Visual Person Identification Device using Raspberry Pi

Yame Asfia[1], Sara Tehsin[1*], Areesha Shahzeen[1], Umar Shahbaz Khan[1]
[1]National University of Sciences and Technology
Islamabad, Pakistan

***Abstract*— People with low vision find it difficult to socialize publicly due to lack of face recognition ability. Robust assistive solutions are available but either those are too heavy on the pocket or are not wearable. In this paper, we have proposed a low cost lightweight wearable device, based on raspberry pi 3 that uses enhanced machine learning algorithms. The device has two user-selectable operation modes; recognition mode and new entry or training mode. In recognition mode, a combination of Haar cascade and Hog features are used for face detection and recognition. The accuracy of recognition system is further improved by capturing a burst of five images for each subject. Face detection and recognition algorithms are next applied to each frame and the name with maximum votes is announced on the earphones. In case of unrecognized faces, system announces "Unknown". For the training mode, a web based user interface has been proposed that can help the user for enrolling and updating entries, into the dataset. The system has been tested in real time environment. Results show the accuracy of the proposed methodology.**

I. Introduction

Last decade has seen a massive advancement in computing technology and increase in computing power. Availability of such high-end systems has catalyzed the research of computationally expensive algorithms like face recognition. Face recognition has become one of the most active areas of study. While most of them are focusing on Face recognition problems and algorithms, some researchers are also implementing the knowledge base for developing wearable electronics that can assist users in a variety of ways.

The assistive technology was first introduced in 1960s. Since then, variety of wearables have been developed to address security, social and bio-medical needs of the disabled. In the context of visual assistance, the technology can be further divided into three categories: visual enhancement, visual substitution and visual replacement, as explained in a survey paper about sensor based assistive devices for visually impaired [1]. Visual replacement relies on medical technology to provide assistance by displaying the visual information directly to the visual cortex or through ocular nerve. This technology is the most complex of all. Visual enhancement, as obvious from its name, deals with enhancement of camera input. Enhanced results are there on displayed visually. Contrary to this, in case of visual substitution, processed results are displayed non-visually e.g. through some audio signal or vibration etc.

Our proposed work lies in the third category of visual assistance i.e. visual substitution. The aim is to provide a cost effective yet a robust solution to visually impaired persons in form of a device that can identify faces of people whom they meet daily and then announce their names on the earphones. The wearable is based on Raspberry pi 3 model B [2] that is a lightweight Single Board Computer (SBC), while face recognition is achieved using a combination of Haar cascades classifier and HOG transform.

The main advantage of using Raspberry pi is the range of free software and libraries available for the board. Operating system used is Raspbian Jessie that is freely available. Code is written in python using the well-known library **face-recognition v1.2.3** [3-4] that wraps **dlib's** [5] functions for calculating face encodings and later on using them for recognition. **OpenCV's** [6] functions are employed for the purpose of face detection with Haar cascades as the detector and for image tagging. For the purpose of name announcement in the end, Espeak, an open source utility for text-to-speech conversion, is employed. Buttons and LED interfaced on the GPIO header are controlled using **wiringpi** library that is also free ware.

Related work in the area of visual assistance technology can be found in [7-14]. iCare Interaction Assistant developed by Krishna et al. [7], and FaceSpeaker [8] helped people with visual impairment to interact socially and helps recognize the faces they meet. In case of success, system outputs the subject's name as audio signal on headphone. Hardware employed in iCare consists of analog CCD camera glasses, a digitizer that connects through USB cable to a tablet PC with an Intel® Centrino 1.5 GHz processor and 512 MB of RAM and currently supports 10 subject faces at max. FaceSpeaker on the other hand is a prototype wearable device. It is based on a laptop worn as backpack. The laptop ran the open source FaceSpeaker software. Both iCare and FaceSpeaker are partial wearable solutions as a visual impaired which person is expected to carry the tablet plus digitizer or a laptop everywhere. Another eye wearable system was built around the same aim for patients with prosopagnosia by Wang et al. [9]. This system assisted these patients to identify the people they meet (limited to 20 faces) using local binary pattern (LBP) features. However, due to high processing requirements, the system was designed to run on a smartphone. Though the results showed some improvement over iCare but the camera and display units' placement in the wearable eye glass (Vuzix STAR 1200XL third generation augmented reality device), made it an expensive solution. Mandal et al. proposed a face detection, eye localization and face recognition system using Google glasses [10]. His Facial recognition methods showed massive improvement over eye wearable system and iCare but all of these techniques did not present cost effective ideas. A Microsoft kinect sensor based wearable face recognition system was proposed by Neto et al. to aid visually impaired [11]. As presented, it was based on computationally efficient algorithms

(Histogram of Oriented Gradients HOG, PCA and K nearest neighbor K-NN) and was expected to run in real time on small sized low power devices. Moreover, unlike the previously mentioned schemes in which monaural audio feedback was used, Neto used spatial audio to convey the position of face. However, the range of Kinect sensors limits the application to only in-doors [11]. Another system named as DEEP-SEE FACE claimed that it can recognize people both in indoor and outdoor environment. It extended previously proposed idea in DEEP-SEE [12-13] with a face recognition module. With DEEP-SEE FACE, patients can even recognize celebrities or people appearing in media. Hardware architecture consisted of mobile acquisition device, a light processing unit equipped with Nvidia GPU and a pair of bone conduction headphones. Vadiraja et al. proposed a raspberry pi based face recognition device with image tagging facility for blind [14]. The device is expected to communicate information to the user in a timely and private manner as a full name. Notification is accomplished by broadcasting an audio message, which was recorded during the registering process of this face, over earphone. Face detection is carried out using OpenCV[6] library using Fisher faces algorithm. Our proposed scheme differs from [14] in both recognition and training modes. Instead of using Fisherfaces, we have incorporated a combination of Haar cascade and HOG features that are less complex and hence work well for real time recognition and require less power. Moreover, instead of saving audio of the person as in [14], we have enrolled their names using a web based user interface that shall be explained in the next section

Before going into the details of the proposed scheme, Section II explains the system model and methodology while the experimental setup and results are explained in Section III of the paper. Section IV contains the conclusion of the paper.

II. PROPOSED METHODOLOGY

The proposed system is a lightweight, inexpensive wearable based on Raspberry pi 3 Model B interfaced with a 5MP pi Camera module as shown in Fig. 1. Raspberry Pi 3 Model B belongs to the third generation of Raspberry Pi SBCs with maximum current rating of 2.1A. It is a single board computer with a quad-core 1.2GHz 64 bit processor equipped with 1GB RAM. The operating system (raspbian) and filesystem are stored on an SD card. A size of 16GB is good enough to store the dataset required for experiment. The device is powered with the help of a power bank that can provide 5-6 hours of continuous supply.

The device works in two modes (1) Training/ New Entry mode and (2) Recognition mode. These modes can be switched with the help of a mode selector switch. Another push button is interfaced on the GPIO header for requirement based recognition control. Whenever required, this button can be pressed to capture the image that is then processed to identify the presence of any known face. In case of success, name of that person is announced on the earphones interfaced with the audio out jack of the SBC. If no face is found or the captured face does not match with any of the faces in the dataset, the system says “Unknown”.

The new entry mode can be selected with Mode switch. This mode is required for creating and managing dataset of faces that user wants to recognize. For this purpose, a web based user interface has been created that can be accessed by connecting to the device via its wifi IP address 192.168.1.1 with raspberry pi acting as an access point.

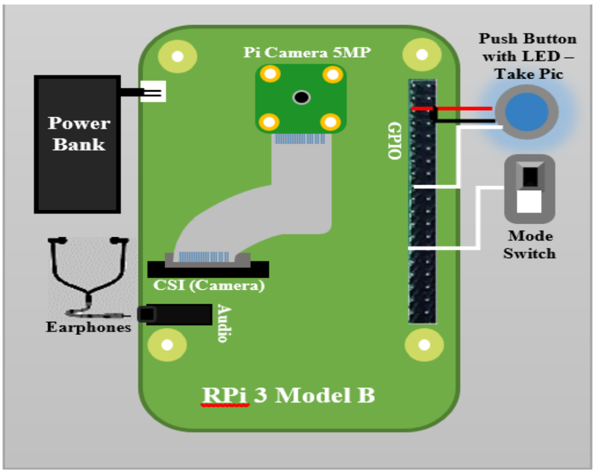


Fig. 1. System Model

A. New Entry and Training Mode

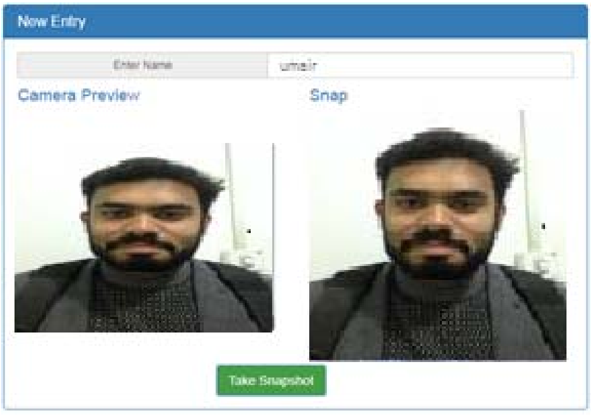
Fig. 2, 3 and 4 show the web interface provided to users. For new entry, user is required to enter name of the person whose pictures are going to be taken. A new directory inside the dataset shall be created for every new enrollment. The user will then press “Take Pic” to take and save pictures in this directory. For the training purpose, at least five pictures taken from different facial angles are mandatory and preferably with varying expressions such as with a smile, also those with a straight face. In this way, new candidates and their images shall be enrolled in the database.

In case of record update of an existing candidate, form shown in Fig.3 shall be used. User will select a name from the dataset. New images of existing candidates can then be taken and stored in their respective directories inside the dataset as shown in Fig.4. Once the training dataset has been finalized, the next step involves locating coordinates of faces in each image.

For each detected face, a vector of 128 values, also called face encodings, is computed. Face encodings along with their “names” are then saved into a file that shall be accessed later during the process of face recognition.

ALGORITHM 1: NEW ENTRY AND TRAINING MODE

```
0:  If New Entry: Step 2, Else: Step 4;
1:  Get NAME from user
2:  Create dir NAME, Set Image Counter = 0
3:  If Take Pic = ‘Yes’ : Step 5, Else: Step 9
4:  Capture Image
5:  Save file ImageCounter.png
6:  Image Counter ++
7:  If Take More = ‘Yes’: Step 5, Else: Step 9
8:  Compute HOG features and face encodings
9:  EXIT
```



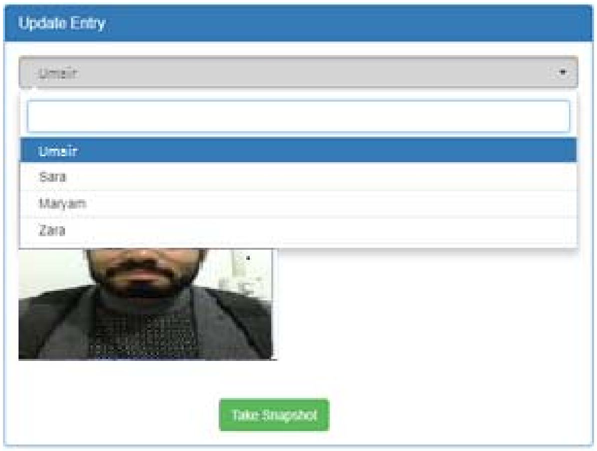
New Entry

Enter Name: umair

Camera Preview

Snap

Fig. 2 New Entry Form

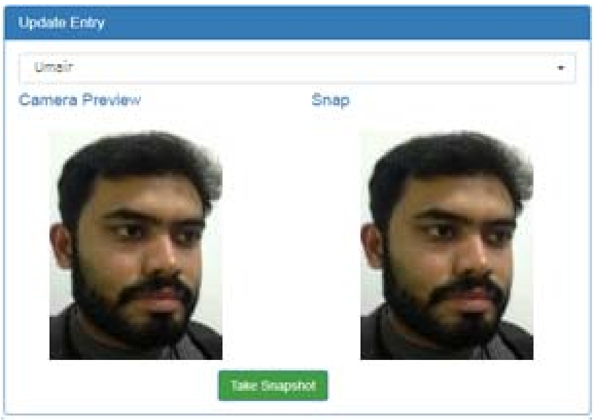


Update Entry

Umair

- Umair
- Sara
- Maryam
- Zara

Fig. 3 Update Record Form – Record Selection



Update Entry

Umair

Camera Preview

Snap

Fig. 4 Update Record Form – New Snapshots

Recognition Mode

Algorithm-II shows the steps for recognition mode that is the default mode of the device. The recognition process is divided into two stages, face detection and then face recognition. As shown in the figure, whenever the button is pressed, camera captures a burst of five frames.

Captured frames are then processed using Haar features to detect faces in them. Haar cascading algorithm was proposed by Voila and John [15]. This algorithm is used widely for face detection which consists of multiple steps. There are four key concepts which are combined in Haar cascade algorithm. First one is rectangular Haar features which is a weak classifier for face detection. Second step is creating integral image, means adding small portion pixel values together. The whole image is integrated by few integrated operations per pixel, starting from top left corner of the image and traversing to the right and down. Third step is minimizing the feature set through adaboost algorithm by selecting the weak classifier with the lowest error rate as output of each round and then reweight all the examples. Last steps of Haar cascading is to pass minimized feature set from cascaded classifiers. The chain of filters identify the presence of face in each sub region and eliminate the non-face sub region quickly.

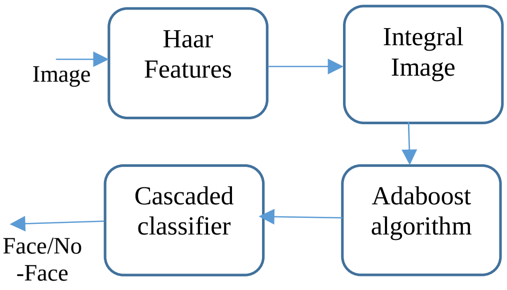


Fig. 5 Face Detection using Haar cascades

Using Haar cascade classifier to train the detector returns the coordinates of the found faces. If a null vector is returned, i.e. no faces are detected, code exits with an announcement made on the earphones about "No face found" and the system returns to the first state where it waits for the next button press.

In case when one or more faces are detected, all frames enter the second stage where 128-D face encodings vector of each face is computed. These encodings are Histogram of Gradient (HOG) features which proposed by Dalal el al [16]. The computation of global descriptor is done through following steps:

1) Compute horizontal G_x and vertical G_y gradient of image by filtering image with [-1 0 1].

2) Compute magnitude and orientation of the gradient using equation (1) and (2) respectively. G_x and G_y are the horizontal and vertical gradient of (u,v) pixel of the image respectively.

$$M_c(u,v) = \sqrt{G_x(u,v)^2 + G_y(u,v)^2} \quad (1)$$

$$O_c(u,v) = \tan^{-1}\left(\frac{G_y(u,v)}{G_x(u,v)}\right) \quad (2)$$

3) Divide image into cells.

4) Compute histogram of each cell

5) Normalize all histogram within a block of cell.

The global descriptor is obtained by combining all normalized histograms in a vector.

These encodings are then compared with the encodings of the dataset. The process repeats for all five frames. If no match is found in any of the frames, parameter "name" is registered as “Unknown Person”. In case of successful matches, corresponding names from the dataset are registered as “name”. At this stage, we have five names registered for each frame of the same person. The value of these names may or may not be same, depending on the orientation of face and illumination. The name with maximum votes is registered as the final name and then announced after converting text to speech.

```
Algorithm 2: Recognition Mode
0:  If Button pressed= 'Yes': Step 2,
1:  Capture 5 frames
2:  Gray_Face <= RGB to GRAY
3:  Detect Faces: Haar_Features (Gray_Face)
4:  If Face Found == 'No' : Announce 'No Face
    Found', EXIT
5:  Compute Face Encodings : HoG(Faces)
6:  Compare with known encodings
7:  If Match_Found==NULL: Announce
    "Unknown Face" , EXIT
8:  Announce Name with Max Votes
9:  EXIT
```

In our implementation of algorithm 1, features have been extracted from captured image that is passed to create integral image. Feature minimization has been done through adaboost algorithm. These features are classified by cascaded classifier and face detected results have been displayed. When face detects through Haar cascade algorithm then HOG features are computed for recognition. HOG features are immune to geometric and photometric variances. It mainly works on local cells and gives good results in human recognition in images. The extracted face is passed to calculate horizontal and vertical gradient of image. From which magnitude and orientation are computed. Histogram of image is calculated and normalized onwards. The results are then classified by comparing training and testing results.

The computational complexity of Haar cascade algorithm is O(N*M) where NxM is the number of rows and column pixels of input image. It’s a linear complexity. The computational complexity of encoding algorithm for face recognition is $O(2^n)$.

B. *Power Consumption*

As already mentioned that the proposed device is battery operated, therefore it must consume least power at all times to prevent early battery drain. The power consumption of Raspberry pi 3 B is 1.4W in idle state and 5.1W at 400% CPU load [20] that is quite reasonable. Additionally, the proposed face recognition system is programmed to operate only at a button press, hence most of the time CPU is at idle state consuming at most 1.4W power. When the button is pressed, the execution involves burst capture and comparison with already existing encodings. As the existing encodings are pre-calculated values stored in a file, therefore average power consumption of the device stays around 1.6W.

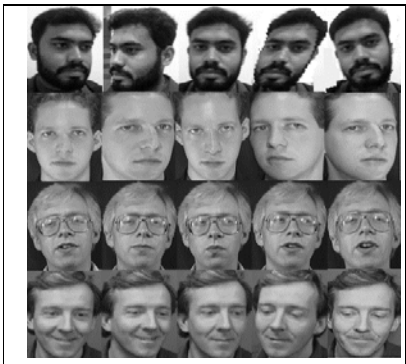
III. Experimental Setup And Results

The proposed method presented in the previous section was validated in two steps.

- Simulation and Verification on MATLAB
- Implementation on Raspberry pi 3

Publicly available dataset of faces images taken at the AT&T Laboratories [17], Cambridge University consisting of forty (40) distinct subjects, in conjunction with our own dataset of facial images sizing 92 X 112 pixels having 256 grey levels and belonging to twenty distinct persons is used for testing the system. This database was initially used for face recognition project done by robotics and speech vision group of Cambridge University. The dataset consists of ten images per subject which was taken in dark homogeneous background in upright frontal position. This database is also called as Olivetti- AT&T-ORL. A part of enrollment used rpi’s camera. However, the candidates were also requested to provide some random pictures for testing accuracy. Table-1 shows some of the dataset entries of publically available AT&T dataset and self-generated dataset.

TABLE I. Subset Of Dataset

	UMAIR
	AT&T Face 1
	AT&T Face 2
	AT&T Face 3

A. *Simulation: MATLAB*

The face recognition algorithm was tested in MATLAB before its implementation on the embedded platform. Figure-6 shows simulation results on MATLAB. First of all the captured image is converted from rgb to gray scale (if not already) for the purpose of face detection using Haar features. The resulting boundary is then used to crop out the face from the image. The extracted face is then compared with the faces in the dataset. In case of successful match, system output’s the Matched Class i.e. the person’s image from the dataset to whom this face belongs.

B. *Implemention on RPi 3*

The device was tested under different light conditions and with different camera angles. For persons with faces already enrolled in the dataset, accuracy of finding the correct face is around 75% to 95% depending on the size of dataset. The larger the size of the dataset, the better the recognition accuracy. For a smaller dataset consisting of around 4 to 5 images per subject with 10 subjects in total, accuracy of recognizing faces is 75%. In such a dataset, accuracy of finding unknown faces is slightly better i-e 85%. However as the dataset size increases, i-e if number of images per subject increases to 10 and the number of participants increases to 40, then the recognition accuracy improves to 95%.

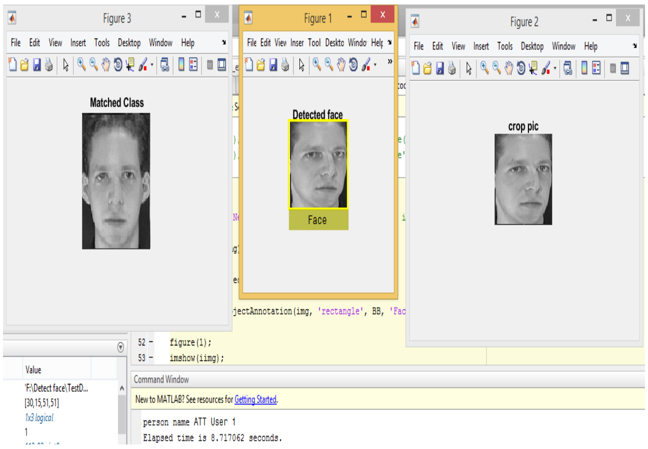


Figure 3
File Edit View Insert Tools Desktop Window Help
Matched Class
Figure 1
File Edit View Inser Tool Deskto Windo Help
Detected face
Face
Figure 2
File Edit View Insert Tools Desktop Window Help
crop pic
Value
Command Window
New to MATLAB? See resources for Getting Started.

```
jectAnnotation(img, 'rectangle', BB, 'Fac
    figure(1);
    imshow(iimg);
```

```
'F:\Detect face\TestD...
[30,15,51,51]
1x3 logical
1
```

```
person name ATT User 1
Elapsed time is 8.717062 seconds.
```

Fig. 6 Simulation Results: MATLAB

[18] has presented a comparative study of PCA, ICA and LDA using SVM classifier on AT&T dataset and according to them, the resulting accuracy is around 91% to 93.9%. [19] has also used the same dataset for studying SVM, CNN, and ANN with BOW, HOG and IP. According to their study, HOG in conjunction with SVM, CNN and ANN results in 96% to 99% accuracy. However as the neural networks are complex and time consuming, the presented work here has employed HOG because of the real time constraints. Hence an accuracy of 95% is comparable to [18] and [19]. Table II shows that there is a tradeoff in accuracy vs. recognition time.

TABLE II. ACCURACY AND RECOGNITION TIME VS. DATASET SIZE

No.	Dataset size (No of subjects)	ACC	Recognition Time (sec)
1.	10 with 5 samples each	75%	0.25
2.	20 with 5 samples each	80%	0.34
3.	30 with 10 samples each	89%	0.94
4.	40 with 10 samples each	95%	2

As regards the recognition time required by the system, for a smaller dataset, faces are recognized in mere fraction of seconds. However, if the dataset size increases to 40 faces with 10 faces each subject, recognition process takes on average 2s. Recognition time and accuracy also depend on the number of frames captured during the recognition process. As the number of frames increase, the accuracy increases but this also increases the recognition time and vice versa. Table III shows this relationship.

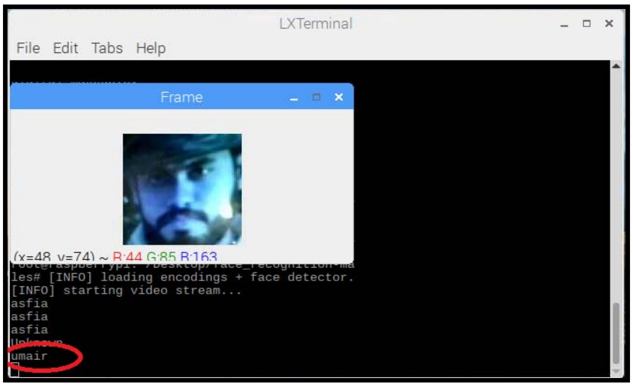
TABLE III. ACCURACY AND RECOGNITION TIME VS. NO OF FRAMES

No.	No of Frames	ACC	Recognition Time (sec)
1.	1	75%	0.1
2.	2	83%	0.24
3.	4	93%	1.3
4.	5	95%	2

Fig.7-10 show the results of recognition in case of enrolled faces. Output shows the names of recognized persons. On the other hand, figure 11 shows a test of recognizing face no 40 from AT&T dataset, which is not enrolled in the dataset. System outputs “Unknown” in this case. Results improve greatly with better light conditions and correct camera angle.

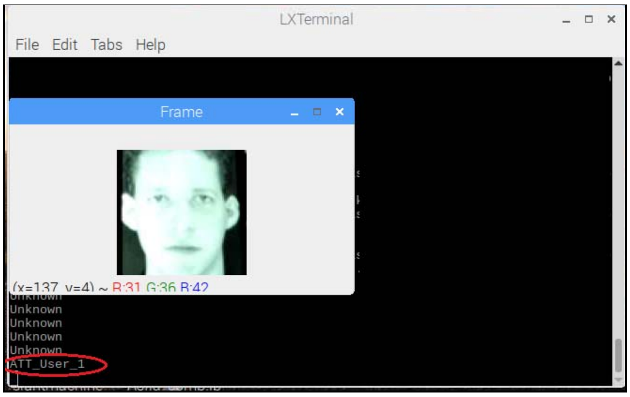
C. Comparison with Existing Work

As already mentioned in Section I, similar schemes exist but most of them [7-13] do not consider cost as a problem. Moreover, power consumption of these devices is also an issue. In [14], a comparable solution, using Raspberry Pi, is presented but they have employed Fisher faces algorithm for face recognition. Fisher faces, although a good choice for recognition, is computationally intensive as compared to proposed algorithm based on HOG and HAARCascade that take require less computation time. Therefore, not only performance speedup is achieved but also power consumption is reduced giving long battery time to the user. Furthermore Fisher faces algorithm is highly dependent on input image’s illumination and hence doesn’t work well for images with differing illumination of the same person. Moreover, [14] presents an algorithm that always stays in face detection and recognition mode that not only increases power consumption but is also highly undesirable for the user. This problem is addressed in our proposed design by introducing a push button. Face detection/ recognition process only runs when required by the user. This gives more control and comfort to the user and as mentioned earlier reduces power consumption to around 1.6W on average.



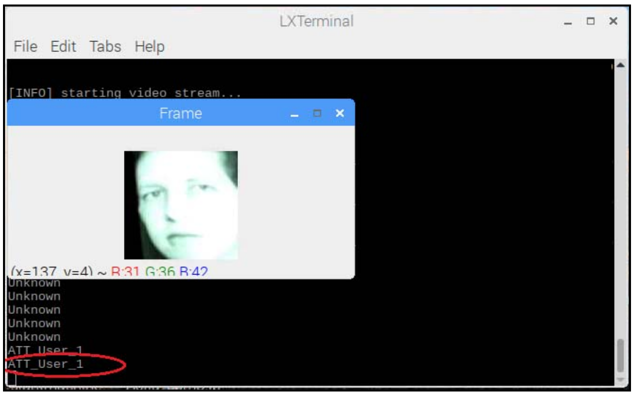
```
les# [INFO] loading encodings + face detector.
[INFO] starting video stream...
asfia
asfia
asfia
Unknown
umair
```

Fig. 7 Face Recognition Result – Known Face “Umair”



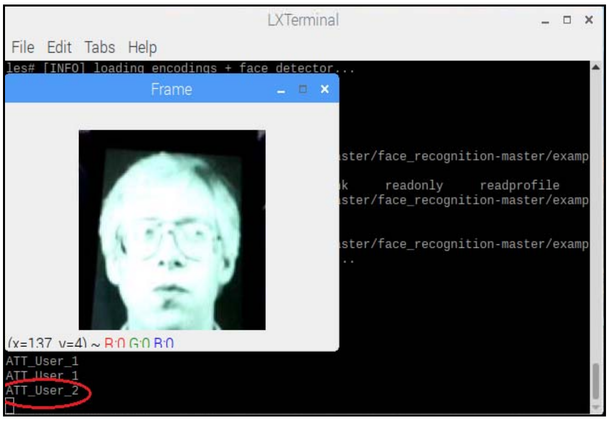
```
Unknown
Unknown
Unknown
Unknown
Unknown
ATT_User_1
```

Fig. 8 Face Recognition Result – Known Face "ATT User 1"



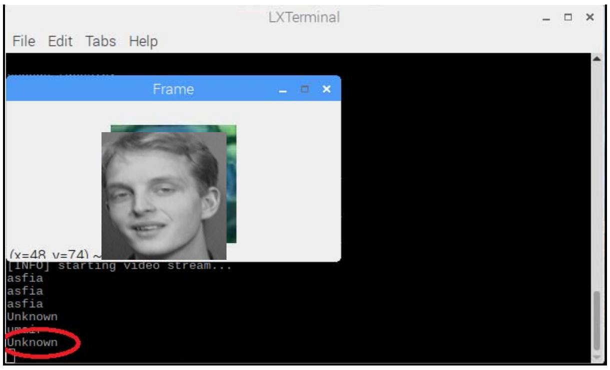
```
[INFO] starting video stream...
Unknown
Unknown
Unknown
Unknown
Unknown
ATT_User_1
ATT_User_1
```

Fig. 9 Face Recognition Result – Known Face "ATT User 1"



```
les# [INFO] loading encodings + face detector...
ster/face_recognition-master/examp
k      readonly      readprofile
ster/face_recognition-master/examp
ster/face_recognition-master/examp
..
ATT_User_1
ATT_User_1
ATT_User_2
```

Fig. 10 Face Recognition Result – Known Face "ATT User 2"



```
[INFO] starting video stream...
asfia
asfia
asfia
Unknown
umair
Unknown
```

Fig. 11 Face Recognition Result – Unknown Face

IV. Applications

This device is primarily developed for people with visual impairment problems. However, this wearable is also useful for people with memory issues like for example Alzheimer's disease. This device is able to assist anyone who has the habit of forgetting names. Moreover, it can make a smart classroom attendance system as well.

V. Conclusion And Future Work

In this paper, we have proposed a low cost, low power, lightweight visual assistive wearable that will enable persons with low vision to recognize people in their community. Face recognition is accomplished using Haar features and Histogram of Gradients (HOG) descriptors that are computationally efficient as compared to other algorithms. The device can work in two modes, training mode and recognition mode. For the training mode, we also propose a user-friendly web interface, as the device acts as an access point and thus can be connected over Wi-Fi link. As a future work, the device can be equipped with a touchscreen LCD to make it more user friendly. User authentication methods needs to be introduced in the system to make it secure.

Acknowledgment

We would like to show gratitude to our class fellows from the Department of Mechatronics Engg, CEME NUST Rawalpindi, who shared their pictures with us and supported us during device testing and simulation. Work of all researchers and open source community is highly appreciated without whom this work was impossible.

References

[1] Elmannai, W., & Elleithy, K. (2017). Sensor-based assistive devices for visually-impaired people: current status, challenges, and future directions. Sensors, 17(3), 565

[2] Raspberry Pi: Web site. https://www.raspberrypi.org/products/raspberry-pi-3-model-b/ (accessed November 13, 2018)

[3] Adrian Rosebrock. Face recognition with OpenCV, Python, and deep learning. Web site: https://www.pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/ (accessed November 13, 2018)

[4] A. Geitgey, Face recognition: The world's simplest facial recognition api for python and the command line, original-date: 2017-03-03T21:52:39Z, Jan. 30, 2018. [Online]. Available: https://github.com/ageitgey/face recognition/

[5] Davis E. King. Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research 10, pp. 1755-1758, 2009

[6] OpenCV: Open source computer vision. Web site. http://opencv.org/ (accessed October 13, 2018)

[7] Krishna, S., Little, G., Black, J., & Panchanathan, S. (2005, October). A wearable face recognition system for individuals with visual impairments. In Proceedings of the 7th international ACM SIGACCESS conference on Computers and accessibility (pp. 106-113). ACM.

[8] FaceSpeaker: Web site. http://www.facespeaker.org/ (accessed October 13, 2018)

[9] Wang, X., Zhao, X., Prakash, V., Shi, W., & Gnawali, O. (2013, May). Computerized-eyewear based face recognition system for improving social lives of prosopagnosics. In Proceedings of the 7th International Conference on Pervasive Computing Technologies for Healthcare (pp. 77-80). ICST.

[10] Mandal, B., Chia, S. C., Li, L., Chandrasekhar, V., Tan, C., & Lim, J. H. (2014, November). A wearable face recognition system on google

glass for assisting social interactions. In Asian Conference on Computer Vision (pp. 419-433). Springer, Cham.

[11] Neto, L. B., Grijalva, F., Maike, V. R. M. L., Martini, L. C., Florencio, D., Baranauskas, M. C. C., ... & Goldenstein, S. (2017). A kinect-based wearable face recognition system to aid visually impaired users. IEEE Transactions on Human-Machine Systems, 47(1), 52-64.

[12] Mocanu, B., Tapu, R., & Zaharia, T. (2018). DEEP-SEE FACE: A Mobile Face Recognition System Dedicated to Visually Impaired People. IEEE Access.

[13] Tapu, R., Mocanu, B., & Zaharia, T. (2017). DEEP-SEE: Joint Object Detection, Tracking and Recognition with Application to Visually Impaired Navigational Assistance. Sensors, 17(11), 2473.

[14] Acharya, Vadiraja. (2015). Raspberry Pi based wearable Face Recognition device with image tagging facility for blind. NCRTS

[15] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154.

[16] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE

[17] The database of faces AT&T Laboratories Cambridge: website:https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html (accessed October 13, 2018)

[18] A. Bouzalmat, J. Kharroubi, and A. Zarghili, Comparative study of pca, ica, lda using svm classifier, vol.2, Journal of Emerging Technologies in Web Intelligence, pp.64-68, 2014.

[19] Islam KTo, Raj RG, Al-Murad A. Performance of SVM, CNN, and ANN with BOW, HOG, and image pixels in face recognition. 2nd International Conference on Electrical and Electronic Engineering; 2017. https://doi.org/10.1109/ CEEE.2017.8412925

[20] Power consumption benchmarks: website http://www.pidramble.com/wiki/benchmarks/power-consumption (accessed October 13, 2019)