Motion Detection Using HD Camera of Microcomputer Raspberry Pi

Zoltán Balogh, Štefan Koprda, Milan Turčáni
Department of Informatics, Faculty of Natural Sciences,
Constantine the Philosopher University in Nitra
Tr. A. Hlinku 1
940 11 Nitra, Slovakia
zbalogh@ukf.sk, skoprda@ukf.sk, mturcani@ukf.sk

***Abstract* — Computer vision, as a science and technological discipline, deals with the ability of the electronic devices to obtain information from a digital image – "to understand the situation" and thus to make a decision or complete a task. It belongs to the technology used in all fields of industry and research. The analysis of the current status, general descriptions of microcomputers and the particular type of Raspberry Pi are introduced in the first part. The concepts of computer vision and OpenCV bundle that has been used in the designed system are described in the analysis. Possibilities of the system use in smart homes as one of the ways of its application in practice are also highlighted. The second part deals with the hardware design and it describes the software code that has been written in C++ and its main function is face recognition in the household. This kind of system may fully serve as a part of Internet of Things.**

***Keywords* — Raspberry Pi, Internet of Things, OpenCV, Face recognition, Motion detection.**

I. INTRODUCTION

In the last few years, the accessibility and the precision of face recognition have increased rapidly. There are a rather big number of systems that are capable of face detection and recognition. Some of the systems are, for example, Picasa, OpenCV, iPhoto. Picasa and iPhoto are similar and primarily they are used to detect faces in pictures. On the contrary, OpenCV has been created by community and contains the most used methods and algorithms for face detection and recognition. It is important to realize that this tool is not a standard system but it is a collection of methods that can be used for this purpose. It has inspired the use of the library on the micro PC Raspberry Pi. Certainly, for the complex functional system design from hardware to software, it was necessary to analyse first the most often used micro PCs and to compare their features.

Built-in computer vision is a rather practical department of computer vision that is interested in its algorithm development or modification running on built-in systems of small mobile computers. There are two main reasons for systems of built-in computer vision – a reasonable use of computing power and low battery consumption. As an example of this kind of built-in system, Raspberry Pi can be described. It is a small open-source micro PC that has gained popularity among fans and researchers because of its simple operation, universal abilities and surprisingly low price even despite its expansion and quality [1].

Raspberry Pi is a micro PC mounted by the processor ARM that can also be used as a full PC. It might not be a first of its kind, but it is different from the others by its price. As for operation system, it is possible to use some of the Linux distributions adjusted for ARM processors, for example, Debian Squeeze, Arch Linux ARM, Fedora Remix, Raspbian or even Windows 10. The PC is being developed by a non-profit organization Raspberry Pi Foundation. Besides other things, it is supported by the University of Cambridge and Broadcomem. The project's goals are to extend the computer science into schools. The idea of a small and cheap computer for children was born in 2006. At schools, children are mostly taught to work with Word, Excel, to create a website but they were missing an accessible product on which they could be taught how to programme. Basically, there was a need to replace computers Amiga, BBC Micro, pectrum ZX and Commodore 64 which were used to teach programming for the past generations [2].

The most interesting way how to use Raspberry Pi is to use it as a project platform [3] [4]. Thanks to its low consumption it can be used as a controller of a smart house [5]. Task list that can be reached depends only on the imagination and the creativity of the user. Besides, GPIO pins, other ports and connectors enable operation and management of all electronic peripherals including sensors, motors, buttons, etc. Digital inputs of Raspberry Pi can be used for analogue sensor reading of quantities such as temperature, distance, level of lighting or motion sensor status. It is possible to carry out these kinds of tasks with the use of analogue signal converter to digital (ADC). Other peripheral devices (CNC, robots [6]) can be connected to GPIO pins directly on Raspberry Pi board or through the local wireless network. It is possible to add a wireless connection to the Internet with a simple USB Wi-Fi adaptor. Usually, the adaptors need a great deal of energy, so it is better to connect them through an externally powered USB hub [7].

Raspberry Pi is supplied also as a separate board with multiple connectors. For starting up, it is necessary to load the distribution on a micro SD memory card (it replaces the hard disc), connect the power supply in form of micro USB and alternatively other peripherals. The mouse and the keyboard can be connected into USB ports. To connect the screen, there is the HDMI connector which supports audio transfer. As a primary audio input/output, there is a 3.5 mm jack connector [8].

Since the paper is focused on visual detection, the designed system needed a camera with high quality. The creators of Raspberry released their own camera board to process image with the application of computer vision. The board is small and weights 3 grams, but it contains 5 megapixel CMOS scanner. For connection, flat cable with a CSI connector is used [1]. It is a camera with HD definition. The definition can be adjusted; it can scan an image with higher FPS in lower values. It has a 5-megapixel lens, and its price is around 24 €. With the camera and the accessible libraries, it is possible to use the micro PC for visual detection.

II. Visual Detection

In the last decade, visual detection and face and object recognition have become the biggest challenges of computer vision. The potential use of detection systems and object recognition is rather wide, starting from security systems (identification of authorized users), medical techniques (detection of tumours), industrial applications (visual inspection of products), robotics (navigation of a robot in an inaccessible terrain), to augmented reality and many others. It is possible to approach the task of detection in various levels of generalization. Either a particular instance of each object is detected, or a whole category of objects. Visual detection and face recognition are a rather complex problem. A few examples:

- inhomogeneous and changeable scene lighting,
- changing spectral characteristics of lighting and consequently the colour of the scanned object,
- objects that overlap,
- various angles of camera objects,
- in the case of category recognition, there is a challenge of variedness of objects in one category.

The main chain of the system operations of recognition detection can be shown as follows:

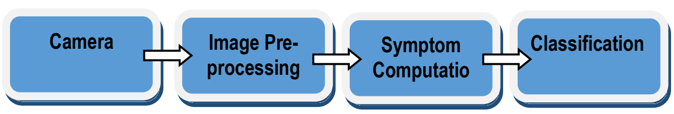

Fig. 1 Visual detection diagram

At the entry point of the chain there is the digital camera from which the image signal is gained, be it colourful or black-and-white. The task of the image pre-processing is to remove the undesirable and for next processing disturbing phenomena such as signal noise, information highlighting (e.g. object contour) that is relevant, but on the contrary, information suppression that is not relevant from the task's point of view. A good example can be a filtration to remove noise, upper-permeable filtration to remove non-homogeneity of the lighting, transition to other colour space or morphological processing. The goal of the next processing level is to gain the symptoms that will represent the image information in a sufficiently representative and discriminative way. If the task is to detect a human face in an image, then the whole image will be passed by a so-called sliding window. In all the positions, the task of the classifier will be to decide between two classes – whether the face is or is not in that part of the image which is being investigated. For the classifier, the input is the vector of symptoms itself according to which the classifier can decide. In the designing process, it is possible to generate the vector of symptoms that is larger than that one used in the final classifier. The relevant symptoms will be determined by methods of symptom selection. In the next decision level, the classifier decides about the class membership according to the symptom vector.

A. Face recognition procedure

Face recognition as one of the main biometric technologies is becoming more important considering the fast progress in digitalization of image (surveillance cameras, camera in a mobile phone), access to a big number of face images on the web, increased requirements for higher security. The technology of facial recognition has developed since the creation of Eigenface. In limited situations, e.g. when it is possible to control the lighting, the position, the appearance and the facial expressions, the automated face recognition may overcome human decision making. Mainly it is possible if the database (gallery) contains a big number of images. The automated face recognition is still facing a lot of challenges, for instance when the face images are gained through an automated process (Stan, 2011). Scenarios for face recognition might be divided into two types, face verification and face identification. In publication Face Recognition Vendor Test (FRVT) 2002, [9] [10] another scenario has been defined under the name "watch list". It is a method that is testing all individuals whether they are or not in the database. The person is compared to all the others in the system database, and a similarity score is assigned for each comparison. The scores are in an order, so the highest similarity score is always the first. If the similarity score is higher than the predetermined threshold, the alarm goes off. In this case, the system thinks that the individual is in the systems database. There are two basic values of the *watch list* application. The first one is that a percentage of the cases when the alarm goes off when it identified the person correctly in the database. It is called the detection and identification percentage. The other one is the percentage when the alarm goes off for the individual who is not in the database. This is the false alarm percentage [11].

B. Factors influencing facial recognition

The human face has a big number of potential intraindividual deviations caused by factors such as:

- The head's position
- Lighting in interior or exterior
- Facial expression

- Object overlaps, e.g. clothes, glasses, etc.
- Facial hair
- Ageing

On the other hand, there are small variations between the subjects considering the similarities of each appearance. Figure 5 presents examples of appearance variants of one subject [11].

C. OpenCV

Open CV (Open-source Computer Vision, opencv.org) is the Swiss knife of the computer vision. It has a range of modules by which it is possible to solve a number of problems in computer vision. The most useful part of OpenCV might be its architecture and memory management. It provides a framework in which it is possible to work with images and videos in any ways. The algorithms to recognize faces are accessible in OpenCV library and are [12]:

- FaceRecognizer.Eigenfaces: Eigenfaces, also described as PCA, first used by Turk and Pentland in 1991.
- FaceRecognizer.Fisherfaces: Fisherfaces, also described as LDA, invented by Belhumeur, Hespanha and Kriegman in 1997 [13].
- FaceRecognizer.LBPH: Local Binary Pattern Histograms, invented by Ahonen, Hadid and Pietikäinen in 2004 [14].

The choice of Fisherfaces has been made because it is based on the algorithm LDA [15]. In their article when comparing different algorithms, with LDA a 95.3% success has been achieved, while the time needed for detection has been compared to the other algorithms. The method of Fisherfaces is taught from class transformation matrix. Unlike the method Eigenfaces, it does not record the intensity of lighting. The discriminatory analysis finds the needed facial traits for person comparison. It is necessary to mention that the Fisherfaces' performance is influenced by input data to a great extent. Practically, if the Fisherfaces teaching runs on a well-lighted image, then in experiments of face recognition with bad lighting, there will be a higher number of wrong results. It is also logical because the method does not have a chance to capture the lighting in the images. The Fisherfaces algorithm is described below.

X is a random vector from the c class:

$$X = \{X_1, X_2, \dots, X_c\}$$

$$X_i = \{x_1, x_2, \dots, x_c\}$$

Scattering of matrix S_B and S_W are calculated as:

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T$$

$$S_W = \sum_{i=1}^{c} \sum_{x_j \in X_i} (x_j - \mu_i)(x_j - \mu_i)^T$$

Where μ is the total mean:

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i$$

And μ_i is the mean of class $i \in \{1, \dots, c\}$:

$$\mu_i = \frac{1}{|X_i|} \sum_{x_j \in X_i} x_j$$

Fisher's standard algorithm is searching for projection W that maximizes the criterion of separability of the given class [16] [13]:

$$W = W_{fld}^T W_{pca}^T$$

These kinds of solutions are used in various industrial applications, in educational projects, and more often in intelligent households. Face recognition and motion detection have been used to construct a system into intelligent solutions in the households that can be used as a separate functional unit or as an element of a bigger system connected with a technology the Internet of Things.

III. Materials and Methods

The solution process of the work was divided into two main parts, a hardware construction with the necessary configuration and a control program creation. The main part of the camera was the board of Raspberry Pi and its module of the HD camera. The created application dealt with image processing.

A. Hardware construction

The system contained six hardware parts. The main element of the system was a micro PC Raspberry Pi 2. In the system, components were used such as: box designed by us, the basic board of Raspberry Pi, camera module, USB HUB, Wi-Fi adaptor, A/D converter, temperature sensor, speaker, power adapter, LED signal diode. The scheme of connection is shown in Figure 2.

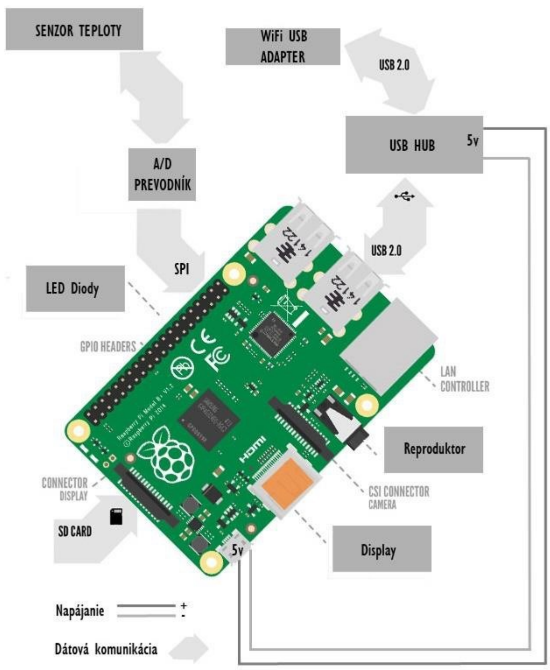

Fig. 2 Connection scheme of hardware components

Certainly, the main element of the solution was the micro PC Raspberry Pi 2 connected with the module of the HD camera. For Raspberry Pi, an operation system was necessary even before configuration. There was a need for a system that did not load the processor and the whole PC because the image processing itself in real time was performance demanding. The choice has been made for the most used OS Raspbian. This operation system is a specially modified and optimized version of Debian for Raspberry Pi.

B. Software design for Raspberry Pi

There are a number of languages that might be used in micro PC development. The most common are Python and C++. For Raspberry Pi, Python is recommended because of its simplicity. It is a scripting language on which programming on this PC is taught, since it contains English words that are easy to learn by children. This application has been created in programming language C++.

1) GUI of the program

The program for the graphical interface is called Qt Designer [17], and the environment in which it was possible to design the graphical parts of the program is shown in Figure 3.

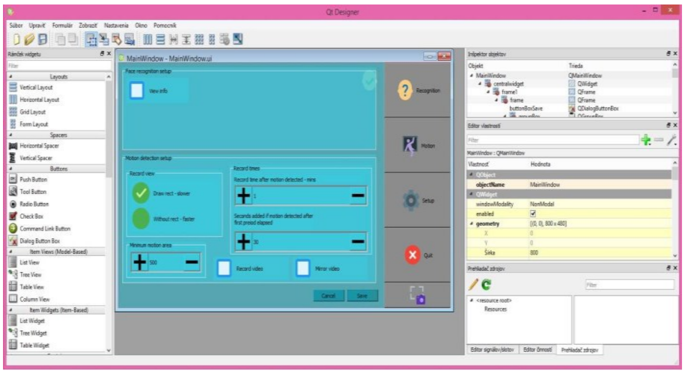

Fig 3. Development environment Qt Designer

The software offered the basic elements that could be laid in various layers. It was possible to import images and to create own graphic elements. To add own elements, it was also possible from the code itself, as well as to modify all the proposed parts directly from the code. After saving the proposition, the program created an xml file with suffix .ui and one file for the sources such as images with suffix .qrc. The file with properties and settings of the project Qt had a suffix .pro. The other parts of the project were formed of the standard files with functions and header files.

For GUI itself, a simple design has been chosen. On the right edge of the window, there were the control elements. The first button served for activating face detection. The next button under, activated the motion detection, then the button “Setup” opened the semi-transparent layer above the currently chosen mode. The settings were divided into two main parts, where it was possible to set the parameters and functions of both modes. Naturally, there was a button to turn off the program. As for the last button, it was possible to choose two modes; either it made a screenshot of the whole display or a picture of the current scene from the camera.

2) Program functions

The created program had multiple functions. As the main one, the person and motion detection was the main one. Motion detection consisted in the fact that when the program detected a motion by the camera, it marked the area on the screen where the motion appeared. It was marking it with a green square around the area. Its size depended on the size of the area in which the motion was detected. Certainly, there was an option of video recording in the case a motion has been detected. This option could be turned on in the settings of the program where the parameters of the recording could be set as well. It was possible to set the length of the recording after recognition. After reaching the end of it, the program decided whether to continue in the recording or not. In the case the motion was still present, it decided to continue. The length of the prolongation could be also set. In the left bottom part of the video, there was a so-called timestamp. It showed the date and the time and in the right part during the recording, there was an icon which showed that the current image was being recorded. Among the settings, there was an option of mirror image. So in the case the camera was turned towards the observer, the image was real as if in a mirror. It was also possible to turn off the video recording, as well as motion marking during recording. In the case this was turned off, higher values of FPS have been reached because the program did not recognize the motion during the whole recording but only when the set time has run out. The last option in settings for the program was the possibility to change the minimal surface of the motion being detected. After the recording ended, the program checked whether Raspberry Pi had an active connection to the Internet or not. In the case it had, it uploaded the saved video into cloud storage Dropbox. Motion detection is shown in Figure 4.

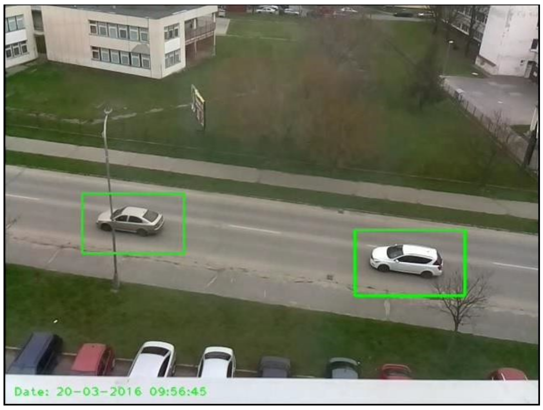

Fig 4. Motion detection

Another function of the system was facial detection and person recognition. The mirror image could be also set for this part as well. Besides, it was possible to display the information on the screen, particularly the actual FPS and the number of detected faces. The program recognised all the faces it detected and took into account only the bigger one. Logically, the one who occupied the bigger place in the image was the one the closest to the camera. After the person recognition and that the person was in the database, the program displayed the scanned face in the upper left corner. Under the image, there was the name of the person and Raspberry Pi greeted him. While scanning on the screen, all the faces that have not been recognized were marked with a square, usually with red colour. The recognized faces were marked with various other colours. Facial detection is shown in Figure 5.

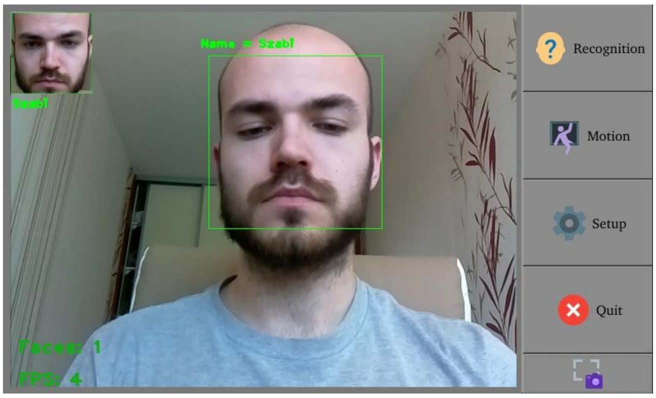

Fig 5. Detection of persons

3) Cloud storage – Dropbox

The program automatically uploaded the screenshots and the videos saved on an SD card of the micro PC into a predefined file. The first step was to create an account on the portal www.dropbox.com, and consequently an application had to be created in order to upload the files into the storage. The applications can be made on the website www.dropbox.com/developers/apps, and API has been chosen as first. In this case, it was the standard Dropbox API. Then access rights have been chosen in order to be able to upload files into one box. After this, it was enough to give the name of the program, and the application has been made. As a last step, a so-called access token has been generated for accessing the application without additional authorization. With this token, the program was able to upload the files into the box that has been made for the Dropbox application. In the source code of the program, a simple script has been assembled that was called up with the function `system()` afterwards. In the source code it was as follows:

```
void QOpenCVWidget::uploadVideo(string pathAndName)
{
	if (system("ping -c1 -s1 www.google.com"))
	{
		cout << "There is no internet connection   \n";
	}
	else
	{
		std::string str;
		str.append("curl -H ");
		str.append(1u, '"');
		str.append("Authorization: Bearer rBoj...bgTGXcub9s_UbkHa6UR-7DV");
		str.append(1u, '"');
		str.append(" https://api-content.dropbox.com/1/files_put/auto/ -T ");
		str.append(pathAndName);
		system(str.c_str());
		cout << str.c_str();
	}
}
```

IV. Discuss and Results

The result of the paper was a system for person recognition and motion detection. The created solution was completely constructed and suitable for use, for example, in smart houses. The system worked reliably and there were failures of hardware or software detected during the tests. The tests showed that motion detection recorded all the movements and in the case of marking the areas, it reached 4 – 5 FPS, without marking even 8 FPS. Since the person detection was a more demanding process, the achieved results were lower in display, around 4 – 5 FPS or even less. In the goals, requirements for action after detection have been mentioned. After detection, the program started to save the recording and after it finished recording, it uploaded it into cloud storage. In the case of face detection, the program greeted the recognized person. On the front side, the LED diodes signalized whether currently there was a person to recognize or not. For input demonstration, a temperature sensor has been applied that was displayed on the screen. It would be suitable to mention the price of the open-source solution, where the cost of the whole device was 160 €. The goal was to create a sophisticated system that was easy to spread at low cost on components. The goal has been achieved. During the tests, weaknesses have arisen as well. The system is not recommended to use when entering a building or similar situation. The system served mainly to increase the comfort in smart solutions where the level of false person detection was accessible. In the case of building entry, the requirements for precise detection are higher. For this kind of authentication, a precise detection is needed where the system cannot be fooled by a simple picture of the person. This solution is not able to meet the requirements. To solve this problem, a 3D detection use would be proposed, where the person's face would be scanned by multiple cameras, and a 3D facial model would be created from the images of the cameras. This kind of authentication would recognize the difference between a picture and a real face.

V. Conclusion

Face recognition and motion detection have been managed by the use of the proposed system. Similar systems have been used, for example, in smart houses. This solution is a cost-effective alternative for commercially available systems. It was easy to adapt to personal requirements of the users. The created system achieved good results and proved the possible use of OpenCV bundle on a not that much efficient device. The contribution of the work was to refer to the possibilities of visual detection using micro PC Raspberry Pi. Possibilities of using computer vision on a cheaper micro PC has been also pointed out. It has been proved that it was possible to use OpenCV library in less efficient systems by setting suitable criteria of optimization. Considering the software, the plan in the future would be to extend the settings option of the program, for example, the option of adding persons and profiles directly from the screen or from the web interface. It would be suitable to try the created software on multiple versions of the micro PC to compare the performance and the options of visual detection optimization. With a built-in Wi-Fi module, it would be possible to connect the device to the home network or to connect it with other devices using IoT. In smart houses it would serve as a web server through which all the devices that are connected to the network could be controlled.

There is a hope that this solution would be an example for other and number of such solutions would be increasing.

REFERENCES

[1] S. Brahmbhatt, *Practical OpenCV*, 1 ed.: Apress, 2013.

[2] V. Vujović and M. Maksimović, "Raspberry Pi as a Sensor Web node for home automation," *Computers & Electrical Engineering,* vol. 44, pp. 153-171, 2015/05/01/ 2015.

[3] Z. Balogh, R. Bízik, M. Turčáni, and Š. Koprda, "Proposal for Spatial Monitoring Activities Using the Raspberry Pi and LF RFID Technology," in *Wireless Communications, Networking and Applications*. vol. 348, Q.-A. Zeng, Ed., ed: Springer India, 2016, pp. 641-651.

[4] Z. Balogh and M. Turčáni, "Complex design of monitoring system for small animals by the use of micro PC and RFID technology," in *Mediterranean Conference on Information and Communication Technologies, MedCT 2015* vol. 380, A. E. Oualkadi, A. E. Moussati, and F. Choubani, Eds., ed: Springer Verlag, 2016, pp. 55-63.

[5] A. Grgurić, M. Mošmondor, and D. Huljenić, "Development of low cost energy efficient home sensing internet gateway: A pilot study," in *2016 IEEE International Black Sea Conference on Communications and Networking, BlackSeaCom 2016*, 2017.

[6] K. Janard and W. Marurngsith, "Accelerating real-time face detection on a raspberry pi telepresence robot," in *5th International Conference on Innovative Computing Technology, INTECH 2015*, 2015, pp. 136-141.

[7] P. Horáček. (2012). *Raspberry π I. – Úvod*. Available: http://www.linuxsoft.cz/article.php?id_article=1937

[8] P. Krushinitskiy and G. Sziebig, "Review of open source computing devices for iSpace in production workshops," in *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*, 2013, pp. 677-682.

[9] J. R. Beveridge, G. H. Givens, P. J. Phillips, B. A. Draper, and Y. M. Lui, "Focus on quality, predicting FRVT 2006 performance," in *2008 8th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2008*, 2008.

[10] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott*, et al.*, "FRVT 2006 and ICE 2006 large-scale experimental results," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, pp. 831-846, 2010.

[11] L. Xiaoguang, W. Yunhong, and A. K. Jain, "Combining classifiers for face recognition," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, 2003, pp. III-13-16 vol.3.

[12] A. Özdil and M. M. Özbilen, "A survey on comparison of face recognition algorithms," in *8th IEEE International Conference on Application of Information and Communication Technologies, AICT 2014 - Conference Proceedings*, 2014.

[13] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 19, pp. 711-720, 1997.

[14] D. Baggio, S. Emami, D. Escrivá, K. Ievgen, N. Mahmood, J. Saragih*, et al.*, *Mastering OpenCV with practical computer vision projects*: Packt Publishing Ltd, 2012.

[15] R. Rashmi and D. Namrata, "A Review on Comparison of Face Recognition Algorithm Based on Their Accuracy Rate," *International Journal of Computer Sciences and Engineering,* vol. 3, pp. 40-44, 2015.

[16] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," in *4th European Conference on Computer Vision, ECCV 1996* vol. 1064, B. Buxton and R. Cipolla, Eds., ed: Springer Verlag, 1996, pp. 45-58.

[17] Y. Wang and H. Peng, "Embedded intelligent home control system design based on ARM and wireless sensor networks," in *Chemical Engineering Transactions* vol. 51, ed, 2016, pp. 781-786.