IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 14, 2021, accepted December 9, 2021, date of publication December 13, 2021,
date of current version December 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3135024

A Google Glass Based Real-Time Scene Analysis for the Visually Impaired

HAFEEZ ALI A., SANJEEV U. RAO, SWAROOP RANGANATH, T. S. ASHWIN, (Member, IEEE), AND GUDDETI RAM MOHANA REDDY, (Senior Member, IEEE)
Department of Information Technology, National Institute of Technology Karnataka Surathkal, Mangalore 575025, India

Corresponding author: T. S. Ashwin (ashwindixit9@gmail.com)

ABSTRACT Blind and Visually Impaired People (BVIP) are likely to experience difficulties with tasks that involve scene recognition. Wearable technology has played a significant role in researching and evaluating systems developed for and with the BVIP community. This paper presents a system based on Google Glass designed to assist BVIP with scene recognition tasks, thereby using it as a visual assistant. The camera embedded in the smart glasses is used to capture the image of the surroundings, which is analyzed using the Custom Vision Application Programming Interface (Vision API) from Azure Cognitive Services by Microsoft. The output of the Vision API is converted to speech, which is heard by the BVIP user wearing the Google Glass. A dataset of 5000 newly annotated images is created to improve the performance of the scene description task in Indian scenarios. The Vision API is trained and tested on this dataset, increasing the mean Average Precision (mAP) score from 63% to 84%, with an IoU > 0.5. The overall response time of the proposed application was measured to be less than 1 second, thereby providing accurate results in real-time. A Likert scale analysis was performed with the help of the BVIP teachers and students at the ‘‘Roman & Catherine Lobo School for the Visually Impaired’’ at Mangalore, Karnataka, India. From their response, it can be concluded that the application helps the BVIP better recognize their surrounding environment in real-time, proving the device effective as a potential assistant for the BVIP.

INDEX TERMS Google glass, human–computer interaction, azure cognitive services, microsoft vision API, ubiquitous computing, visual assistant.

I. INTRODUCTION

According to the World Health Organization, it is estimated that there are at least 2.2 billion people globally who have vision impairment or blindness.[1] Out of these, around 45 million are blind and in need of vocational and social support. This population faces many difficulties in perceiving and understanding their surroundings since more than 80% of the information entering the brain is visual [1]. Studies have shown that vision accounts for two-thirds of the activity in the brain when a person's eyes are open [2]. The loss of sight represents a public health, social and economic issue in developing countries, where 9 out of 10 of the world's blind live. It is estimated that more than 60% of the world's blind reside in India, sub-Saharan Africa, and China. In terms of regional differences, the prevalence of vision impairment in low and middle-income regions is estimated to be four times higher than in high-income regions [3]. The loss of sight causes much suffering to the affected individuals and their families. Despite many efforts, population growth and aging are expected to increase the risk of more people acquiring vision impairment.

A visually impaired person deals with orientation and navigation issues daily. These issues can be alleviated with the help of particular types of equipment that can provide additional support to the individuals. With the improvements in computer vision and human-computer interaction techniques, it is possible to assist Blind and Visually Impaired People (BVIP) with scene recognition tasks. With the motivation of helping the BVIP community, this paper presents an application implemented on Google Glass[2] that acts as a visual assistant to the BVIP.

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

[1] https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment

[2] https://www.google.com/glass/start/

Many efforts by several researchers have been made to design systems that aid the BVIP. Bradley *et al.* [4] experimented, testing whether a group of sighted individuals and visually impaired individuals experience a difference in physical and mental demands when given directions to specific landmarks. Battaglia *et al.* [5] developed an integrated, modular, and expandable open-source package called Blind Assistant to show that it is possible to produce effective and affordable aids for the BVIP. Meza-de-Luna *et al.* [6] designed a social-aware assistant using a pair of smart glasses and a haptic belt to enhance the face-to-face conversations of the BVIP by providing them with vibrational cues from the belt. Chang *et al.* [7], [8], [9] proposed a wearable smart-glasses-based drug pill recognition system using deep-learning for the BVIP to enable them to improve their medication use safety. The system consists of a pair of wearable smart glasses, an artificial intelligence (AI)-based drug pill recognition box, and a mobile phone app. The smart glasses are used to capture images of the drugs to be consumed, and the AI-based drug recognition box is used to identify the drugs in the image. The mobile app is used to track drug consumption and also to provide timely reminders to the user. Zientara *et al.* [10] proposed a shopping assistant system for the BVIP called the 'Third Eye' that aids in navigation and identification of various products inside a shop. Similarly, Pintado *et al.* [11] designed a wearable object detection device in eyewear that helps to recognize items from the produce section of a grocery store.

In addition to shopping assistants, researchers have also developed Electronic Travel Aids (ETA) and obstacle detection systems to assist navigation. Quinones *et al.* [12] performed a needs-finding study to assist in navigation of familiar and unfamiliar routes taken daily among the BVIP. They concluded that a device that can act as an assistant is needed for better navigation. El-taher *et al.* [13] have done a comprehensive review of research directly in, or relevant to, outdoor assistive navigation for the BVIP. They also provided an overview of commercial and non-commercial navigation applications targeted at assisting the BVIP. Lee *et al.* [14] implemented a guidance system that uses map-matching algorithms and ultrasonic sensors to guide users to their chosen destination. Tapu *et al.* [15] implemented an autonomous navigation system for the BVIP based on computer vision algorithms. Similarly, Vyavahare *et al.* [16] used a combination of ultrasonic sensors and computer vision techniques to build a wearable assistant that can perform obstacle detection and image description. Laubhan *et al.* [17] and Trent *et al.* [18] designed a wearable Electronic Travel Aid for the blind, which uses an array of ultrasonic sensors to survey the scene. Bai *et al.* [19] proposed a depth image and multi-sensor-based algorithm to solve the problem of transparent and small obstacle avoidance. Their system uses three primary audible cues to guide completely blind users to move safely and efficiently. Nguyen *et al.* [20] developed a way-finding system on a mobile robot helping the BVIP user in an indoor setting. Avila *et al.* [21] developed a smartphone application that helps in localization within an indoor setting. In this system, 20 Bluetooth beacons were placed inside an indoor environment. When a BVIP user holding the smartphone moves through the building, the user will receive auditory information about the nearest point of interest. A very similar system was developed by Bie *et al.* [22] for an outdoor setting. Finally, Guerreiro *et al.* [23] developed a smartphone based virtual-navigation application that helps the BVIP gain route knowledge and familiarize themselves with their surroundings before visiting a particular location. Lupu *et al.* [24] presented an experimental framework to assess the brain cortex activation and affective reactions of the BVIP to stimuli provided by a sensory substitution device used for navigation in real-world scenarios. The test was done in 5 different types of experimental scenarios. It was focused on evaluating working memory load, visual cortex activation, and emotional experience when visually impaired people perceive audio, haptic, and multimodal stimuli. Chang *et al.* [25] proposed a wearable assistive system comprising a pair of smart glasses, a waist-mounted intelligent device, and an intelligent cane to help BVIP consumers safely use zebra crossings. They used artificial intelligence (AI) based edge computing techniques to help the BVIP users to utilize the zebra crossings.

Other researchers have focused on the design of assistive systems which help in scene description and analysis. Ye *et al.* [26] analyzed how different devices can help the BVIP in their daily lives and concluded that smartphones play a significant role in their daily activities. Pēgeot *et al.* [27] proposed a scene text tracking system used for finding and tracking text regions in video frames captured by a wearable camera. Gonzāles-Delgado *et al.* [28] proposed a smart gloves system that helps in meeting some of the daily needs of the BVIP, such as face recognition, automatic mail reading, automatic detection of objects, among other functions. Memo *et al.* [29] developed a head-mounted gesture recognition system. Their system uses a depth camera and an SVM classifier to identify the different gestures during a human conversation. Barney *et al.* [30] developed a sensory glass system that detects obstacles and informs the user of 3D sound waves. The glasses were fitted with five ultrasonic sensors placed on the left, upper-left, front, right, and upper-right parts. Shishir *et al.* [31] designed an Android app that can capture images and analyze them for image and text recognition. B. Jiang *et al.* [32] designed a wearable assistance system based on binocular sensors for the BVIP. The binocular vision sensors were used to capture images in a fixed frequency, and the informative images were chosen based on stereo image quality assessment (SIQA). Then the informative images were sent to the cloud for further computations. Bogdan *et al.* [33] proposed a system composed of a pair of smart glasses with an integrated microphone and camera, a smartphone connected with the smart glasses through a host application, and a server that serves the purpose of a computational unit. Their system was capable of detecting obstacles in the nearest surrounding, providing an estimation of the size of an object, face recognition, automatic text recognition, and question answering of a particular input image. Pei *et al.* [34] proposed a visual image aid for vocalizing the information of objects near the user.

Some researchers have designed their smart glasses to develop applications that assist visually impaired people. Chang *et al.* [35] and Chen *et al.* [36] proposed an assistive system comprising wearable smart-glasses, an intelligent walking stick, a mobile device app, and a cloud-based information management platform used to achieve the goals of aerial obstacle avoidance and fall detection goals for the BVIP. The intelligent walking stick provides feedback to the user with the help of vibrations to warn the user of obstacles. Furthermore, when the user experiences a fall event, an urgent notification is immediately sent to family members or caregivers. In the realm of wearable intelligent glasses, Chang *et al.* [37] and Chen *et al.* [38] have also proposed a drowsiness-fatigue-detection system to increase road safety. The system consists of wearable smart-glasses, an in-vehicle infotainment telematics platform, an onboard diagnostics-II-based automotive diagnostic bridge, a rear light alert mechanism in an active vehicle, and a cloud-based management platform. The system is used to detect drowsiness and fatigue in a driver in real-time. When detected, the active vehicle real light alert mechanism will automatically be flickered to alert following vehicles, and warning messages will be played to alert the driver.

Although many systems have been proposed and developed to assist the visually impaired, their practical usability is very limited due to the application's wearability and portability. In this era of high-end consumer electronics, where multiple sensors are embedded in light, highly portable smart glasses such as the Google Glass, it is possible to design an application that addresses the usability concerns faced by previous applications while also providing real-time responses to complex problems such as scene recognition and object detection. Therefore, in this paper, a Google Glass based real-time visual assistant is proposed for the BVIP. The rest of the paper is organized as follows. Section II describes related work done by other researchers on Google Glass to solve real-world social problems. In Section III, the proposed application is presented, along with explaining the different design choices. Here, the merits of the proposed application are explained in detail. The various steps involved in using the application are also provided. In Section IV, the results of the proposed work and the feedback obtained by the BVIP users are presented. Finally, the conclusion is given in Section V.

II. RELATED WORK

Google Glass is a brand of smart glasses with a prism projector for display, a bone conduction transducer, a microphone, accelerometer, gyroscope, magnetometer, ambient light sensor, proximity sensor, a touchpad, and a camera. It can connect to other devices using a Bluetooth connection, a micro USB, or a Wi-Fi connection. Application development for the device can be done using the Android development platform and toolkit available for mobile devices running Android OS.

Since its release, researchers have used the device to design systems to solve many real-life problems. Jiang *et al.* [39] proposed a Google Glass application that is used for food nutrition information retrieval and visualization. On similar grounds, Li *et al.* [40] developed a Google Glass application that can be used to assess the uniqueness and aesthetics of a food dish by analyzing its image for visual appeal, color combinations, and appearance. A few of the researchers have used the device in the medical field to treat children with Autism Spectrum Disorder (ASD). For instance, Washington *et al.* [41], [42] developed a Google Glass-based system for automatic facial expression recognition, delivering real-time social cues to children with ASD, thus improving their social behavior.

Lv *et al.* [43] developed a touch-less interactive augmented reality game using Google Glass. Wang *et al.* [44] presented a navigation strategy for NAO humanoid robots via hand gestures based on global and local live videos displayed on Google Glass. Similarly, Wen *et al.* [45] developed a Google Glass-based system to achieve hands-free remote control of humanoid robots. Xu *et al.* [46] used the device to facilitate intelligent substation inspection by using virtual video and real-time data demonstration. Widmer *et al.* [47] developed a medical information search system on Google Glass by connecting it to a content-based medical image retrieval system. The device takes a photo and sends it along with keywords associated with the image to a medical image retrieval system to retrieve similar cases, thus helping the user make an informed decision.

Devices such as Microsoft Kinect and Google Glass have also been used to help visually impaired people. For instance, Lausegger *et al.* [48] developed a Google Glass application to help people with color vision deficiency or color blindness. Anam *et al.* [49] developed a dyadic conversation aid using Google Glass for the visually impaired. Hwang *et al.* [50] implemented an augmented vision system on Glass, which overlays edge information over the wearer's real-world view, to provide contrast-improved central vision to the user. They used a combination of positive and negative laplacian filters for edge enhancement. Neto *et al.* [51] proposed a wearable face recognition system to aid the visually impaired in real-time. Their system uses a Kinect sensor to acquire an RGB-D image and run an efficient face recognition algorithm. Similarly, Takizawa *et al.* [52] proposed Kinect cane - an assistive system for the visually impaired based on the concept of object recognition.

Kim *et al.* [53] performed a systematic review of the applications of smart glasses in various applied sciences, such as healthcare, social science, education, service, industry, and computer science. Their study shows a remarkable increase in the number of published papers on the application of smart glasses since the release of Google Glass. Further, they claimed that the research has been steadily increasing as of 2021. With this, it can be concluded that Google Glass has been extensively used for designing applications to solve problems in various fields. Inspired by its potential, this paper presents a Google Glass-based application to solve some of the problems faced by the BVIP community by developing a scene descriptor using the Custom Vision API provided by Azure Cognitive Services.[3] The merits and features of Google Glass that led to its use in the proposed application and the system design are further explained in the next section.

III. SYSTEM DESIGN

Google Glass is relatively lightweight at 36g, which is quite comfortable to wear and use for extended periods. It comes with a head strap to firmly secure the device while it is in use. The device has a prism projector for a display to allow the user to view a visual output. It is a Single LCoS (Liquid Crystal on Silicon) display with a resolution of 640 × 360. In addition, a camera is mounted on top of the right frame of the Glass. This 5MP photo, 720p video camera allows the user to capture images and store them in its local storage. A second key hardware feature is the integrated bone conduction speaker, which allows transmitting sound directly into the user's ear canal without interference from outside noise. It is beneficial for a device meant to be used both indoors and outdoors. It also has a microphone to capture audio and voice input from the user, which is one of the main user-device interaction mechanisms. A secondary way to interact is the touchpad present on the side of the device.

The Glass comes with a lightweight dual-core Cortex A9 (2 × 1 GHz) processor by Texas Instruments, a built-in PowerVR SGX540 GPU, a 2GB RAM, and an internal storage capacity of 16GB. The device can perform moderate computations using these processing and storage capabilities. It also comes with a 570mAH battery capacity. In terms of connectivity, the Glass has a micro USB port that can connect to a suitable development environment for building and deploying applications on the device. In addition, it can connect to Bluetooth and is Wi-Fi 802.11g compatible. The device also has an accelerometer, gyroscope, magnetometer, ambient light sensor, and proximity sensor. A sample image of Google Glass is shown in Fig. 1.

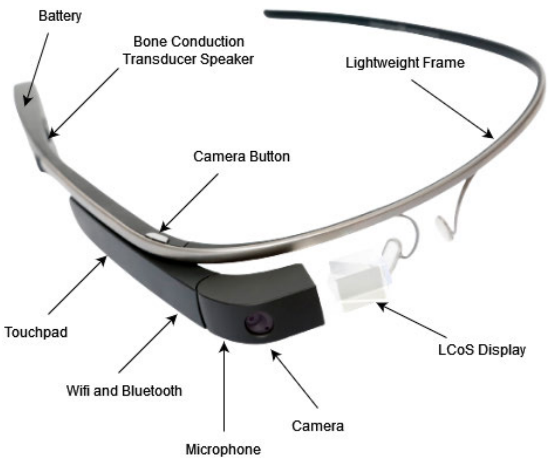


FIGURE 1. Google glass.

In their review of the applications of smart glasses in applied sciences, Kim *et al.* [53] found that the most popular commercial smart glass is Google Glass, followed by Microsoft's HoloLens. Their review shows that the android-based Google Glass is used in various domains of applied sciences, accounting for more than half of all the applications reviewed as part of their research. Furthermore, it is highlighted that since the device has an Android OS, it is effortless for developers to design and build applications on it. Moreover, Google Glass weighs only 36g, which is much lighter than other smart glasses in the market. For instance, Microsoft HoloLens 2 weighs 566g, Epson Moverio BT-350 weighs 151g, and Vuzix Blade M100 weighs 372g [53]. Similarly, in their paper on implementing an edge enhancement application on Google Glass, Hwang *et al.* [50] concluded that the device provides a valuable platform for implementing various applications that can aid patients with various low vision conditions. It is explained that since the device is reasonably priced, cosmetically appealing, highly flexible, and designed in a socially desirable format, it has a vast potential for further innovation. El-taher *et al.* [13], in their systematic review of navigation systems for the visually impaired, highlighted the importance of portability, wearability, latency, feedback interface, and user-friendliness of the application. Google Glass excels in all these critical design considerations. The device is lightweight, weighing only 36g, making it highly portable. Its cosmetic appeal, flexibility, and socially desirable format make it a highly wearable device. Feedback-interface can be defined as the means used by the application to convey information to the BVIP. Google Glass provides an excellent feedback interface due to the presence of a bone conduction speaker that renders audio signals to the user without obstructing any external sound, making it a safe choice for use in both indoor and outdoor environments. The device also has a microphone and any application developed on the device can be controlled entirely using audio-based commands giving the user excellent flexibility and comfort. The audio-based interface also helps keep the user experience as unrestricted as possible while using the device. Hence, due to its superior usability and features mentioned above, Google Glass was used in designing the visual assistant.

Finally, one of the critical aspects of a visual assistant device is low latency and the ability to run in real-time. In order to achieve this, the Custom Vision API from Azure Cognitive Services was used to run state-of-the-art deep learning models for scene description and object detection. It provides superior response time with excellent precision and accuracy. In order to further improve its precision on

[3] https://azure.microsoft.com/en-in/services/cognitive-services/

Indian scenarios, a newly annotated image dataset consisting of 5000 images was created, and the Vision API was trained on this dataset. Finally, the Vision API's precision and accuracy were compared against other state-of-the-art models run on a cloud-based intelligent server. Based on the performance of the Vision API and the superior usability of Google Glass, the proposed application was designed using them.

A. PROPOSAL

Some of the significant issues that restrict the usability of most wearable assistance systems were identified during the literature survey. Firstly, the size and weight of the sensors used in the system directly impact the long-term wearability, portability, and hence, the usability of the system without causing health hazards to the user. El-taher *et al.* [13] emphasized the importance of portability or weight and wearability of the device used for assisting the visually impaired person in their review of urban navigation systems for the visually impaired. Secondly, one of the most critical factors that must be considered while designing a system for the disabled is an intuitive human-computer interaction interface. The system must be designed such that it is easy to use with minimal user training. Finally, the response time from the source of computation must be close to real-time. Achieving real-time performance on a smart glass is very challenging since the algorithm's complexity directly impacts the device's response time unless the algorithm runs on a powerful machine, which is heavy and bulky and hence not portable. On the other hand, reducing the complexity of the algorithm leads to less accurate results. Therefore, it is essential to consider using cloud computing platforms with a fast response time for such systems. The following design choices are used to address the problems mentioned above, thereby improving the usability of the proposed visual assistant system.

1) Google Glass is selected as the core of the visual assistant system. The camera present in the device captures images of the surroundings, which are sent to a mobile app for further processing. Most of the previous applications that serve the purpose of visual assistants have bulky sensors and cameras attached, which are difficult to wear and are not portable. Given the superior portability, wearability, and flexibility of the device, the use of Google Glass will significantly improve the usability of such systems.
2) The application is designed to have a very intuitive interaction interface. Users can interact with it using a voice command that triggers the camera to capture an image and send it to the mobile app and the Vision API for further processing. The result from the Vision API is sent back to the Google Glass device, which is then converted to sound using the bone conduction transducer. The completely hands-free, voice-activated approach leads to superior user-system interaction and helps keep the user as unrestricted as possible while using the application.
3) The Custom Vision API provided by Azure Cognitive Services is used for performing the necessary computation on the image captured by the device. With the help of a cloud-based API, complex vision algorithms can be run on the image with almost real-time responses since the algorithm runs on powerful machines on the cloud. The use of cloud-based APIs prevents the need to carry a bulky computer for processing, thereby boosting the system's portability. The API can categorize the image into 86 categories and can be trained on custom datasets. It can further assign tags to the image and generate captions describing the contents in human-readable sentences.

A comparison of the proposed approach with existing assistive systems for the BVIP is shown in Table 1. The usability and functionality provided by the various applications are also shown. There are no applications that use Google Glass for scene description tasks in real-time on Indian scenarios. Further, the proposed application provides better portability and wearability in scene description tasks while providing a real-time response and a completely hands-free interaction interface. The key contributions of the proposed work are,

- The development of an augmented reality application for real-time scene description using Google Glass as an edge device and Azure Vision API for the BVIP.
- The creation of an annotated image dataset consisting of objects used by the BVIP in Indian scenarios and environments. The annotations correspond to the 86 class labels supported by the Vision API.
- Optimizing the performance of the Vision API by using the newly created annotated image dataset and using the custom vision[4] option provided by the Vision API.

Fig. 2 gives an overview of the proposed approach and the various components involved in it. The BVIP user wearing Google Glass captures the image of his/her surrounding by using the camera present on the device with the help of the voice command - "OK, Glass; Describe Scene." The captured image is compressed and sent via a Wi-Fi connection to the smartphone device of the user. Upon receiving the image, the smartphone app decompresses the image and invokes the Vision API to generate captions and identify the various objects in the image. The smartphone app then processes the API's response to extract the captions and the objects identified. This text response is sent back to Google Glass via the same wifi connection. Finally, Android's text to speech API is used to convert the text response into sound using the bone conduction transducer present in the device. In the following subsections, the proposed application development methodology and the user-system interaction design is described in detail.

[4] https://azure.microsoft.com/en-us/services/cognitive-services/custom-vision-service/#overview

TABLE 1. Comparison of the proposed application with existing assistive applications for the BVIP.

Literature	Source of Compute	Sensors Used	Functionality Provided	Usability (portability, wearability and output interface)
Mauro *et al.* 2015 [21]	Smartphone	Bluetooth Beacons	Auditory information is communicated about the nearest point of interest when the user is close to a Bluetooth beacon placed in different points of interest. Helps in navigation and providing spatial awareness in indoor settings	Highly portable and wearable since the system comprises only a smartphone. Auditory information is communicated using earplugs worn by the user
Barney *et al.* 2017 [30]	Arduino	Ultrasound Sensors	3D sound is generated to give the user a sense of the distance of the objects around him/her. Useful for navigation in indoor settings	Moderately portable and wearable as the system comprises an ultrasound sensor on a smart-glass, an Arduino for computing the distance of the surrounding objects, and a smartphone. Earphones are used to render the generated sound.
Jiang *et al.* 2019 [32]	Cloud	2 sets of CCD cameras and a semiconductor laser	Object detection using convoluted neural networks running on a cloud-based platform	Moderate level of portability and wearability as the system requires to be calibrated for effective binocular image acquisition. Moving the setup around might require re-calibration.
Bai *et al.* 2017 [19]	CPU and Microprogrammed Control Unit (MCU)	Eyeglasses, depth camera, ultrasonic rangefinder and AR glasses	Obstacle avoidance in an indoor environment with the help of depth and ultrasonic sensors	Low portability and wearability since the user must carry the CPU and MCU everywhere. The user is provided with auditory cues to avoid obstacles.
Neto *et al.* 2016 [51]	Laptop computer	Microsoft Kinect, gyroscope, compass sensor, IR depth sensor, stereo headphones	Face detection and recognition using an efficient face recognition algorithm based on HOG, PCA, and K-NN. 3D audio is generated on face recognition as the user-response. Microsoft Kinect is used to capture the RGB-D image of the person.	Low portability and wearability due to a laptop computer and a Microsoft Kinect, both of which are heavy and bulky. Good response interface with the help of 3D sound in the direction of the person identified in the image.
Pintado *et al.* 2019 [11]	Raspberry Pi	Raspberry Pi Camera Module V2	Shopping Assistant- Object recognition and price extraction using convolutional neural networks (CNN) running on a Raspberry Pi	Moderate portability and wearability since the user must carry a Raspberry Pi used as the computing source for running the CNN. It has very high latency, which can significantly reduce the practical usage of the application
Pēgeot *et al.* 2012 [27]	Laptop computer	Head mounted color camera	Scene text detection and tracking using Optical Character Recognition	Low portability and wearability since the user must carry a laptop computer for running the OCR algorithm. The user also needs to wear a head-mounted color camera for capturing images. The identified text is output as an audio signal using a text-to-speech library.
Takizawa *et al.* 2019 [52]	Laptop computer	Microsoft Kinect and a tactile feedback device on a cane	To recognize a pre-trained set of fixed 3D objects in the surrounding. The system also provides the user with instructions to find the 3D object	Low long-term portability and wearability since the user must carry a Microsoft Kinect and a laptop computer for processing. Vibratory cues are provided to the user to help with finding the 3D object.
Chang *et al.* 2021 [25]	Intelligent waist mounted device	Camera, time-of-flight laser-ranging module, 6-axis motion sensor, GPS module, LPWAN module	Zebra crossing safety for the visually impaired	Moderate long-term portability and wearability since the user must carry a waist-mounted device everywhere. Audio feedback is provided to the user with the help of Bluetooth earphones.
Chang *et al.* 2020 [35] and Chen *et al.* 2019 [36]	IR sensors, 6 axis gyroscope and accelerometer in smart glasses and intelligent cane	IR sensors, vibration motor, LPWAN module, 6 axis gyroscope and accelerometer	Aerial object detection using IR sensor data by calculating distance using the triangulation method, fall detection using the six-axis gyroscope and accelerometer in smart glasses, and intelligent cane and notification system in case of fall detection	Highly portable and wearable as the compute is performed by sensors on the smart glasses and intelligent cane. High reliability due to the presence of a notification mechanism in case of fall detection. Vibratory cues signal the presence of aerial obstacles in front of the user.
Chang *et al.* 2019 and 2020 [7] [8] [9]	AI-based intelligent drug pill recognition box with a pre-trained deep learning model	Camera on smart glasses, drug pill recognition box with wifi-capabilities	Drug pill recognition for the visually impaired	Moderately portable as the user must carry the intelligent drug pill recognition box. High wearability as the images are captured with the help of smart glasses. Audio signals are generated to provide reminders to the user and inform the correct or incorrect identification of drugs.
Proposed Method	Azure Vision API used to generate captions and identify objects on the image	Google Glass having a camera, a microphone, a bone conduction transducer, Wi-Fi capability and more	Image captioning and object detection in real time	Highly portable and wearable as the only devices that the user must carry is a smartphone and Google Glass. Audio output is produced with the help of a bone conduction transducer which prevents the obstruction of external sound. Completely hands-free application with voice command capabilities

B. PROPOSED APPLICATION DEVELOPMENT METHODOLOGY

According to the official documentation by Google,[5] the three major design patterns for developing software on Google Glass, also called Glassware, are Ongoing Task, Periodic Notifications, and Immersions. Ongoing tasks are long-running applications that remain active even when users switch focus to a different application within the device. A stopwatch app is an excellent example of an ongoing task. Users can switch to a different application while running

[5] https://developers.google.com/glass/design/patterns

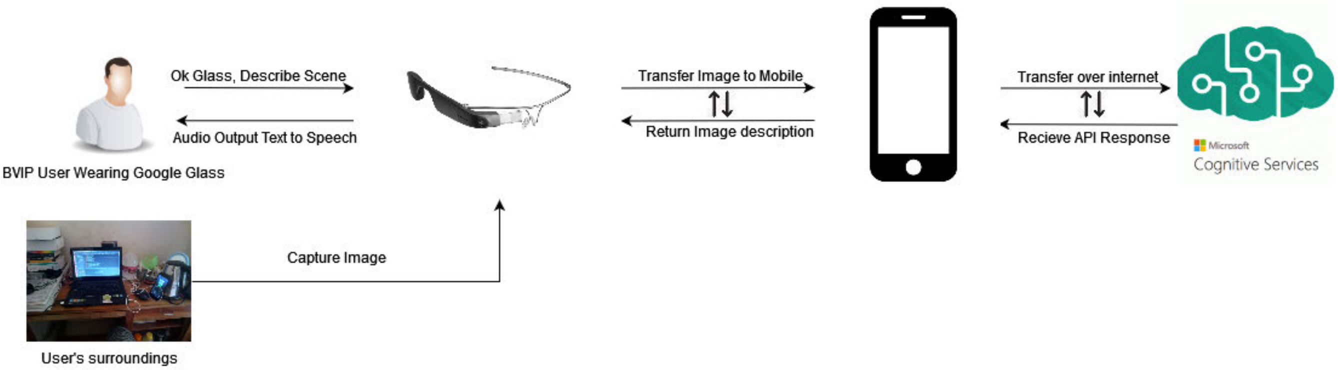


FIGURE 2. System overview.

the stopwatch app without stopping the timer. The Periodic Notifications design pattern is used to develop applications where the user is notified of any new information to be displayed. Examples of applications that use the Periodic Notification design pattern include a news app, an SMS reader, or an email reader. The Immersion design pattern is used whenever the application requires complete control of the user experience. These applications stop when the user switches focus to a different app. Any gaming application is an excellent example of an Immersion Pattern. The proposed visual assistant requires complete control of the user experience, and hence the Immersion Pattern is chosen to design the application.

The system design diagram is shown in Fig. 3. The system can be divided into three major sections: the app on the Google Glass device, the smartphone, and the Vision API. The BVIP user interacts directly with the app on Google Glass. On receiving a user voice command, the camera image handler built into the app uses the camera present on the smart glasses to capture the image of the user's surroundings. This image is compressed and then sent to the smartphone using a socket connection over the internet. The image is compressed to reduce the size of the data to be sent over the internet, thereby reducing the application's response time. Socket programming is a way of connecting two nodes. One node (server) listens on a particular port at an IP, while the other node (client) reaches out to the server node on the same port to form a connection. In this system, the application on the Google Glass is the client, and the application on the smartphone forms the server side of the socket connection.

Upon receiving the image from Google Glass, the server-side application on the smartphone decompresses the image. The captions of the decompressed image are then generated by using the Vision API. The response from the API is received in a JSON format by the Cognitive Services API interface built on the smartphone app. JSON stands for JavaScript Object Notation. It is an open standard data interchange format that uses human-readable text to store and transmit data objects consisting of attribute-value pairs and arrays. The smartphone app processes the JSON response to extract the captions and the objects identified in the image. The processed response is then sent back to the client-side application on Google Glass over the same socket connection. Finally, on receiving the text response from the smartphone, the app on the Google Glass device uses a text to speech API provided by Android to convert the text to audio signals, which is rendered as sound by using the bone conduction speaker present on the device. The BVIP user hears this sound output.

The version of Glass used in developing the proposed system is the Glass Explorer Edition. It comes with a custom Glass OS and Software Development Kit developed by Google. Glass OS or Google XE is a version of Google's Android operating system designed for Google Glass. The operating system version on the Explorer Edition device was upgraded from XE 12 to XE 23 since Android Studio, the integrated development environment (IDE) used for developing the app, supports XE 23, and the SDK documentation available online is also for XE18+. The OS version was upgraded by flashing the device, which was done by programming the bootloader of the Glass.

Kivy,[6] an open-source python library, was used for developing the socket server application on the smartphone. It is a cross-platform library for the rapid development of applications that make use of innovative user interfaces. It can run on Windows, OS X, Android, iOS, and Raspberry Pi. Hence, the server-side of the application can be started on any smartphone, laptop computer, or Raspberry Pi. However, to increase portability and ease of use, smartphones were chosen for the proposed system. The Azure Vision API used to identify the various objects and generate captions of the captured image provides excellent results in real-time. It can be used to categorize objects into 86 different categories. The performance of the API was evaluated against Flickr8k [54] and Microsoft COCO [55] datasets. Different standard evaluation metrics, namely, BLEU, METEOR, ROUGE-L, and CIDEr, were used to evaluate the API, and the evaluation results are shown in Table 2. The response from the API is returned in JSON format. We process the JSON and return the description of the image and the various objects in the

[6] https://kivy.org/#home

TABLE 2. Evaluation metrics of the Azure Vision API against Flickr8K and Microsoft COCO datasets.

Dataset	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Flickr8K	0.383812	0.228540	0.145145	0.088268	0.168121	0.333856	0.594539
MS COCO	0.396322	0.256324	0.173089	0.117195	0.182853	0.357928	0.804114

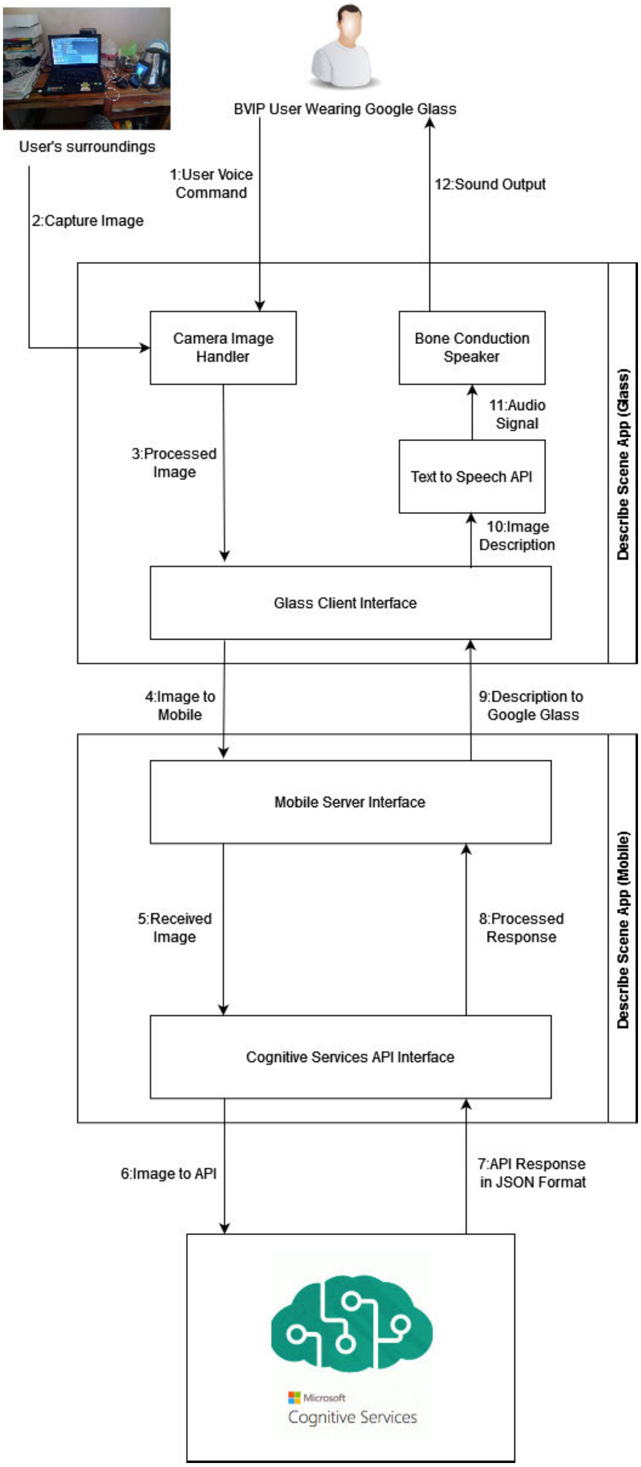


FIGURE 3. System design.

image in text format back to the device. Fig. 4 shows a sample image from the Microsoft COCO [55] dataset, and the caption generated by the API is "A bedroom with a bookshelf and a mirror."

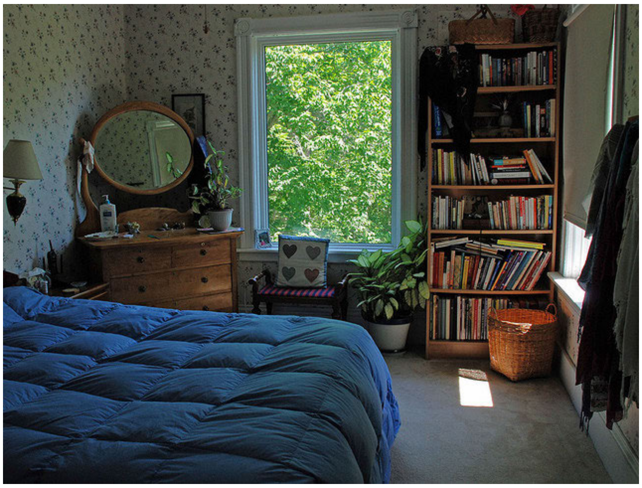
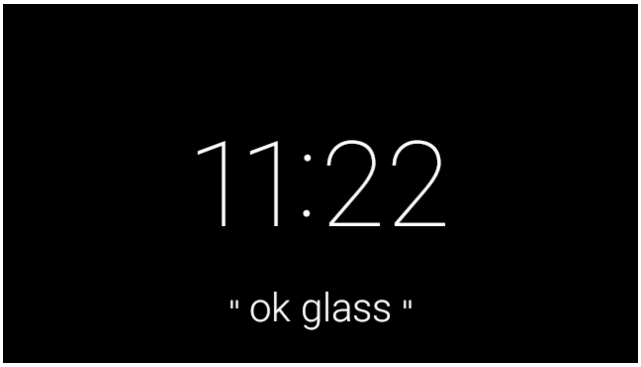


FIGURE 4. Sample image.

From Table 2, it is observed that the performance metric scores can be significantly improved. The Azure Vision API is trained and tested on images obtained from non-Asian countries. Hence, the API can be fine-tuned and customized to Indian scenarios by using Azure Custom Vision API. This feature enables training and testing the Vision API on local image datasets, thereby making the API more robust to local settings. Here, several images were added to the training dataset for better performance. A new image dataset was compiled centered around the daily routine of the BVIP. The images were annotated with class labels already supported by the Vision API. There are 86 different class labels, and a minimum of three images for each category was collected. The BVIP subjects were surveyed to understand their routine, and it was found that they extensively used the following categories: keys, remote, medicine, mobile phone, prescription glasses, and umbrella. A minimum of 50 images was collected for each of the six categories and was used for training the API. The standard annotation procedure is followed, and it is discussed in the Results and Analysis section.

C. USER-SYSTEM INTERACTION

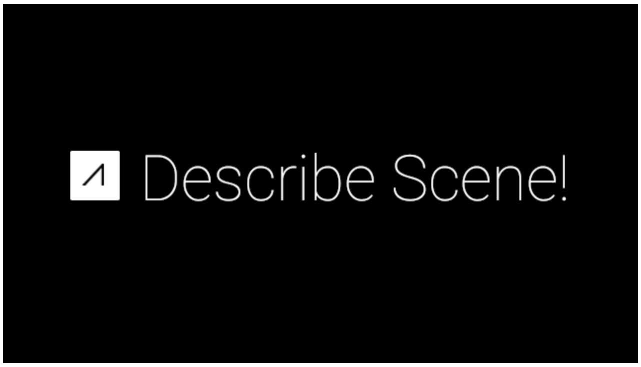
The system is designed in such a fashion to make the user-system interaction entirely audio-based, thus providing the best user experience to a BVIP. Google Glass has a sensor that can detect whenever a user wears it, and the device is configured such that it automatically turns on whenever the user wears the device. The Home screen shown in Fig. 5 is displayed as soon as the user wears the device. Once the device is worn, the following steps are to be followed.



11:22

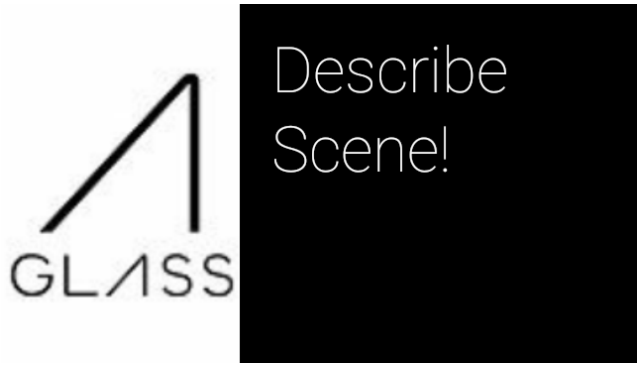
" ok glass "

FIGURE 5. Home screen.



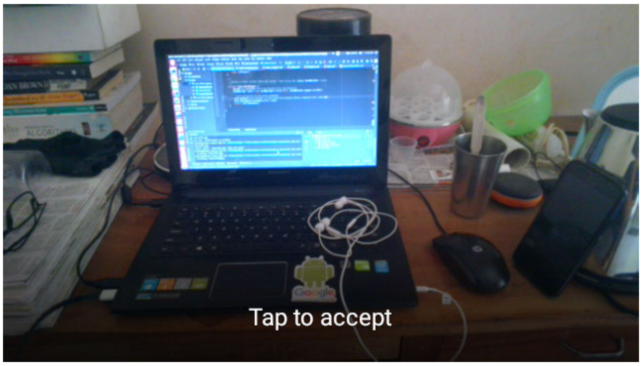
Describe Scene!

FIGURE 6. Describe scene: Invocation screen.



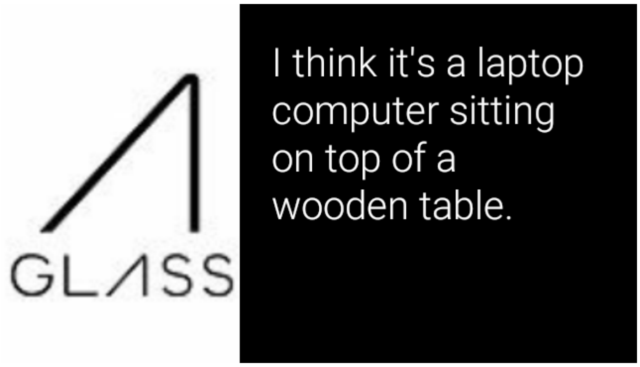
Describe Scene!

FIGURE 7. Main activity screen.



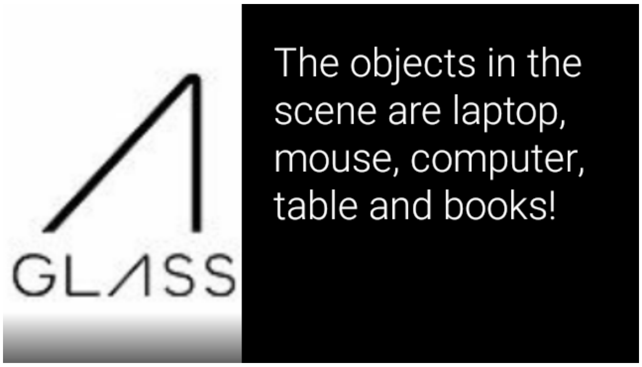
Tap to accept

FIGURE 8. Image captured using the camera on glass.



I think it's a laptop computer sitting on top of a wooden table.

FIGURE 9. Caption response.



The objects in the scene are laptop, mouse, computer, table and books!

FIGURE 10. Objects detected response.

- **Step 1**: Say, **"OK, Glass."** The device recognizes the command and takes the BVIP user from the home screen to the menu screen, containing the list of vocal invocations for various applications installed on the device. It also sends a beep sound so that the user can be assured that the command is recognized and executed. The menu or voice invocation screen is shown in Fig. 6.
- **Step 2**: Say, **"Describe Scene"**. One of the voice commands present on the invocation menu is for starting the virtual assistant. The voice command is "Describe Scene." Upon execution of the command, the main activity screen of the application is displayed on the device. This screen is shown in Fig. 7. Once the application starts and the main activity screen is visible, the camera intent is activated. The camera captures the image of the surroundings in front of the user. The captured image is shown in Fig. 8.

After capturing the image, the device sends it to the socket server, running on the user's smartphone. The description of the image and the various objects present in it are recognized using the Vision API. The generated response is sent back to the Google Glass in text format and is converted to speech by using the bone conduction transducer present in the device. The captions and objects detected are also displayed on the device, as shown in Fig. 9 and Fig. 10, respectively.

Figs. 5, 6, 7, 8, 9, and 10 are used to explain the flow of the application to the readers. The user can use the device without seeing the screen. To summarize, the user has to use the voice command **"OK, Glass"** followed by **"Describe Scene"** to launch the application. Thus, a wearable assistant for the visually impaired was developed by using only voice commands to interact with the application.

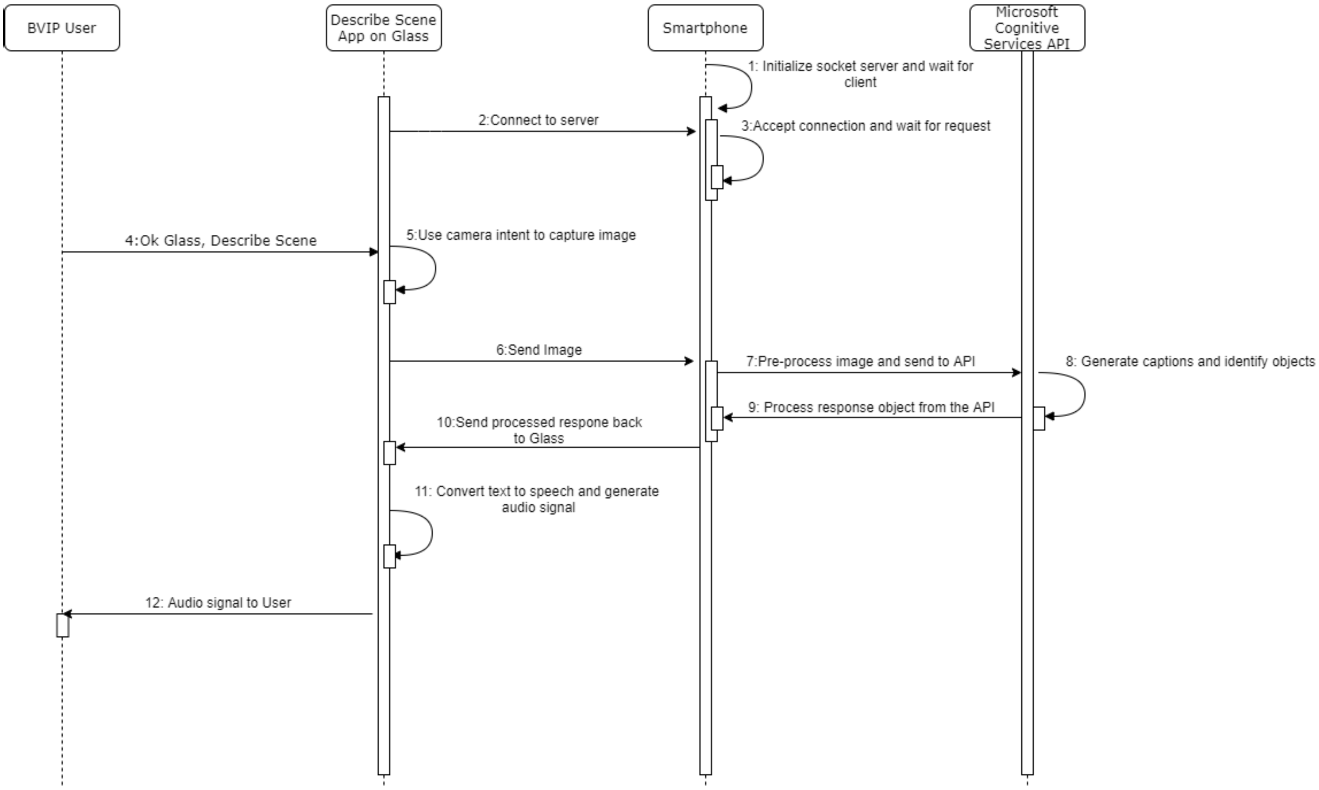

BVIP User
Describe Scene App on Glass
Smartphone
Microsoft Cognitive Services API
1: Initialize socket server and wait for client
2:Connect to server
3:Accept connection and wait for request
4:Ok Glass, Describe Scene
5:Use camera intent to capture image
6:Send Image
7:Pre-process image and send to API
8: Generate captions and identify objects
9: Process response object from the API
10:Send processed respone back to Glass
11: Convert text to speech and generate audio signal
12: Audio signal to User

FIGURE 11. User-system interaction diagram.

A detailed user-system interaction diagram is shown in Fig. 11. It displays the various steps in order of occurrence while using the app. Firstly, the socket server application is started on the smartphone. This server application waits for a client connection from the Google Glass device. The server-client connection is firmly established on receiving a connection request from the smart glasses. This connection remains intact for all interactions between the Google Glass and the smartphone. Next, the BVIP user interacts with the application with the voice command described earlier: "Ok Glass, Describe Scene." As shown in the interaction diagram, the voice command triggers a series of steps on the smart glasses and the smartphone, starting with capturing the image in front of the user and sending it to the smartphone app. Here, after processing the received image, captions are generated, and the objects in the image are identified by using the Vision API. The output of the API is sent back to the application on Google Glass via the smartphone app, and this output is eventually heard by the BVIP user wearing the device.

IV. RESULTS AND ANALYSIS

A. EXPERIMENTAL SETUP

The experimental results and tests were done on a Google Glass Explorer Edition, which comes with a dual-core Cortex A9 (2 × 1 GHz) processor by Texas Instruments, a built-in PowerVR SGX540 GPU, a 2GB RAM, and an internal storage capacity of 16GB. It has a 570mAH battery capacity and is Wi-Fi compatible. The smartphone application was built on a OnePlus 7 phone running Android 10 OS, with a Snapdragon 855 processor, 6 GB RAM, and 128 GM ROM. Application development on both devices was done using Android Studio, an Interactive Development Environment for building Android applications.

The accuracy and precision of Azure Custom Vision API were measured against other state-of-the-art vision models on a Dell G7 laptop computer with an Intel Core i7-9750H CPU running at 2.60GHz and a 16GB RAM. It runs on a Windows 10 OS with an NVIDIA GeForce RTX 2060 GPU and 8GB video memory. For measuring the proposed system's latency and for testing the application on the BVIP, a 4G network was used, which was provided by a local internet service provider. Table 3 shows the attributes used for training the Custom Vision API.

B. ANNOTATION

Five different annotators are used in this study, and at least two different annotators are used to annotate each image by identifying the different objects present in it. Each annotation includes the bounding box and the class label of the identified object. The annotation process followed is the standard procedure for annotation described in [56]. The Custom Vision API has an interface that loads the annotated images and provides tools to place the bounding box and the class label. Once the annotation is done for each image, all the corresponding bounding box coordinates and the class labels present in that image are stored in the API's internal

TABLE 3. Training setup for the custom Vision API.

Attribute	Details	Attribute	Details
Validating and testing batch size	100 Images	Optimize	Adaptive Momentum
Epochs	4000	Loss	Categorical Entropy
Learning rate	0.1	Weight initialization	Pre Trained weights of Imagenet v3 model trained over ImageNet, MS COCO, and Flikr. The model is fine-tuned by retraining the bottleneck layer
Classifier at Bottleneck layer	Softmax classifier	Testing set	10% of training dataset
Number of classes	86	Validating set	10% of training dataset

database. The annotators were informed about all 86 class labels supported by the API. They were also provided with the definition (the smallest rectangle with vertical and horizontal sides surrounding an object) and documentation of bounding boxes. All the annotators were previously familiar with the concept of object localization and classification, and hence no further training was provided. Though the class labels were clear in most cases, the tight bounding box in the case of multiple overlapping objects was an issue, and annotators' reliability is also measured. Since the number of annotators is more than two, quadratic-weighted Cohen's κ (κ_w), and the leave-one-labeler-out agreement was used as shown in Equation 1, where, p_{ij} are the observed probabilities, $e_{ij} = p_i q_j$ are the expected probabilities and w_{ij} are the weights (with $w_{ji} = w_{ij}$). The annotators reliably agree when discriminating against the recognized class label with Cohen's $\kappa = 0.94$.

$$\kappa_w = 1 - \frac{\sum_{i,j} w_{ij} p_{ij}}{\sum_{i,j} w_{ij} e_{ij}} \tag{1}$$

Here, the standard error (se) is calculated using Equation 2.

$$se_w = \frac{1}{1 - p_{e(w)}} \times \sqrt{\frac{\sum_{i,j} p_{ij}[v_{ij} - u_{ij}(1 - \kappa_w)]^2 - [\kappa_w - p_{e(w)}(1 - \kappa_w)]^2}{n}} \tag{2}$$

where,

$$v_{ij} = 1 - \frac{w_{ij}}{w_{max}}$$

$$p_{e(w)} = \sum_i \sum_j v_{ij} p_i q_j$$

$$u_{ij} = \sum_h q_h v_{ih} + \sum_h p_h v_{hj}$$

Accuracy Comparison: Cohen's κ is computed to compare the accuracy of the Vision API against human annotations [56], [57]. The results are shown in Table 4, where it can be observed that the average κ value varies from 0.91 for the API's classification of 80 class labels to 0.94 for the six other class labels (keys, remote, medicine, mobile phone, prescription glasses, and umbrella) of the considered Vision API. The classification accuracy reduces if there are multiple overlapping objects in the images. Also, we observe that the human annotation results vary in line with the API classifications, which vary between 0.91 and 0.96 for the Vision API class labels. From Table 4, it is observed that the API classification of class labels performs equally well when compared to inter-human annotations.

TABLE 4. Cohen's κ for train-test results of annotated images.

Environment	API	Human Annotation
80 Class labels of Vision API	0.91 (0.96)	0.91 (0.94)
6 Other class labels of Vision API	0.94 (0.97)	0.96 (0.98)

C. DATA AUGMENTATION

The collected data for training contains 86 different class labels with a minimum of 3 images for each category, and for a few class labels such as keys, remote, medicine, mobile phone, prescription glasses, and umbrella, more than 50 images per class are collected and annotated. Though the total number of images considered is more than 540, the variants of images considered are fewer as the images are taken from cameras with three different angles, i.e., front view, side view, and top view. Only these three angles were considered since all other variants can be generated using data augmentation. So, data augmentation is used to increase the training data tenfold, thereby increasing its robustness [58], [59]. The data augmentation techniques used are given below. Furthermore, the augmentation values used for the data augmentation are given in Table 5. After data augmentation, the total number of annotated images is 5000.

- channel_shift_range: Random channel shifts of the image.
- zca_whitening: Applies ZCA whitening to the image.
- rotation_range: Random rotation of image with a degree range.
- width_shift_range: Random horizontal shifts of the image with a fraction of total width.
- height_shift_range: Random vertical shifts of the image with a fraction of total height.
- shear_range: Shear intensity of the image where the shear angle is in the counter-clockwise direction as radian.
- zoom_range: Random zoom of the image where the lower value is 1-room_range and upper value is 1+zoom_range.
- fill_mode: Any of constant, nearest, reflect or wrap. Points outside the boundaries of the input are filled according to the selected mode.
- horizontal_flip: Randomly flip the inputs horizontally. Table 5 shows the details of different data augmentations performed on the dataset.

TABLE 5. Types of data augmentation used.

Type of Augmentation	Augmentation Value
channel_shift_range	20
zca_whitening	TRUE
rotation_range	40
width_shift_range	0.2
height_shift_range	0.2
shear_range	0.2
zoom_range	0.2
horizontal_flip	TRUE
fill_mode	Nearest

TABLE 6. COCO object detection results comparison using different frameworks and network architectures vs Azure Custom Vision API. mAP is reported with COCO primary challenge metric (AP at IoU=0.50:0.05:0.95.)

Vision Framework	Model Used	mAP	Billion Mult-Adds	Million Parameters
Azure Custom Vision API	-	26.33%	116	37.43
SSD 300	deeplab-VGG	21.10%	34.9	33.1
	Inception V2	22.00%	3.8	13.7
	MobileNet	19.30%	1.2	6.8
Faster-RCNN 300	VGG	22.90%	64.3	138.5
	Inception V2	15.40%	118.2	13.3
	MobileNet	16.40%	25.2	6.1
Faster-RCNN 600	VGG	25.70%	149.6	138.5
	Inception V2	21.90%	129.6	13.3
	MobileNet	19.80%	30.5	6.1

TABLE 7. Comparison of Accuracy Results of different models vs Azure Custom Vision API.

Model	Accuracy	Billion Multi-Adds	Million Parameters
Azure Custom Vision API	73.1%	116	37.43
MobileNet-224	70.60%	569	4.2
GoogleNet	69.80%	1550	6.8
VGG 16	71.50%	15300	138
Squeezenet	57.50%	1700	1.25
AlexNet	57.20%	720	60

D. PERFORMANCE EVALUATION

After training with the newly annotated image dataset, the Custom Vision API's performance was compared against other deep learning computer vision frameworks, and the results are presented in Table 6. The mean Average Precision (mAP) value was computed on the MS COCO dataset using COCO primary challenge metric.[7] It is calculated by taking the average of multiple IoU (Intersection over Union as shown in Equation 3) values ranging from 0.5 to 0.95 with a step of 0.05. The number of operations and parameters involved in the calculation is also given in Billion Mult-Adds and Million Parameters, respectively. The results show the API performing better than other state-of-the-art vision models such as SSD 300, Faster-RCNN 300, and Faster-RCNN 600. Similarly, the performance of the API on the ImageNet dataset [60] is given in Table 7. From the results, it can be concluded that the API has better performance than most state-of-the-art models for computer vision.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{3}$$

TABLE 8. Latency results.

Frame Resolution	224*224	512*512
	Delay in milliseconds	
Smartphone Time	300	500
Glass Time	200	300
Edge(API) Time	100	150

The mAP of the Vision API is calculated on the newly computed dataset of 5000 annotated images before and after training the Custom Vision API on the new dataset. It is observed that the mAP value increases from 63% to 84% with IoU $>$ 0.5 after training the Custom Vision API.

The application's latency was measured for two resolutions of the captured image - 224*224 and 512*512 pixels and the results are shown in Table 8. The time measured can be classified into three different categories:

1) Smartphone Time: the time taken on the smartphone app
2) Glass Time: the time taken on the Google Glass device
3) Edge Time: the time the Vision API takes to generate the captions for the captured image, identify the objects present in it, and return the results over the Wi-Fi back to the smartphone.

For both the resolutions, the application has a response time of less than 1 second. All the latency values were measured on a 4G network.

Comparison with other computer vision and image recognition APIs: There are several APIs such as Watson, Clarifai, Imagga, and Parallel dots. Though few APIs have better mAP values than Azure Vision API for the standard datasets [61], the customizable option provided by the Azure Vision API, of using images belonging to various other categories which are not present in the standard datasets makes the Azure Vision API a better choice. Apart from that, this API also has computer vision features such as blob detection & analysis, building tools, image processing, multiple image type support, reporting/analytics integration, smart camera integration and also supports the integration with Microsoft Azure Cloud network and various other virtual and augmented reality tools like Microsoft Kinect and so on. All of these make this API a better choice than the rest.

E. LIKERT SCALE ANALYSIS

With the help of students (50) and teachers (5) at the **Roman and Catherine Lobo School for the Visually Impaired** at Mangalore, Karnataka, India, the application was tested, and its usefulness to the BVIP community was determined. The students who took part in the study belonged to the age group of 12 to 15 years and were in their high school years. They were under the supervision of their teachers during the study, who were 30 to 50 years old. After demonstrating how to use the device, the students were asked to use it in their school environment to identify and recognize different areas within the school boundary, such as their classroom, dorm, and playground. Objects like chairs, windows, doors, beds, and stairs were some of the different indoor objects

[7] https://cocodataset.org/#detection-eval

identified using the device. Some of the students who used the device in the playground within the school premises were able to identify outdoor scenes, which included trees, swings, pet cats, and dogs. Using the description given by the device, the students could accurately identify their current location within the school. After performing the study, to determine the application's usefulness, a set of hypotheses was formulated, along with corresponding questionnaires for each of these hypotheses. The feedback and answers to the questions were recorded and presented in the form of charts. The following hypotheses were formulated for the study:

1) **User training period is minimal:** As already described in the section, there are two voice commands to use the application. The first voice command is **"OK Glass,"** followed by the second command, **"Describe Scene."** The voice commands were found very intuitive by the users. The most significant advantage of the proposed system is that the user does not require any visual cues to use the application.
2) **No extra effort is required to use this device daily:** The device is fairly simple to use. The navigation through the device is entirely audio-based. Each of the two voice commands is followed by a beep sound, and the result is the audio-based description of the scene. On receiving the description, the user can recognize a different screen by starting over. Finally, the voice recognition software provided by Google was found to be very effective.
3) **The application helps the user to understand the scene:** Since the application generates captions of whatever scene the person is looking at, it was hypothesized that the application would help the user better understand their surroundings.
4) A null and alternate hypothesis was also formulated:
   - **Null Hypothesis:** A visually impaired person would not prefer to use the application.
   - **Alternate Hypothesis:** A visually impaired person would prefer to use this application every day.

Questionnaires were formulated to evaluate the above hypotheses. The questions are as follows:

Hypothesis 1: User training period is minimal.

1) Were you able to effectively use this application yourself after three or fewer trials/walkthroughs?
2) While trying this application, did you feel confused at any point?
3) After a prolonged period of not using the application, would you use it with the same efficiency you are using now? (Would you be able to remember how to use the device ?)

Hypothesis 2: No extra effort is required to use this device daily.

1) Do you consider wearing this device irritating/ troublesome?
2) How many times (out of five) is your voice recognized by the device?

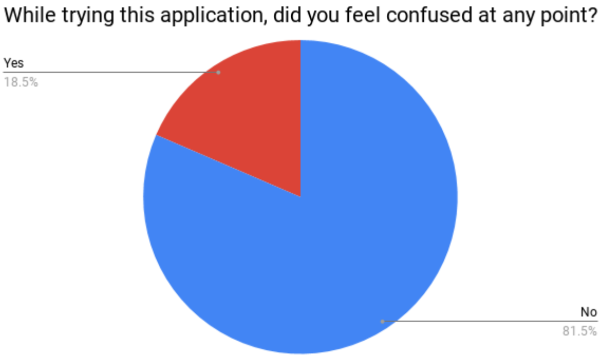


FIGURE 12. Hypothesis 1 Question 2.

3) Would you prefer to use this application instead of having a guide? If not: Would you prefer to use this application when your guide is not available?

Hypothesis 3: The application helps the user to understand the scene.

1) Do you think the device identified all the objects in the scene?
2) Do the identified objects help in better understanding the scene?
3) Are the objects correctly identified?

Hypothesis 4:

Null Hypothesis: A visually impaired person would not prefer to use the application.

Alternate Hypothesis: A visually impaired person would prefer to use this application every day.

1) What do you use to walk in and around your neighborhood? Cane, Guide, Other.
2) Would you prefer a guide or would you rather walk alone?
3) Do you have access to the internet in your area?
4) How would you rate the response time of the application?
5) How likely are you to use this application every day?
6) How comfortable are you with the audio-based interface?
7) Are you able to hear the output from the device?
8) How well do you think the description given by the application matches the actual scene (As described by the guide)?
9) Would you prefer voice-based or touch-based navigation?

A Likert Scale Analysis on the usability of the application was performed using user feedback and responses to the above questions. The following pie charts depict the responses to some of the questions received from the users.

Inference: Fig. 12 gives the percentage of people who felt confused while trying the device. As can be seen, the majority of the users did not feel confused. This question is concerning **Hypothesis 1:** User training period is minimal. The less confused the user in his(her) first attempt at using the application, the smaller the training period.

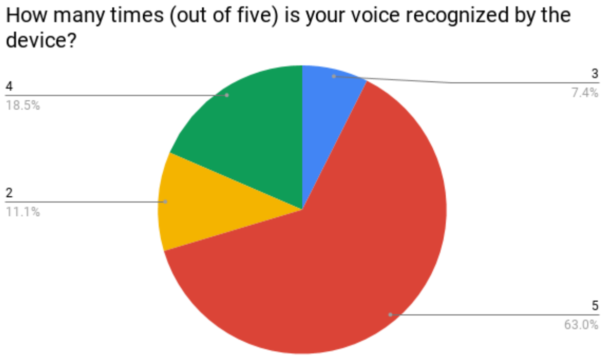


FIGURE 13. Hypothesis 2 Question 2.

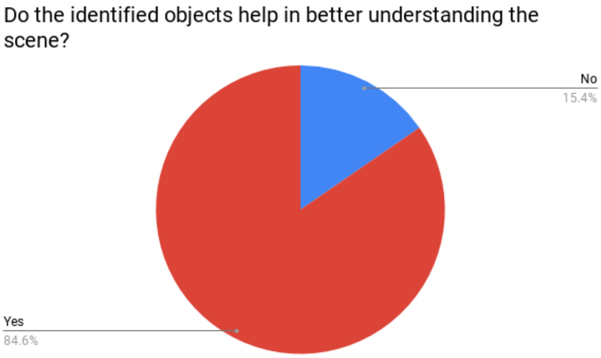


FIGURE 14. Hypothesis 3 Question 2.

Inference: Since the application is entirely audio-based, the users' voices must be correctly recognized. From Fig. 13, it can seen that the users' voice commands are being recognized most of the time correctly. This question is concerning **Hypothesis 2:** No extra effort is required to use this device daily. The fewer times the user has to repeat the voice command, the lesser the effort put into using the application.

Inference: From Fig. 14, it can be concluded that the objects identified from the image helped the user better understand the scene. Thus, the object detection model complements the image captioning model. This question is concerning **Hypothesis 3:** The application helps the user to understand the scene.

Statistical Analysis: To further understand the four different hypotheses, fifty-five subjects were compared with independent sample t-test and χ^2-squared tests on various parameters such as age, gender, the severity of visual impairment, and intellectual level. The differences between groups on parametric data such as chronological age and age of onset of blindness were evaluated with an independent sample t-test, and the non-parametric data such as gender and severity of blindness were evaluated with the χ^2-squared test. School records included age, gender, the severity of visual impairment, and intellectual level. The severity of visual impairment was categorized as 'total blindness,' 'near blindness,' 'profound vision loss,' and 'severe vision loss.' Similarly, the intellectual level was classified into 'normal,' 'borderline,' or 'mental retardation.' Age included 'chronological age' and 'age of onset of visual impairment.' There was no statistically significant difference between the groups in terms of age (18.86 ± 3.05), age of onset (8.81 ± 20.85 months), and gender ($\chi^2 = 0.02$, d.f. = 1, P = 0.95) w.r.t the four hypotheses: training period, the effort required to use the device, application's impact in scene understanding and the inclination to use this application every day. Also, there was no significant difference in severity of blindness for the four hypothesis categories ($\chi^2 = 10.24$, d.f. = 2, P = 0.15). However, we found a significant difference for the intellectual level ($\chi^2 = 36.11$, d.f. = 3, P = 0.001) as the students with borderline and mental retardation found it difficult to understand and use the application. For this complete statistical analysis, the significance level was set with $P < 0.05$.

F. SIGNIFICANCE

The proposed approach was compared with existing state-of-the-art solutions, and the details are shown in Table 1. Previously, devices like smartphones, Bluetooth beacons, and Raspberry Pi were used to develop diverse solutions. The proposed approach attempts to tackle the problem by using Google Glass. Keeping BVIP users in mind, the application was designed to be wholly audio-based, and the user does not require any visual cues to use the device. Another significant improvement is that the user does not need to carry any bulky hardware while using the proposed system. The hardware used here is Google Glass, which is very similar to any regular reading glasses in size and shape, and a smartphone which makes the application highly portable and easy to use. Hence, the proposed system is highly wearable, portable, and provides accurate results in real-time.

A Likert Scale Analysis on the usability of the application was performed. Positive feedback and response were received from the users, as shown in the charts in Figures 12, 13, and 14. It can be concluded from the response that the application can be used effortlessly on a daily basis to understand the BVIP user's surroundings. It can be further concluded that the BVIP require minimal to no training to use the device, and they prefer to use the application as a visual assistant.

G. LIMITATIONS

While testing, certain limitations of the application were identified. Firstly, the proposed system is highly dependent on a strong internet connection and works if and only if there is an internet connection available in the area. The latency of the application was found to vary significantly due to fluctuations in the network speed. Secondly, the device is relatively expensive in developing countries and is not easily affordable. Finally, the battery on the Google Glass was able to run only for 4 hours per charge while using the application continuously. However, the short runtime problem can be overcome by adding an external power pack.

A few other improvements were identified while collecting feedback from the BVIP students and teachers. For instance, a few users commented on the ability of the device to understand regional accents, stating that the voice command was not recognized in certain instances. The statistics are shown in Fig. 13. Another feedback received on very similar grounds was that the audio output from the device can be personalized such that the voice output has a regional accent. The users explained that this would help make the application feel more personalized to the user, given that Indian accents are now available on various electronic gadgets. One BVIP user commented on the audio output being affected in boisterous environments, such as noisy traffic junctions or construction sites. However, this problem was mitigated by switching to Bluetooth earphones instead of the bone conduction transducer, in which case the audio output was not affected by external sounds. The BVIP users explained that the application on Google Glass provides more comfort and usability when compared with smartphone apps for the visually impaired.

V. CONCLUSION

The use of Google Glass to assist the BVIP community is demonstrated by developing an application that acts as a visual assistant. The system is designed to be highly portable, easy to wear, and works in real-time. The experimental results of the Azure Vision API show a mean Average Precision value (mAP) of 29.33% on the MS COCO dataset and an accuracy of 73.1% on the ImageNet dataset. A dataset of 5000 newly annotated images is created to improve the performance of scene description in Indian scenarios. The Custom Vision API is trained and tested on the newly created dataset, and it is observed that it increases the overall mAP from 63% to 84% with IoU > 0.5 for the created dataset. The overall response time of the proposed application was measured and is less than 1 second, thereby providing accurate results in real-time. The proposed application describes the scene and identifies the various objects present in front of the user. It was tested on the BVIP, and their response and feedback were recorded, and a Likert scale analysis was performed. From the analysis, it can be concluded that the proposed system has an excellent potential to be used as an assistant for the BVIP.

The computer vision API from Azure Cognitive Services can add more functionalities to the proposed application. The capabilities of other APIs can be explored to add more functionalities such as text extraction and reading using Read API and face detection and recognition using Face Service.[8] The application can be enhanced by adding more features, such as lane detection, fall detection, pit detection, obstacle avoidance, and shopping assistant, thereby creating a one-stop assistant for the BVIP. Google Glass has embedded sensors that can achieve these functionalities with little to no need for external sensors. Further, there exists a possibility of moving the application entirely to Google Glass by removing the dependency on the smartphone. Currently, the smartphone device is used to process the captured image before making the API calls to the Custom Vision API, which can be avoided by using the Android SDK for Vision API[9] directly on Google Glass.

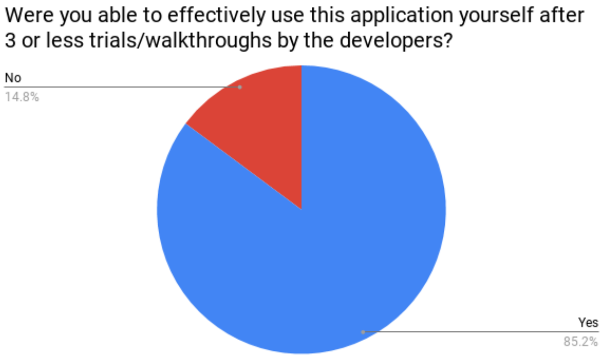


FIGURE 15. Hypothesis 1 Question 1.

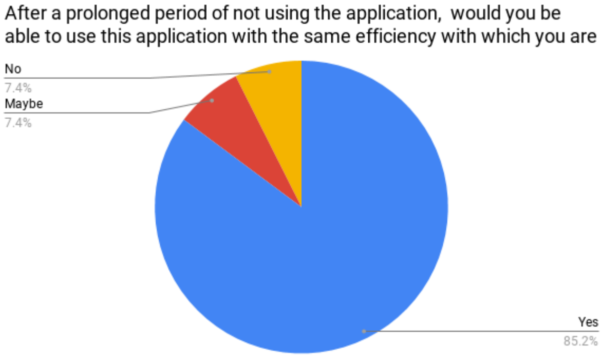


FIGURE 16. Hypothesis 1 Question 3.

Declaration: The experimental procedure and the entire setup, including Google Glass given to the participants, were approved by the Institutional Ethics Committee (IEC) of NITK Surathkal, Mangalore, India. The participants were also informed that they had the right to quit the experiment at any time. The collected data, i.e., video recordings, audio, and the written feedback of the subjects, was taken only after they gave written consent for the use of their collected data for the research experiment.

APPENDIX A
RESULTS OF LIKERT SCALE ANALYSIS

The following pie charts are obtained by performing the Likert Scale Analysis on the usability of the application. In order to perform the analysis, we formulated four types of hypotheses and generated corresponding questionnaires to evaluate these hypotheses. We asked these questions to the

[8] https://azure.microsoft.com/en-us/services/cognitive-services/face/

[9] https://github.com/microsoft/Cognitive-Vision-Android

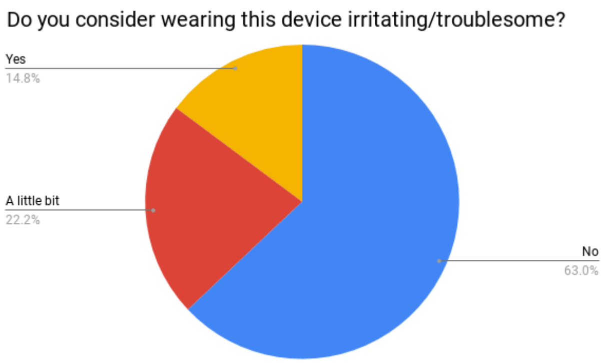


FIGURE 17. **Hypothesis 2 Question 1.**

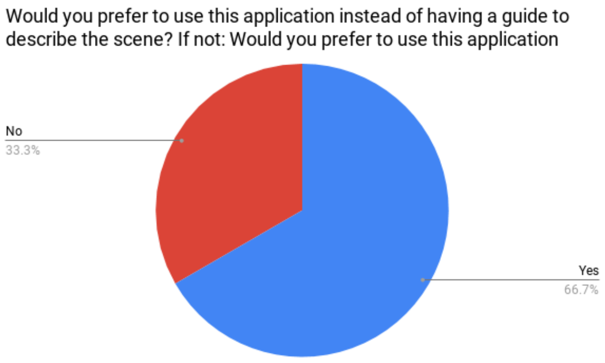


FIGURE 18. **Hypothesis 2 Question 3.**

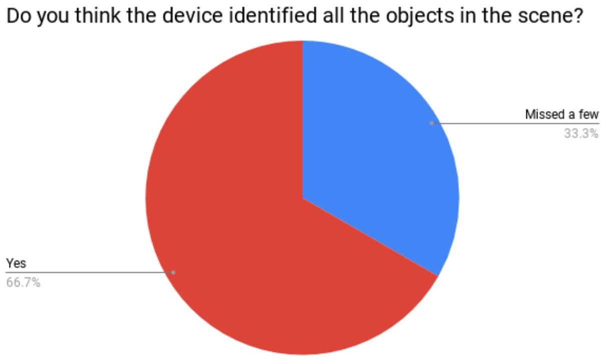


FIGURE 19. **Hypothesis 3 Question 1.**

BVIP students and teachers at the **Roman and Catherine Lobo School for the Visually Impaired, Mangalore, Karnataka, India**.

A. HYPOTHESIS 1: USER TRAINING PERIOD IS MINIMAL

In order to evaluate the first hypothesis, a description of the proposed application was given to the users, along with an explanation of how to use the device. After trying the application, the BVIP users were asked the questions shown in Figs. 15, 12 and 16. As can be seen from the responses in the graphs, most of the users found the application easy to use and were able to use the application effectively after a single walk-through.

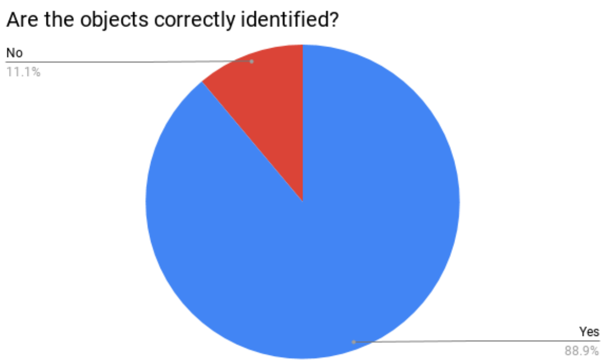


FIGURE 20. **Hypothesis 3 Question 3.**

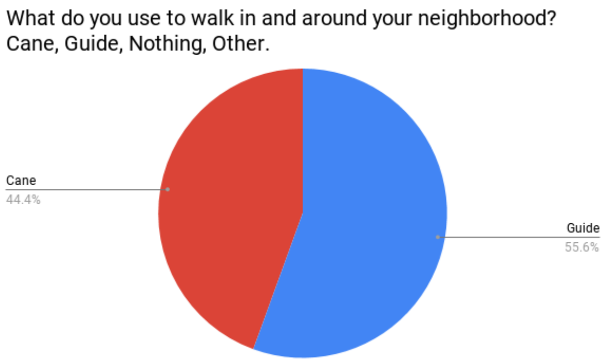


FIGURE 21. **Hypothesis 4 Question 1.**

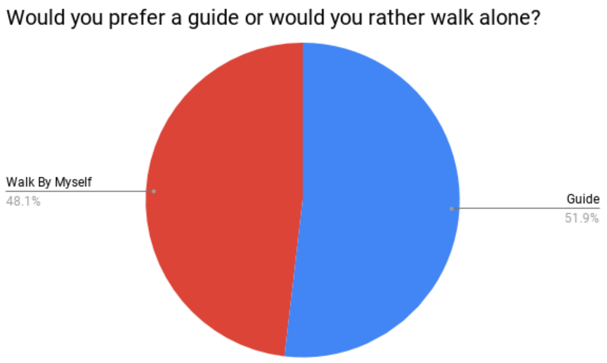


FIGURE 22. **Hypothesis 4 Question 2.**

B. HYPOTHESIS 2: NO EXTRA EFFORT IS REQUIRED TO USE THIS DEVICE ON A DAILY BASIS

The second set of questions were asked to determine if the users found the device to be usable on a daily basis. For this, the questions shown in Figs. 17, 13 and 18 were asked. Most of the users were comfortable using the device regularly, but a few of them found the device irritating as overusing the application sometimes led to the device's heating. The voice recognition system provided by Google Glass was effective except for a few cases where the users had to repeat the commands a few times for the device to recognize the command.

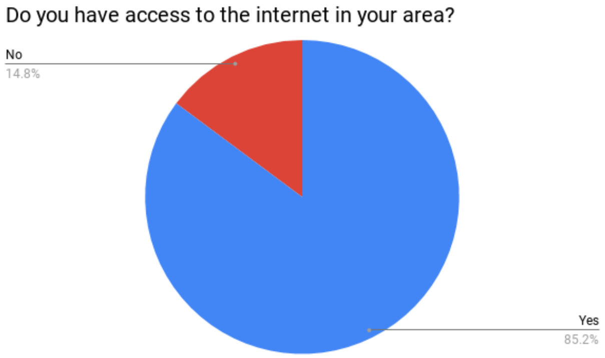


FIGURE 23. Hypothesis 4 Question 3.

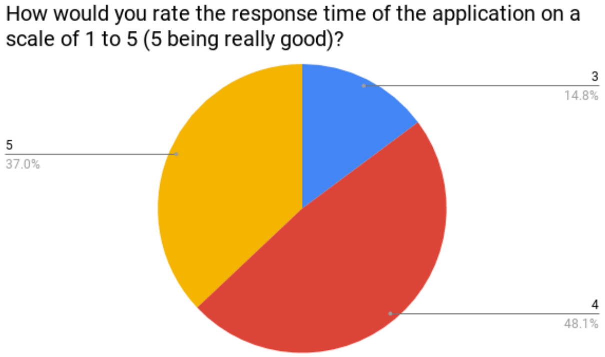


FIGURE 24. Hypothesis 4 Question 4.

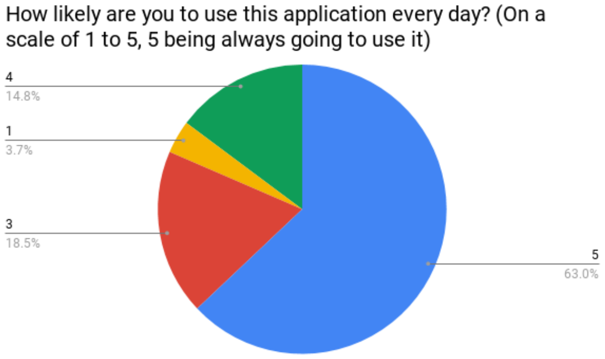


FIGURE 25. Hypothesis 4 Question 5.

C. HYPOTHESIS 3: THE APPLICATION HELPS THE USER TO UNDERSTAND THE SCENE

The third set of questions were focused on the actual use case of the application: Scene Description. The questions asked are as shown in Figs. 19, 14 and 20. As can be seen from the responses displayed in the charts, most of the BVIP users agreed that the objects identified helped them better understand the scene.

D. HYPOTHESIS 4

Null Hypothesis: A visually impaired person would not prefer to use the application.

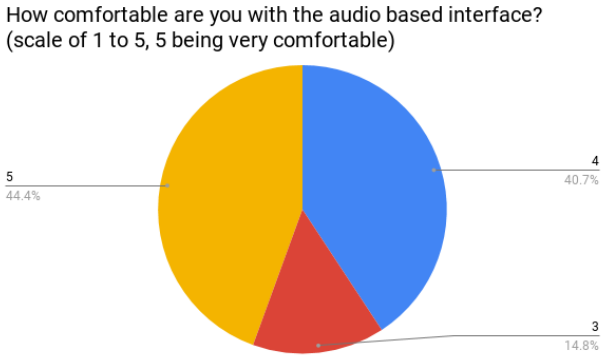


FIGURE 26. Hypothesis 4 Question 6.

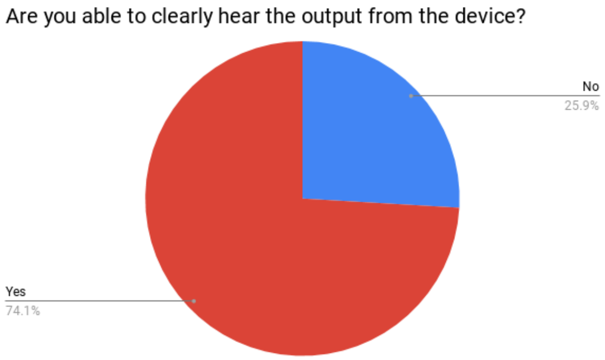


FIGURE 27. Hypothesis 4 Question 7.

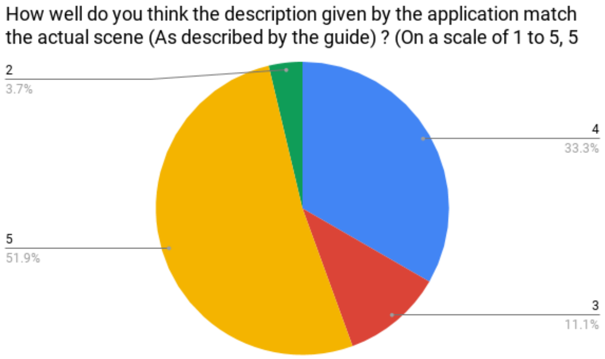


FIGURE 28. Hypothesis 4 Question 8.

Alternate Hypothesis: A visually impaired person would prefer to use this application every day.

The final set of questions were asked to determine if the visually impaired person would prefer to use the application. Various questions were asked to evaluate this hypothesis as can be seen from Figs. 21 to 29. The questions were asked to determine the current lifestyle of the visually impaired individuals and if the use of the application would help them in better scene analysis. From their responses, it can be concluded that the majority of the users found the application effective, portable, and easy to use.

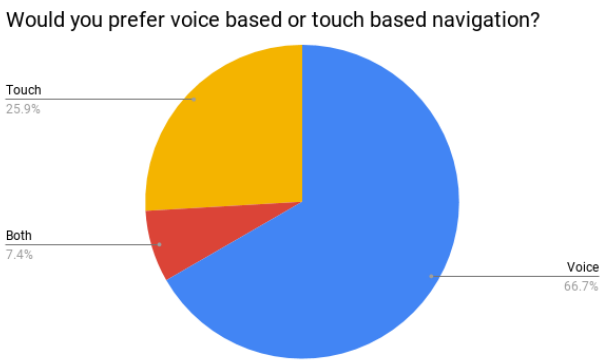


FIGURE 29. Hypothesis 4 Question 9.

REFERENCES

[1] E. Jensen, *Brain-Based Learning: The New Paradigm of Teaching*. Corwin Press, 2008.

[2] A. Berger, A. Vokalova, F. Maly, and P. Poulova, "Google glass used as assistive technology its utilization for blind and visually impaired people," in *Proc. Int. Conf. Mobile Web Inf. Syst.* Cham, Switzerland: Springer, Aug. 2017, pp. 70–82.

[3] C. Van Lansingh and K. A. Eckert, "VISION 2020: The right to sight in 7 years?" *Med. Hypothesis, Discovery Innov. Ophthalmol.*, vol. 2, no. 2, p. 26, 2013.

[4] N. A. Bradley and M. D. Dunlop, "An experimental investigation into wayfinding directions for visually impaired people," *Pers. Ubiquitous Comput.*, vol. 9, no. 6, pp. 395–403, Nov. 2005.

[5] F. Battaglia and G. Iannizzotto, "An open architecture to develop a handheld device for helping visually impaired people," *IEEE Trans. Consum. Electron.*, vol. 58, no. 3, pp. 1086–1093, Aug. 2012.

[6] M. E. Meza-de-Luna, J. R. Terven, B. Raducanu, and J. Salas, "A social-aware assistant to support individuals with visual impairments during social interaction: A systematic requirements analysis," *Int. J. Hum.-Comput. Stud.*, vol. 122, pp. 50–60, Feb. 2019.

[7] W.-J. Chang, L.-B. Chen, C.-H. Hsu, J.-H. Chen, T.-C. Yang, and C.-P. Lin, "MedGlasses: A wearable smart-glasses-based drug pill recognition system using deep learning for visually impaired chronic patients," *IEEE Access*, vol. 8, pp. 17013–17024, 2020.

[8] W.-J. Chang, Y.-X. Yu, J.-H. Chen, Z.-Y. Zhang, S.-J. Ko, T.-H. Yang, C.-H. Hsu, L.-B. Chen, and M.-C. Chen, "A deep learning based wearable medicines recognition system for visually impaired people," in *Proc. IEEE Int. Conf. Artif. Intell. Circuits Syst. (AICAS)*, Mar. 2019, pp. 207–208.

[9] W.-J. Chang, L.-B. Chen, C.-H. Hsu, C.-P. Lin, and T.-C. Yang, "A deep learning-based intelligent medicine recognition system for chronic patients," *IEEE Access*, vol. 7, pp. 44441–44458, 2019.

[10] P. A. Zientara, S. Lee, G. H. Smith, R. Brenner, L. Itti, M. B. Rosson, J. M. Carroll, K. M. Irick, and V. Narayanan, "Third eye: A shopping assistant for the visually impaired," *Computer*, vol. 50, no. 2, pp. 16–24, Feb. 2017.

[11] D. Pintado, V. Sanchez, E. Adarve, M. Mata, Z. Gogebakan, B. Cabuk, C. Chiu, J. Zhan, L. Gewali, and P. Oh, "Deep learning based shopping assistant for the visually impaired," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–6.

[12] P.-A. Quinones, T. Greene, R. Yang, and M. Newman, "Supporting visually impaired navigation: A needs-finding study," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, May 2011, pp. 1645–1650.

[13] F. E.-Z. El-Taher, A. Taha, J. Courtney, and S. Mckeever, "A systematic review of urban navigation systems for visually impaired people," *Sensors*, vol. 21, no. 9, p. 3103, Apr. 2021.

[14] J.-H. Lee, D. Kim, and B.-S. Shin, "A wearable guidance system with interactive user interface for persons with visual impairment," *Multimedia Tools Appl.*, vol. 75, no. 23, pp. 15275–15296, Dec. 2016.

[15] R. Tapu, B. Mocanu, and T. Zaharia, "A computer vision-based perception system for visually impaired," *Multimedia Tools Appl.*, vol. 76, no. 9, pp. 11771–11807, May 2017.

[16] P. Vyavahare and S. Habeeb, "Assistant for visually impaired using computer vision," in *Proc. 1st Int. Conf. Adv. Res. Eng. Sci. (ARES)*, Jun. 2018, pp. 1–7.

[17] K. Laubhan, M. Trent, B. Root, A. Abdelgawad, and K. Yelamarthi, "A wearable portable electronic travel aid for blind," in *Proc. Int. Conf. Electr., Electron., Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 1999–2003.

[18] M. Trent, A. Abdelgawad, and K. Yelamarthi, "A smart wearable navigation system for visually impaired," in *Proc. Int. Conf. Smart Objects Technol. Social Good*, Jul. 2017, pp. 333–341.

[19] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Smart guiding glasses for visually impaired people in indoor environment," *IEEE Trans. Consum. Electron.*, vol. 63, no. 3, pp. 258–266, Aug. 2017.

[20] Q.-H. Nguyen, H. Vu, T.-H. Tran, and Q.-H. Nguyen, "Developing a way-finding system on mobile robot assisting visually impaired people in an indoor environment," *Multimedia Tools Appl.*, vol. 76, no. 2, pp. 2645–2669, Jan. 2017.

[21] M. Avila and T. Kubitza, "Assistive wearable technology for visually impaired," in *Proc. 17th Int. Conf. Hum.-Comput. Interact. Mobile Devices Services Adjunct*, Aug. 2015, pp. 940–943.

[22] J. van der Bie, B. Visser, J. Matsari, M. Singh, T. van Hasselt, J. Koopman, and B. Kröse, "Guiding the visually impaired through the environment with beacons," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, Sep. 2016, pp. 385–388.

[23] J. Guerreiro, D. Sato, D. Ahmetovic, E. Ohn-Bar, K. M. Kitani, and C. Asakawa, "Virtual navigation for blind people: Transferring route knowledge to the real-world," *Int. J. Hum.-Comput. Stud.*, vol. 135, Mar. 2020, Art. no. 102369.

[24] R.-G. Lupu, O. Mitruţ, A. Stan, F. Ungureanu, K. Kalimeri, and A. Moldoveanu, "Cognitive and affective assessment of navigation and mobility tasks for the visually impaired via electroencephalography and behavioral signals," *Sensors*, vol. 20, no. 20, p. 5821, Oct. 2020.

[25] W.-J. Chang, L.-B. Chen, C.-Y. Sie, and C.-H. Yang, "An artificial intelligence edge computing-based assistive system for visually impaired pedestrian safety at zebra crossings," *IEEE Trans. Consum. Electron.*, vol. 67, no. 1, pp. 3–11, Feb. 2021.

[26] H. Ye, M. Malu, U. Oh, and L. Findlater, "Current and future mobile and wearable device use by people with visual impairments," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, Apr. 2014, pp. 3123–3132.

[27] F. Pégeot and H. Goto, "Scene text detection and tracking for a camera-equipped wearable reading assistant for the blind," in *Proc. Asian Conf. Comput. Vis.*, vol. 7729, Nov. 2012, pp. 454–463.

[28] L. González-Delgado, L. Serpa-Andrade, K. Calle-Urgilez, A. Guzhnay-Lucero, V. Robles-Bykbaev, and M. Mena-Salcedo, "A low-cost wearable support system for visually disabled people," in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Nov. 2016, pp. 1–5.

[29] A. Memo and P. Zanuttigh, "Head-mounted gesture controlled interface for human-computer interaction," *Multimedia Tools Appl.*, vol. 77, no. 1, pp. 27–53, Jan. 2018.

[30] M. Barney, J. Kilner, G. Brito, A. Araájo, and M. Nogueira, "Sensory glasses for the visually impaired," in *Proc. 14th Int. Web Conf.*, Apr. 2017, pp. 1–2.

[31] M. A. K. Shishir, S. R. Fahim, F. M. Habib, and T. Farah, "Eye assistant: Using mobile application to help the visually impaired," in *Proc. 1st Int. Conf. Adv. Sci., Eng. Robot. Technol. (ICASERT)*, May 2019, pp. 1–4.

[32] B. Jiang, J. Yang, Z. Lv, and H. Song, "Wearable vision assistance system based on binocular sensors for visually impaired users," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1375–1383, Apr. 2019.

[33] O. Bogdan, O. Yurchenko, O. Bailo, F. Rameau, D. Yoo, and I. Kweon, "Intelligent assistant for people with low vision abilities," in *Proc. Pacific-Rim Symp. Image Video Technol.*, Feb. 2018, pp. 448–462.

[34] S.-C. Pei and Y.-Y. Wang, "Census-based vision for auditory depth images and speech navigation of visually impaired users," *IEEE Trans. Consum. Electron.*, vol. 57, no. 4, pp. 1883–1890, Nov. 2011.

[35] W.-J. Chang, L.-B. Chen, M.-C. Chen, J.-P. Su, C.-Y. Sie, and C.-H. Yang, "Design and implementation of an intelligent assistive system for visually impaired people for aerial obstacle avoidance and fall detection," *IEEE Sensors J.*, vol. 20, no. 17, pp. 10199–10210, Sep. 2020.

[36] L.-B. Chen, J.-P. Su, M.-C. Chen, W.-J. Chang, C.-H. Yang, and C.-Y. Sie, "An implementation of an intelligent assistance system for visually impaired/blind people," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2019, pp. 1–2.

[37] W.-J. Chang, L.-B. Chen, and Y.-Z. Chiou, "Design and implementation of a drowsiness-fatigue-detection system based on wearable smart glasses to increase road safety," *IEEE Trans. Consum. Electron.*, vol. 64, no. 4, pp. 461–469, Nov. 2018.

[38] L.-B. Chen, W.-J. Chang, J.-P. Su, J.-Y. Ciou, Y.-J. Ciou, C.-C. Kuo, and K. S.-M. Li, "A wearable-glasses-based drowsiness-fatigue-detection system for improving road safety," in *Proc. IEEE 5th Global Conf. Consum. Electron.*, Oct. 2016, pp. 1–2.

[39] H. Jiang, J. Starkman, M. Liu, and M.-C. Huang, "Food nutrition visualization on Google glass: Design tradeoff and field evaluation," *IEEE Consum. Electron. Mag.*, vol. 7, no. 3, pp. 21–31, May 2018.

[40] Y. Li and A. Sheopuri, "Applying image analysis to assess food aesthetics and uniqueness," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 311–314.

[41] P. Washington, C. Voss, N. Haber, S. Tanaka, J. Daniels, C. Feinstein, T. Winograd, and D. Wall, "A wearable social interaction aid for children with autism," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst.*, May 2016, pp. 2348–2354.

[42] P. Washington, C. Voss, A. Kline, N. Haber, J. Daniels, A. Fazel, T. De, C. Feinstein, T. Winograd, and D. Wall, "SuperpowerGlass: A wearable aid for the at-home therapy of children with autism," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 3, pp. 1–22, Sep. 2017.

[43] Z. Lv, A. Halawani, S. Feng, S. U. Réhman, and H. Li, "Touch-less interactive augmented reality game on vision-based wearable device," *Pers. Ubiquitous Comput.*, vol. 19, nos. 3–4, pp. 551–567, Jul. 2015.

[44] Z. Wang, X. Wen, Y. Song, X. Mao, W. Li, and G. Chen, "Navigation of a humanoid robot via head gestures based on global and local live videos on Google glass," in *Proc. IEEE Int. Instrum. Meas. Technol. Conf. (I2MTC)*, May 2017, pp. 1–6.

[45] X. Wen, Y. Song, W. Li, G. Chen, and B. Xian, "Rotation vector sensor-based remote control of a humanoid robot through a Google glass," in *Proc. IEEE 14th Int. Workshop Adv. Motion Control (AMC)*, Apr. 2016, pp. 203–207.

[46] C. F. Xu, Y. F. Gong, W. Su, J. Cao, and F. B. Tao, "Virtual video and real-time data demonstration for smart substation inspection based on Google glasses," in *Proc. Int. Conf. Renew. Power Gener.*, Jan. 2015, p. 5.

[47] A. Widmer, R. Schaer, D. Markonis, and H. Müller, "Facilitating medical information search using Google glass connected to a content-based medical image retrieval system," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 4507–4510.

[48] G. Lausegger, M. Spitzer, and M. Ebner, "OmniColor—A smart glasses app to support colorblind people," *Int. J. Interact. Mobile Technol. (iJIM)*, vol. 11, pp. 161–177, Jul. 2017.

[49] A. I. Anam, S. Alam, and M. Yeasin, "Expression: A dyadic conversation aid using Google glass for people with visual impairments," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, Sep. 2014, pp. 211–214.

[50] A. D. Hwang and E. Peli, "An augmented-reality edge enhancement application for Google glass," *Optometry Vis. Sci.*, vol. 91, no. 8, pp. 1021–1030, 2014.

[51] L. B. Neto, F. Grijalva, V. R. M. L. Maike, L. C. Martini, D. Florencio, M. C. C. Baranauskas, A. Rocha, and S. Goldenstein, "A kinect-based wearable face recognition system to aid visually impaired users," *IEEE Trans. Human-Mach. Syst.*, vol. 47, no. 1, pp. 52–64, Feb. 2017.

[52] H. Takizawa, S. Yamaguchi, M. Aoyagi, N. Ezaki, and S. Mizuno, "Kinect cane: An assistive system for the visually impaired based on three-dimensional object recognition," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2012, pp. 740–745.

[53] D. Kim and Y. Choi, "Applications of smart glasses in applied sciences: A systematic review," *Appl. Sci.*, vol. 11, no. 11, p. 4956, May 2021.

[54] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, 2013.

[55] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: Common objects in context," 2015, *arXiv:1405.0312*.

[56] J. Whitehill, Z. Serpell, Y.-C. Lin, A. Foster, and J. R. Movellan, "The faces of engagement: Automatic recognition of student engagement from facial expressions," *IEEE Trans. Affect. Comput.*, vol. 5, no. 1, pp. 86–98, Jan./Mar. 2014.

[57] T. S. Ashwin and R. M. R. Guddeti, "Affective database for e-learning and classroom environments using Indian students' faces, hand gestures and body postures," *Future Gener. Comput. Syst.*, vol. 108, pp. 334–348, Jul. 2020.

[58] T. S. Ashwin and R. M. R. Guddeti, "Automatic detection of students' affective states in classroom environment using hybrid convolutional neural networks," *Educ. Inf. Technol.*, vol. 25, no. 2, pp. 1387–1415, Mar. 2020.

[59] T. S. Ashwin and R. M. R. Guddeti, "Unobtrusive behavioral analysis of students in classroom environment using non-verbal cues," *IEEE Access*, vol. 7, pp. 150693–150709, 2019.

[60] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[61] S. Chen, S. Saiki, and M. Nakamura, "Toward flexible and efficient home context sensing: Capability evaluation and verification of image-based cognitive APIs," *Sensors*, vol. 20, no. 5, p. 1442, Mar. 2020.

HAFEEZ ALI A. received the B.Tech. degree in information technology from the National Institute of Technology Karnataka Surathkal, India, in 2019. His research interests include distributed computing, data analytics, pattern recognition, computer vision, computer networks, and software engineering.

SANJEEV U. RAO received the B.Tech. degree in information technology from the National Institute of Technology Karnataka Surathkal, India, in 2019. His research interests include deep learning, big data, and the Internet of Things.

SWAROOP RANGANATH received the B.Tech. degree in information technology from the National Institute of Technology Karnataka Surathkal, India, in 2019. His research interests include deep learning, reinforcement learning, AI applications in IoT systems, advanced analytics in business insights, and explainable AI.

T. S. ASHWIN (Member, IEEE) received the B.E. degree from Visvesvaraya Technological University, Belgaum, India, in 2011, the M.Tech. degree from Manipal University, Manipal, India, in 2013, and the Ph.D. degree from the National Institute of Technology Karnataka Surathkal, Mangalore, India. He has more than 35 reputed and peer-reviewed international conferences and journal publications, including five book chapters. His research interests include affective computing, human–computer interaction, educational data mining, learning analytics, and computer vision applications. He is a member of AAAC, ComSoc, and ACM.

GUDDETI RAM MOHANA REDDY (Senior Member, IEEE) received the B.Tech. degree from S.V. University, Tirupati, Andhra Pradesh, India, in 1987, the M.Tech. degree from the Indian Institute of Technology Kharagpur, India, in 1993, and the Ph.D. degree from The University of Edinburgh, U.K., in 2005. Currently, he is a Senior Professor with the Department of Information Technology, National Institute of Technology Karnataka Surathkal, Mangalore, India. He has more than 200 research publications in reputed/peer-reviewed international journals, conference proceedings, and book chapters. His research interests include affective computing, big data, cognitive analytics, bio-inspired cloud and green computing, the Internet of Things and smart sensor networks, social multimedia, and social network analysis. He is a Senior Member of ACM; a Life Fellow of IETE, India; and a Life Member of ISTE, India, and Computer Society of India.