EYERIS: A Virtual Eye to Aid the Visually Impaired

Jinesh A Shah
Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
jineshah5@gmail.com

Aashreen Raorane
Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
aashreenraorane@gmail.com

Akash Ramani
Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
akashramani98.ar@gmail.com

Hitanshu Rami
Computer Science
Illinois Institute of Technology
Chicago,United States of America
hrami@hawk.iit.edu

Narendra Shekokar
Computer Engineering
D J Sanghvi College of Engineering
Mumbai, India
narendra.shekokar@djsce.ac.in

Abstract — **With just little aid to the visually impaired people existing currently, there is a need to implement a device that aids them in their daily activities. The existing systems such as Screen Reading software and Braille devices help visually impaired people to read and have access to various gadgets but these technologies become useless when the blind wants to carry out basic tasks, which involve recognizing the scene in front of him, for instance, the people or objects. Further, little devices exist that facilitate communication between a blind and deaf-dumb. Our system will facilitate around 285 million visually impaired people around the globe. Eyeris is made with an aim to aid a person with total or partially impaired eyesight without the need of a guardian. The configuration of hardware and software units is to be designed, which leads to detecting and recognizing objects, people and signs in the frame of vision and notifying the user about the same. We present a method, which uses object detection and facial recognition on the live stream of images. To enable communication between a blind and deaf person, American Sign Language (ASL) is also to be detected and recognized. The resultant object or person or sign is then transmitted to the impaired person in the form of audio. The guardian can be notified via a text message when the person needs help or is in danger. The goal is to provide inexpensive solution to the visually impaired and make their life better and self-sufficient.**

Keywords — Real time Object Detection; Facial Recognition; SSD; Deep Residual Learning; Raspberry Pi; Text to speech; Visually Impaired; Sign Language Recognition

I. INTRODUCTION

"EyeRis" is the device that is being developed for the people who are visually impaired. There are about 285 million people in the world diagnosed with some form of visual impairment [1]. Life for visually impaired is difficult and they rely on others to tell them about their surroundings. They are not able to experience the world the way we do. In day to day, it is very difficult for them to carry out basic tasks. Existing systems only target activities like reading and writing [2]. "EyeRis" will make them free of bounds faced by them in performing daily activities so that they can live an independent life.

EyeRis uses Pi Camera module connected with Raspberry Pi 3 Model B+ to achieve the desired solution proposed. EyeRis has the Pi-camera placed in the middle of the palm and also as a source for live video stream. The stream captured by the camera is broken down into frames. Each frame is passed through our single deep neural network which draws approximate bounding boxes of different aspect ratios [3]. Then, depending on the prediction score of each bounding box, the best fit box is selected for the object [3]. For Facial Recognition, each box is analyzed against increasing set of facial features and if it fails for one feature, that box is discarded and declared as not a face region [4]. If it is found to be a face region then it is sent to a residual network to compare it with the database to see if it recognizes the face or not [5]. The result of both object detection and facial recognition is translated into audio using EspeakNG for the blind person to hear. EyeRis will notify the guardian of the user, in case of any danger and the guardian will be able to understand the situation from anywhere around the globe on his phone or any screen.

Object Detection is achieved by training the model using SSD (Single Shot MultiBox Detector) that eliminates proposal generation and encapsulates all computation in a single network [6]. Our network architecture for face recognition is based on ResNet-34 from the Deep Residual Learning for Image Recognition but with fewer layers and the number of filters reduced to half [7]. The object detection model was trained using the COCO Dataset which is a large-scale object detection, segmentation, and captioning dataset [8].

EyeRis aims to achieve portability and easy user interface. EyeRis divides itself into major aspects like object detection, face recognition, sign language recognition, full duplex video communication and text to speech synthesis.

II. EXISTING SYSTEM

The paper "Oculus" [9], focusses on enabling the visually impaired to self-navigate themselves without the requirement of a third person. It calculates proximity of the object nearby, thereby guiding the path. This paper provided functionality in terms of user navigation without incorporating a response to any dangerous or emergency situations. In 2014, a paper named "Smart Vision System for Blind" [1] objected to support blind users with the features of movement within unacquainted surrounding. Another paper [10] used different individual device portions for indoor and outdoor movements as well as incorporated GPS to track the coordinates of the position place of the user. Also, there are smartphone based guiding systems[11] with obstacle identification and multiple modes for convenient user interfacing modes.

Processing texts from real-time image [12] is also a breakthrough in the field of Computer Vision. In 2017, face recognition from comparing with the database was also an appreciated paper in the IEEE paper conference [13]. Face detection, face recognition and audio output module were merged in a portable and wearable device. OpenCV has been a first-rate approach for image processing and gives a tranquil solve to our stated problem. In a paper [14], images are

978-1-7281-2339-4/20/$31.00 ©2020 IEEE

classified using OpenCV libraries and AdaBoost algorithm. The paper "Real-Time Hand Gesture Recognition Using Finger Segmentation" [15] proposes a four-step architecture for hand gesture recognition. The background subtraction method is used for hand detection. The fingers and palm are then split to distinguish and recognize the fingers from the entire hand. Using a simple rule classifier, the hand gestures can be classified and then matched with the existing image in order to recognize the hand gestures. The paper highlights an issue wherein the hand gestures are not efficiently recognized if the background of the stream is similar to skin colour. Moreover, it does not use any machine learning algorithms for gesture recognition, which tend to perform better. Another publication [16] also used the OpenCV application and emphasized on the object detection based on hue, saturation and colour value (HSV) range.

With the flow of technology advancement, more efficient algorithms, methods and systems are proposed and developed for robust service as well as real time detection with minimum delay and maximum accuracy.

The algorithm of this paper is SSD (Single Shot Multibox Detector) which is efficient for real-time robust system development. Here the processing server is separate from the input device. There has to be a connection over the cellular device and the server-side database which has to be on online mode for maximum continuous facility.

These works were inspirational and prodigious guidance for our target solution and marked the hindrance that we might face on our progress so our effort started based on those drawbacks to come up with more progressive solution and develop a smart system to support the blind people. None of the above works covered the situations of emergency in the project which is of utmost importance to the visually impaired. Our prototype will not only admonish the visually impaired individual of the danger in the surrounding but also alert their guardian about the same. Thus, allowing the guardian not to be physically present but still guide the visually impaired through Real Time Communication.

III. WORKING

The process takes place according to the following steps:
1. Capturing the image: Continuous stream of images are captured in real time with the help of pi camera.
2. Detecting the objects: A deep learning SSD model is trained on the dataset containing different objects. During this stage, a bounding box is drawn around the detected object.
3. Informing the user: This stage uses text to speech recognition [17]. User is informed about the object detected in his/her frame vision in the form of audio.
4. Danger Alert: In case, if the user comes across a dangerous object such as a fire, an alert message will be sent to the guardian informing that the person is in danger.
5. Real Time Facial Recognition: In addition to the above listed, the prototype also has facial recognition feature that helps the user to identify individuals in the dataset and output their name in the form of audio.
6. Sign Language Recognition: The signs generated by a deaf person will be detected and matched with the American Sign Language (ASL) database and if recognized, the alphabet will be pronounced [18].

IV. OBJECT DETECTION

Object Detection is probably one of the most promising areas in the field of computer vision. It refers to the capability of the computer system to locate and identify instances of objects [10]. The goal is to detect all the instances of the object from the known class. In object detection algorithms we try to draw a bounding box around the detected object. The evolution of deep learning brought into existence various detection algorithms such as R-CNN, Fast-RCNN, Faster-RCNN, RetinaNet and fast yet highly accurate ones like SSD and YOLO [19]. In our system, we have used SSD.

A. Single Shot Detection SSD

SSD is designed as shown in Fig. 1 for object detection in real-time. It consists of two parts: 1) Extracting feature maps 2) Applying convolutional filters to detect objects.

The architecture of SSD is built on the prominent VGG-16. To enable extraction of features at multiple scales, reduce the number of parameters and to decrease the input size at every subsequent layers, a set of auxiliary convolution layers are used instead of the original VGG fully connected layers. The reverence for VGG-16 and it being selected for the base network comes from its great strength in image classification problems [3]. Also, it is highly used in places where transfer learning helps to ameliorate the outcomes [6].

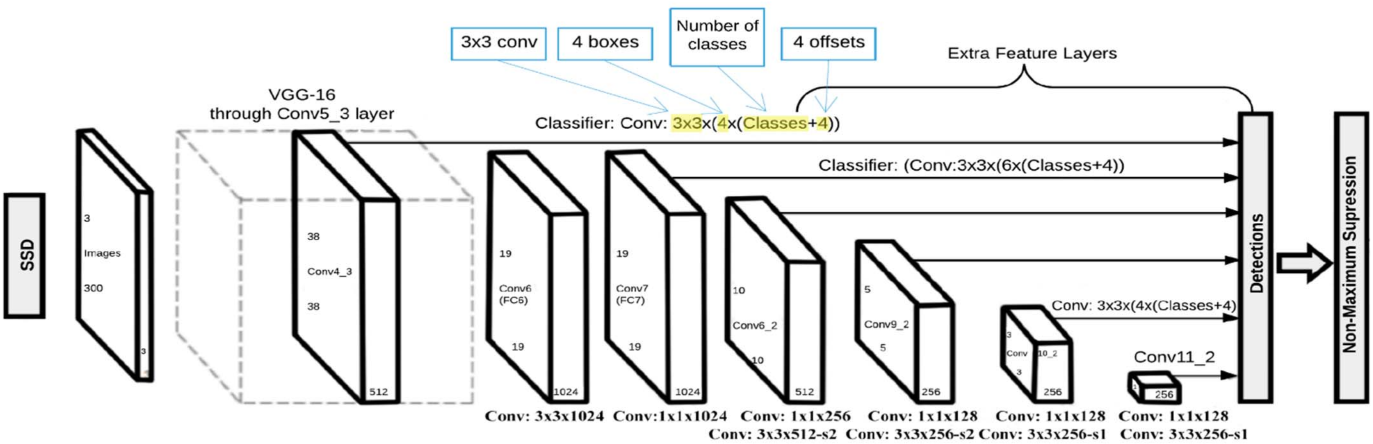

Fig 1. SSD Architectural Diagram [3]

B. COCO Dataset

COCO stands for Common Object in Context. With the goal of proceeding the state-of-the-art and with an aim to facilitate the studies of scene understandings, Microsoft created this diverse dataset. The labeling of objects is done using instance segmentation which labels every instance of the object in every segmentation [8]. Fig. 2 shows some images of COCO dataset. COCO dataset contains more than 1.5 million instances of objects. Instances are evenly distributed among object classes. In paper [20] shows graphical representation of various Dataset Statistics and Number of Instances per object category for analysis of MS COCO and other datasets.

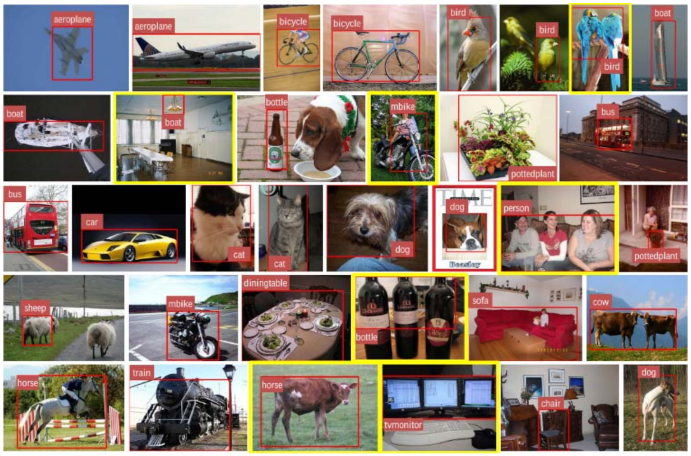

Fig 2. Objects of COCO Dataset [8]

Fig. 3 shows the project flow for object detection. The user wears the device. The system then detects object using the designed algorithm. If any harmful object is detected by the system, an SMS is sent to the guardian along with the location of the user. The guardian can then access Live Feed of the user in order to guide the user further. All the objects that are detected are notified to the user in form of an Audio output.

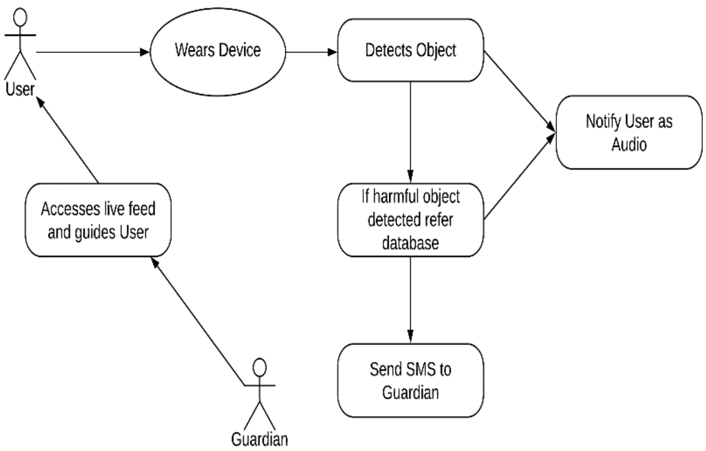

Fig 3. Project Flow – Object detection

V. FACE RECOGNITION

Face Recognition refers to the detection and identification of the face automatically by computerized systems by looking at the face [5]. Identifying a person with an image has been popularized through the mass media. Although, it is less robust to fingerprint or retina scanning, it recognizes individuals at a distance even without having them touch anything. This prototype uses the Open-Computer-Vision (OpenCV) library and ResNet-34 for facial detection and recognition. OpenCV was designed for computational efficiency and with a strong focus on real-time applications [21]. So, it's perfect for real-time face recognition using a camera. Fig. 4 lists and explains the phases involved in facial recognition. The .yml file after phase 2 is created to match face ids from the dataset to their corresponding user names for real-time detection. Additional information about the users can also be stored in this file. These names act as audio output of the system after undergoing text-to-speech conversion.

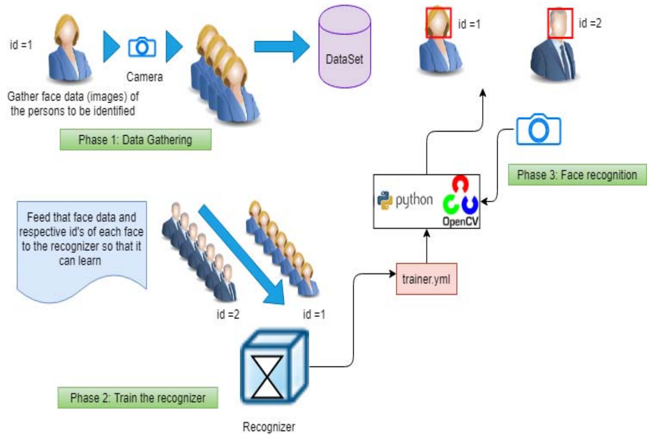

Fig 4. Phases of Facial Recognition

A. ResNet-34

Our network architecture, for face recognition as shown below in Fig. 5 is based on ResNet-34 model. Since deeper neural networks are more difficult to train, we present a residual learning framework to ease the training of networks. One of the drawbacks of VGG is that its performance of generalization degrades as the layers gets deeper so it could not be used for accurate facial recognition [22]. The reason why we use the ResNet model for recognition of faces is due to the overall residual network has very low rate of error (approximately 3.57%). The Recognition problem can be solved by using the Residual Learning Framework.

Let H(x) be the mapping function being used in the layers and let x be the input to the first layer. Based on the belief that neural network can work as function approximators it would not be wrong to say that they will be able to be a solution for identity function such that $\mathrm{F}(\mathrm{x}) = \mathrm{x}$ [22]. As shown by Yuan X, and Yahagi T multiple non-linear layers can be used to approximate complicated functions asymptotically, we can use multiple non-linear layers to approximate our desired residual function [23]. So, this would mean that if consider x as the input of the first layer, it can be bypassed to be the output of the last layer such that $\mathrm{H}(\mathrm{x}) = \mathrm{F}(\mathrm{x}) + x$ [6]. This allows ResNet to pass through the skip connections backwards from later layers to the commencing layers directly [22]. The building block of Resnet can be defined as shown in the Fig 5: $\mathrm{y} = \mathrm{F}\,(\mathrm{x}, [\text{Weight of the layer}]) + \mathrm{x}$ where F (x, [Weight of the layer]) is representing the learning of the residual network.

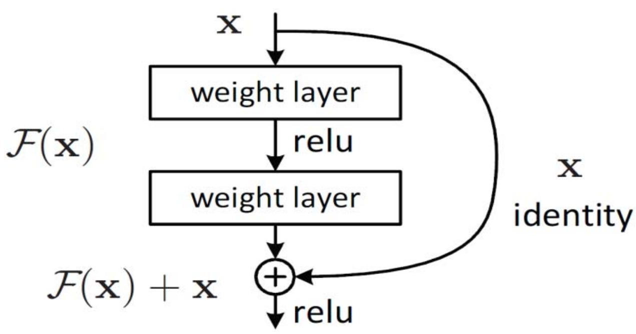

Fig 5. ResNet-34 architecture [23]

The advantages of using this model are:

1. The Residual Neural Network is deep enough. So, it facilitates the extraction of all abstract, complex and easily distinguishable features and can also map more complex mapping functions [24].
2. The Residual Network is meant such that the GPU on the training and testing can be feasible.
3. The use of Residual Network helps to achieve high precision results [24].

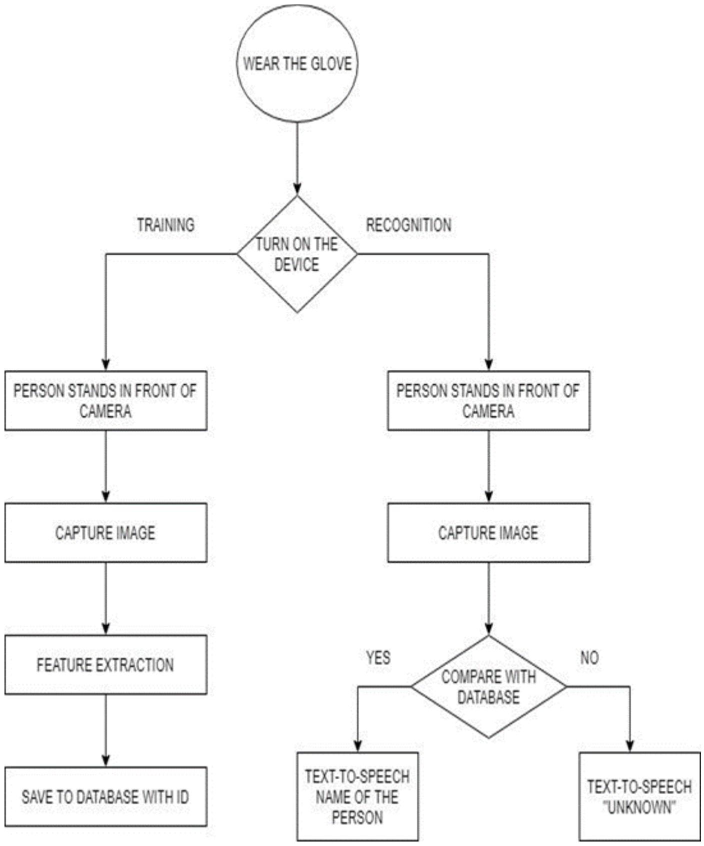

Fig 6. Flowchart for Facial Recognition

The above Fig.6 shows the project flow for Facial Recognition. The process takes place in two phases, training and recognition. In training, images of the person's face are added to the dataset. Feature extraction is performed on the images and they are saved into the database. In recognition phase, the video is captured from camera which is them converted to image frame; the model then uses the database and the knowledge learned by it to recognize the face [5]. If the face is recognized, an output of the person's name is given in audio, else "unknown" is given as audio after converting it from text to speech.

VI. SIGN LANGUAGE RECOGNITION-ARCHITECTURE

The below Fig. 7 shows the architecture. The input image is fed to the CNN. These CNN are composed of layers. The convolution layers are the layers where filters are applied to the original image. The presence of various features and the patterns in an input frame is being used [18].

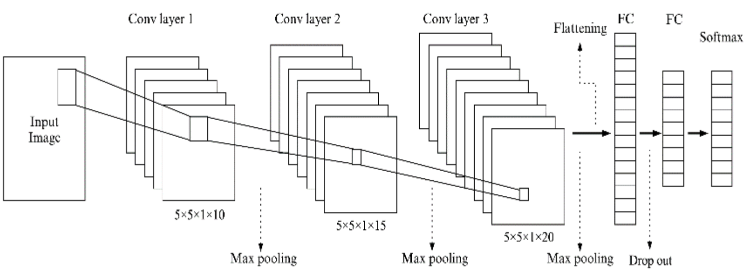

Fig 7. Architecture for Sign Language Recognition

The pooling layer comes after every convolution layer. This layer reduces the number of parameters and computation in our network and thus controlling overfitting by progressive reduction in the size of the spatial network. To avoid overfitting, we take the name states with maximum value from the pool called as max pooling. After the series of convolution layers the Fully connected (FC) layer is encountered. Fully connected layers are placed before the classification output of a CNN and they are used to flatten the results before classification. The neurons are fully connected to the activation functions of the previous layers. Matrix multiplication is used for computing the activations which is followed by an offset that is given by a bias.

SoftMax function, an activation function that turns numbers(logits) into probabilities that sum to one [25]. SoftMax function outputs a vector that represents the probability distributions of a list of potential outcomes. These probabilities specify the probability that the particular sign belongs to a specific class. The class with highest probability is regarded as output class. Hence, SoftMax function helps us to better classify the signs distinctly with ease.

VII.EXPERIMENTATION

We carried out a few experiments on our system to assess the performance and accuracy. Fig.8 and Fig.9 below are the snapshots of the system's performance for object detection and facial recognition respectively.

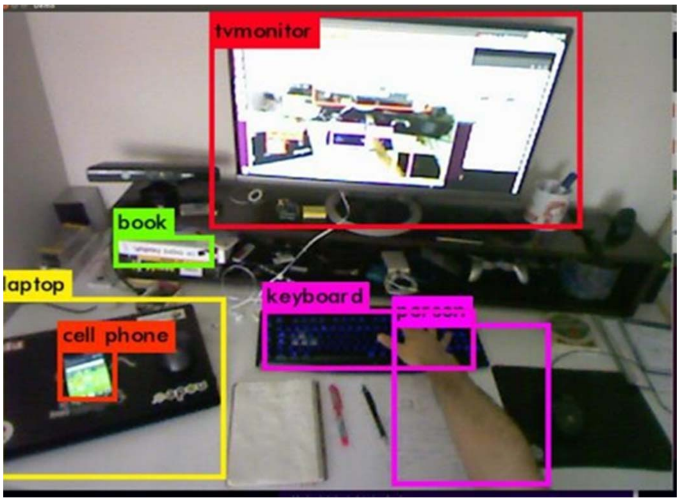

Fig 8. Bounding Boxes around objects

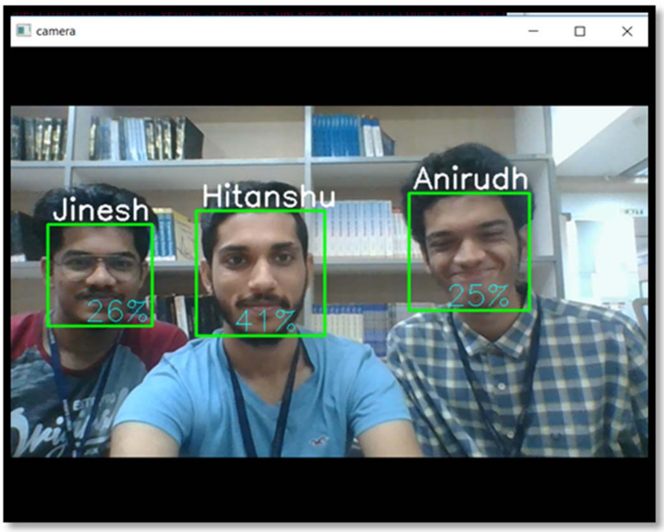

Fig 9. Detection and Recognition of faces

After successfully conducting the experiments, analysis of the results was carried out. To characterize the performance of the system we calculate performance metrics: Accuracy, Precision, Recall and F-score.

A. Analysis for Object Detection

For object detection we define the confusion matrix terms as follows:

True Positive (TP): Model trained on object; system detects object.
False Positive (FP): Model not trained on object; system detects it as something else.
False Negative (FN): Model trained on object; system does not detect object.
True Negative (TN): Model not trained on object; system does not detect object.

The system was examined for real-time detection on 100 objects out of which 80 objects were a part of dataset on which the model was trained. The remaining 20 objects were not a part of the dataset. A total of 10 iterations were performed for each object. The result obtained is represented as shown in Table I. in the form of a confusion matrix.

TABLE I. CONFUSION MATRIX FOR OBJECT DETECTION

Model Detects Object?	Model Trained on Object?	
	True Positive (TP) 763	False Positive (FP) 42
	False Negative (FN) 37	True Negative (TN) 158

The performance metrics of the system can then be evaluated by using the formula:

$$Accuracy = \frac{(TP+TN)}{Total\ number\ of\ objects \times Iterations}$$

$$Accuracy = \frac{(763+158)}{100 \times 10} = \frac{921}{1000} = 92.1\% \quad (1)$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{763}{763+42} = \frac{763}{805} = 0.948 \quad (2)$$

$$Recall = \frac{TP}{(TP+FN)} = \frac{763}{763+37} = \frac{763}{800} = 0.954 \quad (3)$$

$$F-score = \frac{2 \times Recall \times Precision}{(Recall+Precision)} = \frac{2 \times 0.954 \times 0.948}{0.954+0.948} = 0.951 \quad (4)$$

B. Analysis for Facial Recognition

Similarly, for facial recognition we define the confusion matrix terms as follows:

True Positive (TP): Model trained on face; system recognizes face.
False Positive (FP): Model not trained on face; system recognizes him/her as someone else.
False Negative (FN): Model trained on face; system does not recognize face.
True Negative (TN): Model not trained on face; system does not recognize face.

The system was examined for real-time detection on 20 faces out of which 10 faces were a part of dataset on which the model was trained. The remaining 10 faces were not a part of the dataset. A total of 20 iterations were performed on each face. The result obtained is represented as shown in Table II. in a form of confusion matrix.

TABLE II. CONFUSION MATRIX FOR FACIAL RECOGNITION

Model Detects Face?	Model Trained on Face?	
	True Positive (TP) 171	False Positive (FP) 33
	False Negative (FN) 29	True Negative (TN) 167

The performance metrics of the system can then be evaluated by using the formula:

$$Accuracy = \frac{(TP+TN)}{Total\ number\ of\ faces\ \times Iterations}$$

$$Accuracy = \frac{(171+167)}{20 \times 20} = \frac{338}{400} = 84.5\% \quad (5)$$

$$Precision = \frac{TP}{(TP+FP)} = \frac{171}{171+33} = \frac{171}{204} = 0.838 \quad (6)$$

$$Recall = \frac{TP}{(TP+FN)} = \frac{171}{171+29} = \frac{171}{200} = 0.855 \quad (7)$$

$$F-score = \frac{2 \times Recall \times Precision}{(Recall+Precision)} = \frac{2 \times 0.855 \times 0.838}{0.855+0.838} = 0.846 \quad (8)$$

TABLE III. SUMMARY OF PERFORMANCE METRICS

System	Precision	Recall	F-score	Accuracy (%)
Object Detection	0.948	0.954	0.951	92.1
Facial Recognition	0.838	0.855	0.846	84.5

The above Table III. indicates that the object detection system shows a good balance between precision and recall and in turn in F-score. The accuracy of the system is high enough to be used in real-time situations for the fact that model continuously detects the objects and wrong predictions can be easily distinguished by the user if the predictions vary as the object moves closer. On the other hand, task of recognizing faces is far more complex than recognizing objects. The system exhibits a good accuracy of 84.5%. In actual usage, the system might prove to be confusing if the system detects a person it is not trained on. A high value of precision denotes the system, when detects positive, has a higher chance of a correct prediction thus the system can tackle the above issue appropriately. Overall, the system, as it stands, is ready to be used by visually impaired user.

C. Text Message

If the system detects any harmful objects such as fire, a text message is generated which includes the live location of the visually impaired person and is sent to the registered guardian. We implemented the feature and the snapshot of the same is as shown in Fig.10 below.

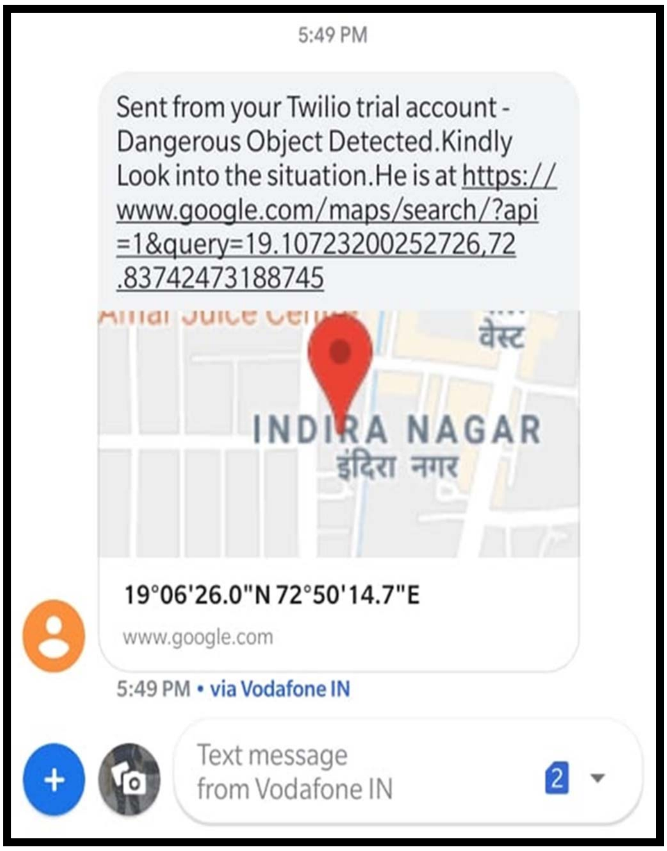

Fig 10. Text message – Link to the live location

VIII.CONCLUSION AND FUTURE SCOPE

We are providing with the basic implementation of our device-EyeRis. The focus is on alleviating the day-to-day difficulties faced by the visually impaired people. EyeRis is able to detect objects that a human interacts with on a daily basis. The challenge was to design a system which will eliminate the need of a guardian to be always present along with the user. After comprehending that completely discarding any human assistance in such situations is inevitable, we resolved on to sending the guardian a text message including the live location of the user if the device senses any dangerous situation. This novel feature not only allows the guardian to be in a constant touch with the user but also frees him/her to carry out personal tasks.

Inferencing, our aim to make the affected person free of human help, may prove to be useful for them and implementable in a real time environment. After obtaining fairly appropriate results above upon experimentation, we are firm in making this into a reality using all the specified hardware.

We also wish to extend our project in the future by providing GPS so that the person can instruct the device where they want to go, and the device will guide the person accordingly. We intend to add a proximity sensor which will tell the blind person how far the object/person is. Also, we can add an inbuilt reading system as well, where we can upload some book on the device and the device can read it. Other than this, making the device lighter and water resistant is also something that we wish to work on.

REFERENCES

[1] "Global data on visual impairment," World Health Organization, 08-Dec-2017.[Online]. Available:https://www.who.int/blindness/publications/globaldata/en/.

[2] P. A. K. S. M. B. Siddharth Pandey, "SMART VISION SYSTEM FOR BLIND", int. jour. eng. com. sci, vol. 3, no. 05, Dec. 2017.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," Computer Vision – ECCV 2016 Lecture Notes in Computer Science, pp. 21–37, 2016.

[4] R. Chellappa, C. Wilson, and S. Sirohey, "Human and machine recognition of faces: a survey," Proceedings of the IEEE, vol. 83, no. 5, pp. 705–741, 1995.

[5] M. D. Kelly, Visual Identification of People by Computer. Stanford, Calif: Stanford University, Computer Science Department, 1970.

[6] E. Forson, "Understanding SSD MultiBox - Real-Time Object Detection In Deep Learning," Medium, 09-Jun-2019. [Online]. Available:https://towardsdatascience.com/understanding-ssd-multibox-real-time-object-detection-in-deep-learning-495ef744fab.

[7] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778.

[8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," Computer Vision – ECCV 2014 Lecture Notes in Computer Science, pp. 740–755, 2014.

[9] D. J. Mistry, V. V. Mukherjee, N. T. Panchal, and A. U. Gawde, "Oculus Vision for Blind," DJ ASCII-18, vol. 2, Jun-2018.

[10] F. Henriques, H. Fernandes, H. Paredes, P. Martins, and J. Barroso, "A prototype for a blind navigation system," 2nd World Conference on Information Technology, Nov. 2011.

[11] B.-S. Lin, C.-C. Lee, and P.-Y. Chiang, "Simple Smartphone-Based Guiding System for Visually Impaired People," Sensors, vol. 17, no. 6, p. 1371, 2017.

[12] M. Rajesh, B. K. Rajan, A. Roy, K. A. Thomas, A. Thomas, T. B. Tharakan, and C. Dinesh, "Text recognition and face detection aid for visually impaired person using Raspberry PI," 2017 International Conference on Circuit ,Power and Computing Technologies (ICCPCT), 2017.

[13] 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 Transactions on Human-Machine Systems, pp. 1–13, 2016.

[14] P. Y. Kumbhar, M. Attaullah, S. Dhere, and S. Hipparagi, "Real Time Face Detection and Tracking Using OpenCV," International Journal for Research in Emerging Science And Technology, vol. 4, no. 4, pp. 39–43, Apr. 2017.

[15] Z.-H. Chen, J.-T. Kim, J. Liang, J. Zhang, and Y.-B. Yuan, "Real-Time Hand Gesture Recognition Using Finger Segmentation," The Scientific World Journal, vol. 2014, pp. 1–9, 2014.

[16] Nidhi, "Image Processing and Object Detection," International Journal of Applied Research, vol. 1, no. 9, pp. 396–399, Jul. 2015.

[17] S. Ferreira, C. Thillou, B. Gosselin, "From Picture to Speech: An Innovative Application for Embedded Environment" in ProRISC 2003, Veldhoven, Netherland, 2003.

[18] S. Madhumitha, N. S. Sathya, P. Pinto, and R. J. Stephan, "A Novel Method for Recognizing Sound of the Silent through Gestures - Sign Language Interpreter," International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 3, no. 5, pp. 9646–9651, May 2014.

[19] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[20] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," Computer Vision – ECCV 2014 Lecture Notes in Computer Science, pp. 740–755, 2014.

[21] M. Naveenkumar and A. Vadivel, "OpenCV for Computer Vision Applications," Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), Mar. 2015.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.

[23] J. Lu, X. Yuan, and T. Yahagi, "A Method of Face Recognition Based on Fuzzy c-Means Clustering and Associated Sub-NNs," IEEE Transactions on Neural Networks, vol. 18, no. 1, pp. 150–160, 2007.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning Research 15 (2014), vol. 15, no. 56, pp. 1929–1958, Jun. 2014.

[25] D. Jurafsky and J. H. Martin, "Logistic Regression," in Speech and Language Processing, 2019.