, the device examines a database of people’s faces that it has already stored. If the person’s face is stored in the database, the device will recite the person’s name aloud into the visually impaired person’s earphones using text-to-speech technology. This method can benefit blind people with trouble identifying familiar faces in public settings like shopping centres or the street. Thanks to the device, they no longer need to depend on other visual cues to know who is in front of them. Face detection and Text-To-Speech (TTS) synthesis using a Raspberry Pi (RPi) device are among the various tools and technologies. A secure protocol called RPi SSH (Secure Shell) enables users to view and manage their RPi remotely through a terminal or command-line interface. [8, 22, 23]. The RPi is remotely controlled, and Python scripts are run on the RPi using RPi SSH in this experiment. Python is a well-liked computer language for various tasks, including data analysis and machine learning. Python is used in this project to create facial detection and TTS synthesis scripts. [13, 24, 25] Espeak is a Linux-based TTS synthesizer that turns text into voice. In this paper, speech output for the visually impaired individual is produced using Python scripts and the espeak module.

[7] Machine learning libraries like OpenCV, NumPy, and dlib are used to perform facial recognition. The open-source computer vision and machine learning software library, OpenCV (Open Source Computer Vision), offers tools for picture processing and object detection. [11-13, 26] The image data is subjected to numerical processes using the Python library for numerical computing known as NumPy [27, 28].

Dlib is a cutting-edge C++ toolkit with facial detection and recognition machine learning algorithms. The facial recognition library for Python is used to find faces in captured images and compare them to images that have already been saved in the database [14]. The code will run the espeak module and summon the person’s name using the syntax “espeak. synth ()” if the faces match. Through a speaker or earphones attached to the RPi device, the visually impaired person can hear this, produce speech output and call out the person’s name [7]. In conclusion, the project implements facial recognition and text-to-speech synthesis, enabling the visually impaired person to recognize people they are familiar with, using a combination of technologies including RPi SSH, Python, machine learning libraries like OpenCV, NumPy, and dlib, and the espeak module.

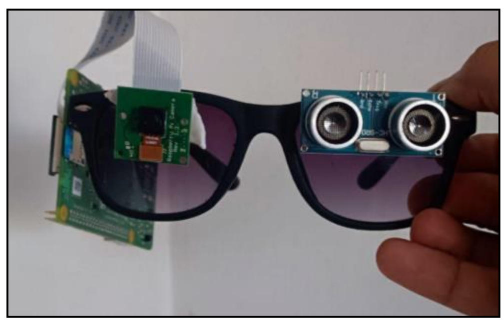

Fig. 2 Hardware image front view

Fig. 3 Hardware image side view

7. Results

Numerous samples were created and evaluated, and one of them, as illustrated in Figure 1, demonstrates the process of obtaining audio from the Raspberry Pi and the results of experimenting with the proposed smart glasses Figure 1, validating the fundamental idea behind the design. Utilizing the Raspberry Pi 4 camera module, implemented an intelligent glass specifically designed for individuals with visual impairments. This innovative project introduces an affordable and accessible solution using a low-cost single-board computer, the Raspberry Pi 4, and its camera. The smart glasses showcased in this paper feature face recognition and distance measurement capabilities, with the potential to expand their functionality by incorporating additional models into the core program. However, the size of the Raspberry Pi's SD card may limit the number of tasks that can be added. Each model corresponds to a specific mode or task, enabling users to run their desired way independently.

This revolutionary concept aims to enhance the quality of life for visually impaired students, regardless of their financial situations. The following steps will focus on assessing the device's user-friendliness and optimizing its power management. A plan to expand the project's scope by integrating the details of unknown individuals from government databases, allowing blind individuals to identify unknown individuals in their surroundings.

8. Conclusion

The low-cost intelligent smart glasses system we have described in this paper can help persons with vision problems by allowing them to recognize people in their line of sight and become aware of potential threats. Using computer vision and natural language processing methods, the system collects and processes images and audio from the surroundings using a Raspberry Pi 4 board, a camera module, an ultrasonic sensor, and a pair of AR glasses. The system can carry out facial recognition and distance detection duties while giving the user feedback via sounds and improved visuals. A functioning prototype and some early testing have shown the viability and efficiency of our technology. Our technology differs from previous systems in a number of ways, including low cost, high portability, simple usability, and real-time performance.

The battery life, field of view, and precision of our system are only a few of its drawbacks. We want to make our system better in the future by fixing these problems and including more capabilities like object detection, scene description, and navigation advice. We come to the conclusion that our approach can enable people with vision impairments live better lives and become more independent and socially included.

People with visual impairments could experience a considerable improvement in their quality of life thanks to the proposed smart glasses system. It is an innovative new technology that is still in the works but has the potential to completely change how those who have vision problems interact with their surroundings. The suggested system could be improved even further by adding more sensors, such as a GPS sensor to aid in navigation or a contact sensor to alert the user of impending obstructions. As technology advances, it is likely that smart glasses systems like the one suggested in this paper will become more accessible and affordable to people with visual impairments. It could also be integrated with other assistive technologies, such as a screen reader or a speech synthesizer. Their independence and quality of life will both benefit from this.

References

[1] V. Bindhu et al., "International Conference on Communication, Computing and Electronics Systems," *Proceedings of ICCCES 2020*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[2] Tsung-Yi-Lin et al., "Microsoft COCO Common Objects in Context," *European Conference on Computer Vision*, pp. 740-755, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shrilekha Bangar, Preetam Narkhede, and Rajashree Paranjape, "Vocal Vision for Visually Impaired," *The International Journal of Engineering and Science*, vol. 2, no. 3, pp. 1-7, 2013. [Google Scholar] [Publisher Link]

[4] Alessio Carullo, and Marco Parvis, "An Ultrasonic Sensor for Distance Measurement in Automotive Applications," *IEEE Sensors Journal*, vol. 1, no. 2, pp.143-147, 2001. [Google Scholar] [Publisher Link]

[5] Akshansh Sharma et al., "Python: The Programming Language of Future," *International Journal of Innovative Research in Technology*, vol. 6, no. 12, pp. 115-118, 2020. [Google Scholar] [Publisher Link]

[6] Eben Upton, and Gareth Halfacree, *The Raspberry Pi Camera Module*, Raspberry Pi® User Guide, 4, Wiley Online Library, 2016. [CrossRef] [Publisher Link]

[7] N.A. Azis et al., "Evaluation of Text-to-Speech Synthesizer for Indonesian Language Using Semantically Unpredictable Sentences Test: IndoTTS, eSpeak, and Google Translate TTS," *2011 International Conference on Advanced Computer Science and Information Systems*, Jakarta, Indonesia, pp. 237-242, 2011. [Google Scholar] [Publisher Link]

[8] N. Ifeanyi, O. Ikenna, and O. Izunna, "Text-To-Speech Synthesis (TTS), *International Journal of Research in Information Technology*, vol. 2, no. 5, pp. 154-163, 2014. [Google Scholar]

[9] Joyassree Sen et al., "Face Recognition Using Deep Convolutional Network and One-Shot Learning," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 4, pp. 23-29, 2020. [CrossRef] [Publisher Link]

[10] I. Indriyani et al., "Using Convolutional Neural Networks to Detect Races in Human Faces Using MATLAB Programming," *International Journal of Engineering Trends and Technology*, vol. 71, no. 6, pp. 66-74, 2023. [CrossRef] [Publisher Link]

[11] Reinhard Klette, *Concise Computer Vision, An Introduction into Theory and Algorithms*, Springer, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] Bernd Jähne, and Horst Haußecker, *Computer Vision and Applications*: *A Guide for Students and Practitioners*, Academic Press, 2000. [Google Scholar] [Publisher Link]

[13] M. Naveenkumar et al., "OpenCV for Computer Vision Applications," *Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15)*, pp. 52-56, 2015. [Google Scholar] [Publisher Link]

[14] Earle Kit M. Sanchez et al., "Three Triangle Method for Face Recognition Using Dlib and OpenCV," *2020 IEEE 12th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, pp. 1-5, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[15] G. Madhan Kumar et al., "Facial Recognition Smart Glasses for Visually Challenged Persons," *International Journal of Engineering Science and Computing*, 2020. [Google Scholar]

[16] Naveen Tiwari et al., "Face Recognition Smart Glasses for Visually Challenged Persons," *International Journal of Research and Development in Applied Science and Engineering (IJRDASE)*, vol. 21, no. 1, 2021. [Google Scholar] [Publisher Link]

[17] Narayan T. Deshpande, and S. Ravishankar, "Face Detection and Recognition Using Viola-Jones Algorithm and Fusion of PCA and ANN," *Advances in Computational Sciences and Technology*, vol. 10, no. 5, pp. 1173- 1189, 2017. [Publisher Link]

[18] Kavia, and Manjeet Kaur, "A Survey Paper for Face Recognition Technologies," *International Journal of Scientific and Research Publications*, vol. 6, no. 7, pp. 441-445, 2016. [Publisher Link]

[19] A.D. Chandan et al., "Survey Paper on Vehicle Security Using Facial Recognition & Password," *SSRG International Journal of Electronics and Communication Engineering*, vol. 9, no. 6, pp. 5-9, 2022. [CrossRef] [Publisher Link]

[20] Hirak Dipak Ghael, L. Solanki, and Gaurav Sahu, "A Review Paper on Raspberry Pi and its Applications," *International Journal of Advances in Engineering and Management (IJAEM)*, vol. 2, no. 12, pp. 225-227, 2020. [Publisher Link]

[21] Ankita Bhuniya et al., "Smart Glass for Blind People," *AMSE Journals*, vol. 38, no. 1, pp. 102-110, 2017. [Google Scholar] [Publisher Link]

[22] Sanjay Rajaraman, and Ps Shivapriya, "A-Eye: Computer Vision and Deep Learning Based Smart-Glasses with Edge Computing for Visually Impaired," *2023 3rd International Conference on Artificial Intelligence and Signal Processing (AISP)*, pp. 1-5, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Peer Azmat Shah, H. Ali Rashed Ali Alhefeiti, and R. Hussain Mohamed Ali Alrayssi, "Machine Learning-Based Smart Assistance System for the Visually Impaired," *2023 9th International Conference on Information Technology Trends (ITT)*, pp. 139-144, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mukhriddin Mukhiddinov, and Jinsoo Cho, "Smart Glass System Using Deep Learning for the Blind and Visually Impaired," *Electronics 2021*, vol. 10, no. 22, pp. 1-30, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] K. Shruthi et al., "A Study on Face, Expression, Object and Character Recognition to Implement An Aid for the Visually Impaired," *Proceedings of the International Conference on IoT Based Control Networks & Intelligent Systems (ICICNIS)*, pp. 1-7, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] A. Pentland, and T. Choudhury, "Face Recognition for Smart Environments," *Computer*, vol. 33, no. 2, pp. 50-55, 2000. [CrossRef] [Google Scholar] [Publisher Link]

[27] A. Hafeez Ali et al., "A Google Glass Based Real-Time Scene Analysis for the Visually Impaired," *IEEE Access*, vol. 9, pp. 166351-166369, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[28] Stefan van der Walt, S. Chris Colbert, and Gael Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22-30, 2011. [CrossRef] [Google Scholar] [Publisher Link]