Face Recognition with Audio Output: An Aid for the Visually Impaired

Khyati Morparia
dept. of Electronics and telecommunication
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
khyatimorparia@gmail.com

Anushka Kanabar
dept. of Electronics and telecommunication
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
anushkakanabar@gmail.com

Sachin Bhadoriya
dept. of Electronics and telecommunication
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
sachinsinghbhadoriya1@gmail.com

Poonam Kadam
dept. of Electronics and telecommunication
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
poonam.kadam@djsce.ac.in

***Abstract*—80 percent of what we perceive comes through our sense of sight. The human eye provides us with vision, which allows us to understand more about the world than the other four senses combined. However, not everyone is born with good or great vision. According to a recent survey, at least 2.2 billion individuals worldwide suffer from near or distance vision impairment. The inability to recognise individuals in large groups is a disadvantage for disoriented people in a range of professional and educational circumstances. Vision impairment has been treated in at least a billion or almost half of these instances. Recognition of persons, things, and routes is without a doubt one of the most significant deterrents for visually impaired individuals which restricts their freedom of movement. It is not always appropriate to identify someone just based on their voice. To recognise that someone is in front of them, the visually challenged require assistance to avoid any trickery and make them feel safe. Like the Braille System, which helps visually impaired people read and write, there is optimism that in the near future, technology may help blind people identify faces more quickly. This framework's goal is to assist the client in recognising persons without the aid of a third party.**

Keywords— Computer Vision, Deep Learning, Face Recognition, Raspberry Pi, Image Processing

I. Introduction

Image processing is the methodology employed to either get an enhanced image or extract useful information from the image. In this model, we have used image processing techniques to achieve an accurate face recognition model. Thus, image processing forms the foundation of the proposed model. Further modifications are made in order to make the model more advance. These points are highlighted later in the article.The main objective of this project is to provide a fresh hypothetical strategy in order to assist visually impaired people. Our model aims to ease some significant problems faced by the visually impaired in today’s world and in the process helps increase security and independence in them. The paper also enlists the shortcomings of the project and describes features that can be integrated in the future for more efficient working. The model typically detects and recognizes an individual. This is made more compatible for a visually impaired person by converting the text output to audio. The method proposed here has a one-tier architecture. The main components of this project are a Raspberry Pi 4 Module which was chosen for its configuration that catalyzes image processing. The in-built camera operates as an eye to send live streaming footage to the processing unit so that it can conduct functions. The algorithm used in this paper is known as deep metric learning. We chose this algorithm since it proved to be more accurate than the other proposed methods of face recognition. Raspberry Pi was chosen because it is inexpensive and the prototype is small in comparison to other prototypes.[3] The Raspberry Pi device is in charge of detecting and recognising faces as well as informing the visually impaired user about the name of the person. The recognised face's output result will be the text to speech output.

Our paper is broadly divided into three parts:

1) Face detection:
   In this step, the task of the model is to determine whether a given group of images belong to a certain person or not
2) Face recognition:
   This step involves both, face identification and detection. In the identification step, the task of the model is to recognize a person from a database of images
3) Text to speech of the output of the recognized person.

An in depth explanation of the above mentioned categories has been given as we move forward with the paper. The objective of adding a text to speech feature was to make the model more suitable to aid those who are visually challenged. Through this we aim to help society through our knowledge in the technical domain. Incorporating this fundamental model in various other applications can prove beneficial and aid the visually impaired community.

II. Literature Review

The suggested primary face recognition application was developed in Python using the Open CV image processing package and the LBPH algorithm on an HD camera [5]. The facial recognition algorithm was divided into three distinct and independent portions:

a) Face Detection

b) Pre-processing

c) Feature Extraction

d) Feature Matching

They built their own dataset, which had various facial expressions and postures used in a particular scene in order to recognise faces.[6] A face detection method based on LBPH was utilised. The LBPH method is a mixture of the descriptors Local Binary Patterns (LBP) and Histograms of Oriented Gradients (HOG). [7][1] The application connects an externally connected camera to the PC and uses the HAAR classifier to recognise and collect facial photographs. This technique allowed for the storage of face pictures in a folder labelled displaying subject ID and sample number. [5] Following pre-processing, all 2000 recorded photos of four subjects were saved in the same folder. Each photograph received a subject ID and a sample number. The amount of photos per face image is referred to as the sample number. A single face image has different sample number but same subject ID, thus leading to the successful recognition of the input face. [1]

Drawing inspiration from the above summarised article, we have employed the Deep Metric Learning algorithm in order to carry out face recognition for a given dataset in our model. We have further added a text to speech feature which converts the output to audio, thus making it a beneficial feature for the visually impaired, increasing their safety as well as independence.

III. Proposed System

To begin with, we used an example dataset of characters from the movie Jurassic Park to train our algorithm. Then we sought to put the system to the test by feeding the algorithm with their photos and letting it detect and recognise their faces. We also provided a video input, which our system looped as numerous frames before applying the facial recognition algorithm to each of those frames. We trained our system to recognise faces not just on pre-existing photographs and movies, but also utilising a camera in this way. We just had to provide the model with sample photographs and train it in the same manner. Once our system has identified the person, we convert the text to voice so that visually impaired people can hear the name of the person in front of them.

The following schematic shown in Figure 1. gives an overview of the working of the model using the required hardware. This was successfully implemented in order to achieve the desired result.

.

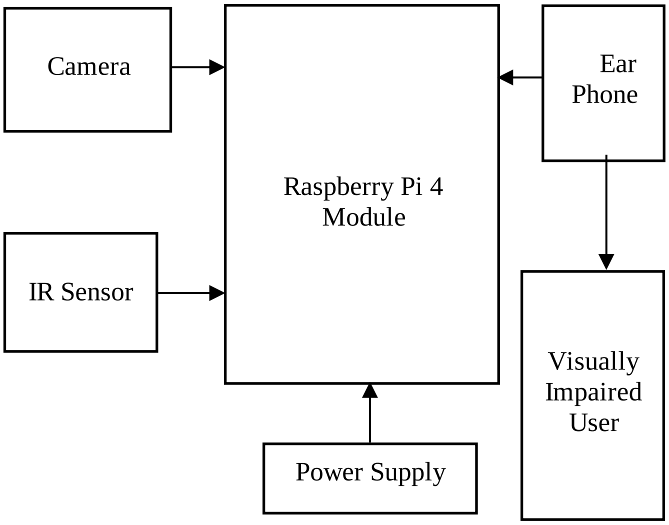

Fig. 1. Block diagram of the working of the project

The proposed model uses Deep Metric Learning. Metric learning algorithm is the combination of rules or a complex function that basically maps the inputs to the corresponding output labels. Metric learning aims to learn a similarity function from data. It is a method for determining picture similarity or dissimilarity that relies solely on a distance measure. Building an efficient face recognition system for large scale essentially involves the designing of an appropriate loss function that discriminates the classes under study.[13] For many Computer-Vision tasks such as face recognition, face detection, picture classification, and so on, metric learning with deep learning has proven to deliver accurate and efficient results. Metric learning is only capable of capturing non-linearity in data to a limited extent. However, when combined with deep learning, we can learn a non-linear transformation of the feature space to capture non-linear feature structures. Deep Metric Learning can be classified into two different approaches:

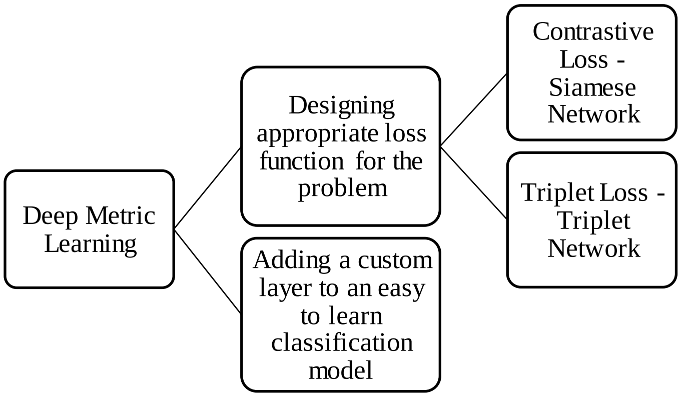
Deep Metric Learning
Designing appropriate loss function for the problem
Adding a custom layer to an easy to learn classification model
Contrastive Loss - Siamese Network
Triplet Loss - Triplet Network

Fig. 2. Classifications of Deep Metric Learning approaches

Custom Layer: One reason for deep learning's success is the availability of a diverse set of layers that may be combined in novel ways to create architectures appropriate for a wide range of applications. For example, academics have created layers specialised for dealing with graphics, text, looping over sequential data, and dynamic programming. Custom layers may be created using the basic layer class, which enables us to develop flexible new layers that act differently from any other layers in the library. Custom layers, once specified, can be used in a variety of settings and designs.

Local parameters for layers can be defined using built-in functions.

The Triplet network is used in this project. It's a symmetrical neural network design. It is made up of three identical subnetworks with the same set of characteristics. Learning is done in the form of a series of three pictures:

1) The baseline image
2) The positive image
3) The negative image

The goal of this technique is to ensure that a person's baseline picture is closer to all positive sets of photos than it is to all negative groups of instances [13].

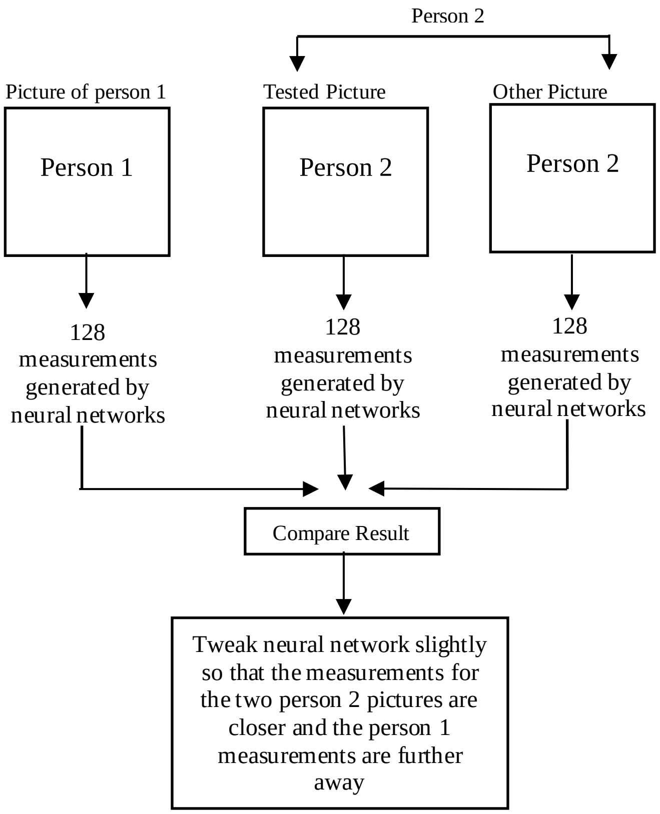
Person 2
Picture of person 1
Tested Picture
Other Picture
Person 1
Person 2
Person 2
128 measurements generated by neural networks
128 measurements generated by neural networks
128 measurements generated by neural networks
Compare Result
Tweak neural network slightly so that the measurements for the two person 2 pictures are closer and the person 1 measurements are further away

Fig.3. Triplet Network Flow Chart

The equation for triplet loss can be expressed as:

$$L = L(a, p, n) = \max\left(0, m + ||f(a) - f(p)|| - ||f(a) - f(n)||\right.$$

Where:
a = Baseline image
p = Positive image
n = negative image
m = margin

Speed of deep metric learning algorithm: less than 10 milliseconds.

Despite its high accuracy, the Deep Metric Learning algorithm is a relatively unexplored algorithm in terms of execution.

Thus, in this paper, we chose to use the algorithm to implement our model.

A. *Software Used*

OpenCV: Computer vision is the study of how images and movies are stored, as well as how one may manipulate and extract information from them. In new developing technologies such as self-driving vehicles and robotics, computer vision is becoming increasingly vital. OpenCV is an open-source library which is usually utilised for Computer Vision, machine learning, image processing, and real-time operations in various systems today. It has the ability to distinguish objects, people, human handwriting, etc. in photographs as well as videos. When paired with additional libraries such as numpy, Python processes the OpenCV array structures for analysis. To identify visual patterns along with its attributes, vector space is used on these characteristics and mathematical operations are performed. Some quotidian uses of OpenCV include, keeping track of moving items, disclosure, and even Covid applications like detection of face masks.[4] In our model we have employed the OpenCV library in the python code in order to assist the model detect and recognise the person.

Advantages of OpenCV:

1. OpenCV provides access to a variety of powerful computer vision techniques for 2D and 3D image and video processing. [11] Normally, these methods are only available in high-end image and video processing software.
2. OpenCV API also makes video analytics much more efficient and easier to implement
3. Real-time video analytics capabilities include the ability to identify, recognise, and monitor minute aspects of both objects and people

Text to Speech: Python has various APIs which are used to convert text to speech. Google Text to Speech (gTTS) is one such API. It is a python library and command-line interface for interacting with Google Translate API. It is imported and used for speech translation. It effectively converts the typed text to an audio that can be sent as an MP3 file. A lot of languages including English, Hindi, French and German can be used in this conversion. The text variable is in the form of a string which is stored as user input. Rapid and Slow are two audio speeds possible:

Advantages of using gTTS:

1. Only one model needs to be employed to perform text analysis
2. It adapts to new data easily
3. It is capable of generating international and natural speech
4. It has a powerful capacity to capture the minute internal structure of data

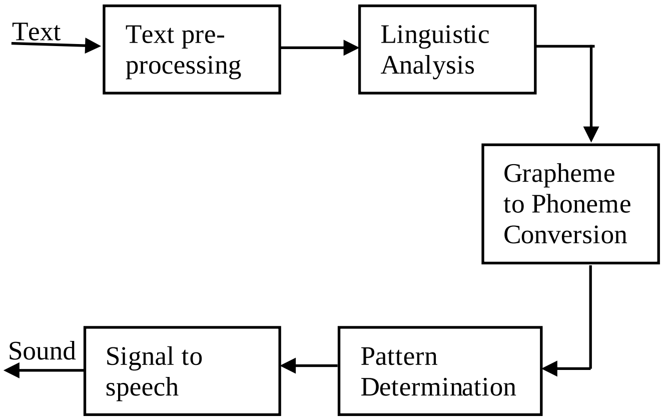

Fig. 4. Text to speech output flowchart

Text-to-speech (TTS) synthesis in general involves the following processes:

Text preprocessing: This preliminary step includes sentence segmentation, tokenization, and normalisation of nonstandard terminology. In its most basic form, tokenization is achieved by dividing the text at white spaces and punctuation marks that do not correspond to the abbreviations defined in the previous step.

Linguistic examination: The study of language or its structure that is concerned with the language itself rather than its subject matter. It delves into everything from the complexities of the sounds we use to generate language to the situations surrounding speech occurrences. The most specialised include phonetics, phonology, morphology, syntax, semantics, and pragmatics.

Graphene to Phoneme Conversion: Grapheme-to-phoneme (G2P) conversion is the process of developing pronunciation for words based on their written form.

Pattern recognition: It basically determines whether or not a string has a specified character pattern.

Signal to speech: Following all of the above processes, the transitory form of the text is converted to speech.

Speed of GTTS: 1 second

Visual Studio Code, an open source text editor that supports a broad range of programming languages from Java, C++, and Python to CSS, Go, and dokerfile, was used to implement the code mentioned above. We chose this platform since it offers cross platform support, advanced extensions, robust architecture which supports all languages and provides all required libraries. One can also easily save the python file in order to upload it on the Raspberry Pi.

Dataset used : In the initial testing of the proposed model, we trained a Jurassic Park characters dataset. This data set consisted of images of 4 characters/people, around 50 images of one person, thus the model successfully trained 222 images in the first. In the next step, we added images of 2 more people, thus making it a 6 people dataset of 260 images – approximately 43 of each person. The model was successfully trained for this dataset as well, and the required output was efficiently showcased.

B. Hardware Used

Table I. Component Specifications

Name of component	Specifications
Raspberry Pi 4	It has a Broadcom 2711 CPU, which is a 64-bit quad-core Cortex-A72 processor with 2GB of RAM. It has two USB 3.0 "Super-Speed" connectors and a real gigabit ethernet port. Wireless LAN 802.11b/g/n/ac (2.5 GHz & 5GHz) H.265 decode (4kp60)H.264 decode Bluetooth 5.0 Dual micro-HDMI ports, 4K UHD video H.265 decode (4kp60) (1080p60) OpenGL ES 1.1, 2.0, and 3.0 graphics are required, as well as a 5V3A USB-C power source. Upgraded and re-engineered from the ground up, Faster and more powerful than ever, Dual monitors The 4K output is quiet, energy-efficient, and can transport data 10 times quicker.
Raspberry Pi camera	The camera is compatible with all Raspberry Pi versions 1, 2, 3, and 4. It may be accessible via the MMAL and V4L APIs, and various third-party libraries, such as the Pi-camera Python library, have been developed for it.
LCD Screen	5inch LCD Display with USB resistive touch control. Compatible Raspbian, NOOBs with Raspberry Pi and Ubuntu Mate.
Micro HDMI to HDMI cable	It can be used with HDTV's, digital cameras, camcorders and other HDMI devices. It supports Ethernet, 3D and ARC

Raspberry Pi 4: Since it includes a dedicated camera input connection that allows users to capture HD video and high-resolution images, the Raspberry Pi 4 module was utilised in the project.[4] Its ability to act as a portable computer when connected to a keyboard along with its high-speed processing makes it ideal for the system. Any task that can be carried out by the computer, can be performed on the Raspberry Pi, such as checking sites, connecting to WiFi, etc.

Reasons of using Raspberry Pi 4 Module:

1. It has various interfaces such as HDMI, multiple USB, Ethernet, onboard Wi-Fi, and Bluetooth, several GPIOs etc.
2. Its adaptive technology and its ability to display and capture images and play videos at high definition resolution make it an ideal fit for the proposed model
3. The Raspberry Pi 4 module is also cost effective and gives a lot of room for experimentation while building the model

4. We can develop programs that take photographs and video and analyse them in real time or save them for later processing using Python and particular libraries designed for the Raspberry Pi.
5. Enough processing power is required to get facial recognition to operate properly, so a Raspberry Pi 4 is perfect. It provides additional memory.

LCD Screen: Liquid screen display – LCD, the name itself describes the technology. A liquid crystal display (LCD) is made up of two states of matter: solid and liquid. Liquid crystal displays (LCDs) employ liquid crystals to generate a visible image. Liquid crystal displays (LCDs) are incredibly thin technological displays screens that are often used in computer screens, TVs, mobile phones, and portable video games. When compared to cathode ray tube technology, LCDs prove to be much thinner and more efficient. In our model, an LCD screen is required to display the initial output of the model when implemented on the Raspberry Pi module. The text to speech feature can then be efficiently added to achieve the aim of the model.

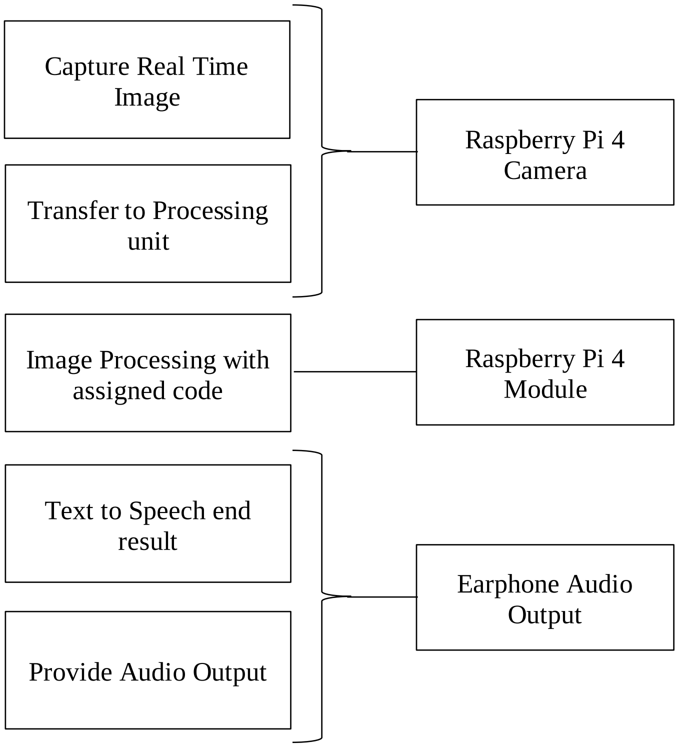

Fig. 5. Usage of Hardware

IV. Comparative Study

Face recognition can be achieved through various different methods. The objective of this model was to chose the most efficient model, with maximum accuracy. Hence we carried out a review of various algorithms that can be employed for face detection and recognition and compared their accuracies. Several face recognition algorithms and techniques, such as PCA (Principal Component Analysis Algorithm), LBP (Local Binary Pattern), etc., have been employed in a number of activities.

The Principal Component Analysis Algorithm (PCA) is an unsupervised machine learning algorithm that is used to accomplish dimensionality reduction. It is a method for simplifying the challenge of finding a consistent representation for any eigen values and their corresponding eigen vectors [2]. This is accomplished by reducing the representation's dimensional space. [3] The dimensions space must be decreased in order to provide rapid and reliable object detection. In general, PCA preserves the data's original information. PCA methods are used in Eigen face-based algorithms.

The Local Binary Pattern (LBP) is a basic yet effective texture operator that identifies pixels in an image by thresholding the pixels' immediate surroundings and treating the output as a binary number. In this method, the input image is divided into cells. Each pixel is compared with its 8 neighboring pixels that surround it. The pixel that has value less than the value of the center pixel, value 0 is assigned, and value 1 is assigned otherwise. On computing the histogram of the entire cell, a 256-dimentional feature vector can be seen. Normalizing the histograms of the cells results in the feature vector of the entire window.[5]

Table II. Comparative study of algorithms

Principal Component Analysis	**Local Binary Pattern**	**Deep Metric Learning**
98.59%	98.59%	99.83%

The table shows a comparative analysis of the three proposed algorithms. While both PCA and LBP show high accuracy rates, Deep Metric Learning supersedes it and proves to be the best fit algorithm.

V. Conclusion and Future Scope

Our goal was to successfully implement face detection and recognition with the help of the selected datasets and algorithms. Our model was a success and gave the desired outputs. Initially, our model only gave text output. This output was modified and enhanced by using the text to speech feature so as to make it accessible to the visually impaired people. This feature enables them to identify the person standing in front of them with the help of a speech output from our system, thus minimising their reliance on a third party to identify other people. This not only ensures independence but also aids in the safety and security of the visually impaired person. Furthermore, we attempted to recognise a person whose face wasn't present in the dataset. This gave us "Unknown" as output, implying that the model didn't recognise the new individual.

This model has various modifications that can be implemented in order for it to cater to several other purposes. It can be added to eye glasses, thus increasing convenience for the user.Face recognition is also considered the rudimentary step for more advance models of emotion recognition and other such features.[12] Thus, this model can act as the framework for further work that can be carried out in the domain to increase the functionality of the model. Thus, this model can prove beneficial as a foundation for many more applications in order to ease the daily struggle of the visually impaired.

VI. References

[1] I.Taufik, M.Musthopa, A.Atmadja, M.Ramdhani, Y.Gerhana, N.Ismail, "Comparison of principal component analysis algorithm and local binary pattern for feature extraction on face recognition system", MATEC Web of Conferences 197, AASEC 2018.

[2] K. Revathi1, J.Bharathi, Saranya.U, "Face recognition using image processing for visuallychallenged" , International Conference on Science, Technology, Engineering & Management [ICON-STEM'15], July 2015.

[3] B.Vishwakarma, P.Dange, A.Chavan, H.Galiyal, "FACE AND FACIAL EXPRESSIONS RECOGNITION FOR BLIND PEOPLE", International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 01, Jan 2017.

[4] Z.Balogh, M.Magdin, G.Molnar, "Motion Detection and Face Recognition using Rasbperry Pi as a part of Internet of Things", Acta Polytechnica Hungarica, June 2019.

[5] F.Deeba, A.Ahmed F.Dharejo, A.Gaffar, H.Memon, "LBPH-based Enhanced Real-Time Face Recognition", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 5, 2019.

[6] Shang-Hung Lin, "An Introduction to Face Recognition Technology", Informing Science Special Issue on Multimedia Informing Technologies-Part-2 Vol 3, 2000.

[7] X.Zhao, C.Wei, "A real-time face recognition system based on the improved LBPH algorithm", IEEE 2nd International Conference on Signal and Image Processing (ICSIP), 2017.

[8] L.Lang, W.Gu, "Study of Face Detection Algorithm for Real-time Face Detection System", Second International Symposium on Electronic Commerce and Security, 2009.

[9] N.Borkar, S.Kuwelkar, "Real-time implementation of face recognition system", International Conference on Computing Methodologies and Communication (ICCMC), 2017.

[10] K.Goyal, K.Agarwal, R.Kumar, "Face detection and tracking: Using OpenCV", International conference of Electronics, Communication and Aerospace Technology (ICECA), 2017.

[11] G.Singh, N.Yadav, G.Nagpal, J.Singh, "Facial Detection and Recognition using OpenCV on Raspberry Pi 0", International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 2018.

[12] M.Zuo, G.Zeng, X.Tu, "Research and improvement of face detection algorithm based on the OpenCV", The 2nd International Conference on Information Science and Engineering, 2010.

[13] E.Hoffer, N.Ailon, "Deep Metric Learning using Triplet Network", International Workshop on Similarity-Based Pattern Recognition, 2015.

[14] U.Bakshi, R.Singhal, "A new approach of face recognition using DCT, PCA and Neural Network in MATLAB", International Journal of emerging trends and Technology in Computer Science,2014.

[15] A.Sanyal, U.Bhattacharya, "A Hybrid Deep Architecture for Face Recognition in Real-life Scenario", International Conference on Computer Vision, Graphics, and Image Processing, 2017.

[16] Ranganathan, G. "An Economical Robotic ArmPlaying Chess Using Visual Servoing." Journal of Innovative Image Processing (JIIP) 2, no. 03, 2020.

[17] Pandian, Dr A. Pasumpon. "Recognition Aid for Visually Challenged to Make Out Indoor Environment." Journal of Artificial Intelligence and Capsule Networks 2, no. 1: 11-19.