IoT Solution for Smart Library Using Facial Recognition

Maulik Upala
Platform Application Engineer
Internet of Things Group
Intel Corporation, Halaman Kampung Jawa,
11900 Bayan Lepas, Pulau Pinang, Malaysia
maulik.upala@intel.com

WK Wong
Dept. of Engineering and Computer
Faculty of Engineering and Science
Curtin University Malaysia, CDT 250,
98009 Miri Sarawak, Malaysia
Weikitt.w@curtin.edu.my

Abstract—Internet has taken a giant leap ahead from ‘ICT’ to ‘Internet of Things’, making it possible to connect devices and transfer data with or without human intervention. IoT pillars such as devices, people, processes and data are connected with one another in a way that they can communicate with each other over internet. An attempt has been made to explain the potential impact of IoT on academic libraries and identify library areas where it can be implemented effectively. This paper outlines an initiative to develop an Internet of Things environment in academic library to support library space management such as study room occupancy service using IoT applications. This would enable a growing hands-on experience with the Internet of Things, using the security parameter of “face recognition” as face is a non-intrusive strong biometric for user identification within library management. This proposed solution is a step towards a smart library. The results and analysis show that the proposed IoT solution will accelerate the library rooms’ availability process without librarians’ intervention.

1. Introduction

Libraries are essential part of education system to improve our knowledge. Because of the rapid development in information and communication technologies, it has an impact on libraries. Internet of Things (IoT) is one of the most interesting concepts of recent years, which could potentially be a big challenge of libraries [1]. The proposed solution in this project can be considered asset intelligence system as it provides stakeholders such as students, staff and authorized users with a secured real-time view of library asset, to justify study room or conference room.

The IoT has excellent potential in the education sector to create smart library system, and open new path for research and feasible solution by interconnecting multiple devices to improve operational efficiency, real-time visibility and user learning experiences [2]. In year 2014, OCLC conducted the survey to understand librarians’ views and response to IoT solution for smart library [3]. In addition, according to survey they positively expressed their concerns about promising IoT technologies for libraries, feasibility, privacy, security and hacking which was the motivation to select this topic as a project.

This paper has 6 sections: section 2 explains the details of IoT based library, including problem description, definition and solution requirements. In section 3, IoT architecture of the proposed system,
IoT platform and IoT devices are introduced. Section IV describes the methodology to implement and deploy the proposed IoT solution. In section 5, Tests and results are analyzed to verify the solution meets requirements and expected outcome. At the end, section 6 briefs the paper, critical analysis and highlights of areas of future development.

2. The concept of IoT and smart library

IoT is a huge intelligent network that can establish connection between people, process, data and things, therefore; after the introduction of IoT technology in the library, the sensor technology embedded in variety of objects, can interact via wired or wireless communications, multimedia network system, and provide high-quality, thoughtful, convenient and efficient services to the library users [4]. Our proposed IoT solution will particularly focus on “Assistive technology”, “Access and Authentication”, “Resource availability for physical plant”. There are three IoT components in proposed system: (1) Hardware- compressed of sensors and embedded communication hardware (2) Middleware – Storage and computing tools for data analytics and (3) Presentation – easy to understand virtualization [5].

a. Problem Description

The proposed solution for the smart library has the potential to be the fastest, most efficient way to track and locate library services such as availability of study rooms with enhanced part in connectivity of the users to the library system that allows them to access to the system with higher level of authentication protocols. The main goal of this system is to overcome the issues such as finding the occupancy of library study-rooms and reducing manual effort. In addition, face recognition is introduced in the proposed system for access and authentication as face recognition-based application are nonintrusive, not replicable, makes human recognition a more robust and computerized process, and biometric data of the faces can be easily taken with available devices like camera [6][7]. In this paper, the specific objectives include:

1) Define the form of smart library service and create basic overview of the IoT based library services and functionalities. And explain general mechanism to deploy and hosting IoT based library.
2) Determine the broader IoT environment in present library system for the improvement of library services and development framework to demonstrate the procedures to establish IoT technology.
3) Describe IoT application in smart library using Open API under Thingspeak IoT platform. We develop a smart system for library users to provide access to IoT based library using the LBPH face recognition algorithm.
4) Formulate prospects for the use of the IoT system in smart library by providing secure real-time view to library users of available study rooms or conference rooms (library resources).

3. IoT solution requirements

a. IoT Architecture

The IoT system architecture for IoT solution designed into four logical level, including the physical level, communication level, system level and end-user level. We are considering the IoT architecture where the user is at the center and can develop or enable the data of the IoT solution using proposed architecture [11].
1) Physical level is the core layer of this IoT system which obtains the information about human presence with the help of the ultrasonic sensor, camera, single board computer (up squared) and others. Physical level is the brain of this IoT system which contains IoT devices and process collected information [5]
and consider as the bottom layer of the system whose job is to capture and change information to readable digital signals with IoT devices and sensors. All the information gathering and sensing was done within this layer [10].

2) Communication Level is the information transmitting center of the IoT system and conveys processed information through Internet and wireless network using Data-link, network, transport and session protocols [4].

3) System Level provides software environment of IoT platform (ThingSpeak*) that includes data management and application services [8].

4) End-User level is responsible for the presentation of data, monitoring and management of interface services [9].

b. IoT Platform

IoT platform is an entire suite of administrations that encourages administrations like development, support, analytics and in addition clever basic decision-making abilities to an IoT application. It also provides functionality to deploy and run IoT application as well as offers the necessary execution environment. The IoT application platform used for this project was ThingSpeak which enabled real time data collection, analysis and actuation with Open API. ThingSpeak also helped to build the open application (ThingSpeak channel) which worked upon the information collected by sensors from the board. Users of ThingSpeak platform can use ThingSpeak channel to send and store data. ThingSpeak channel was configured to save and represent data which were sent from any application or devices using the REST and MQTT APIs. Representational state transfer (REST) is an architectural style designed as a request/response model that communicates over HTTP. From REST and MQTT APIs, REST API calls were used to create and update ThingSpeak channels for the presentation of collected data in a chart form. In addition, the channel was configured to send real-time sensor data by posting a feed with a private API key for security purposes as no one can manipulate the sensor data.

c. IoT Devices

The cost of IoT solution can be reduced by using low-cost IoT devices such as the Up-Squared which is the low cost single board computer with an affordable price. Up-Squared with Linux based OS was used to develop a IoT solution as it has provision of connecting sensors, actuators, Ethernet for internet connectivity, multiple USB adapters and supports other peripherals. In this project, Up-Squared is used to configure the face recognition module as well as a controller to collect information from ultrasonic sensor, run the python program to interface sensor and ThingSpeak IoT application platform hosted on the private cloud.

The Sensor system detects and sense numerous conditions, data, and emits alert or response as per requirements. Ultrasonic Ranging Module HC-SR04 was used in this project to analyze and feed the captured data into the process of the system. HC-SR04 ultrasonic sensors exhibit different detection patterns, effective ranges, continuous detection and sensitivities. Thus, HC-SR04 module was used because for commercial buildings typically ultrasonic sensors are ideal for lighting control applications or object detection condition.

4. Implementation of the IOT solution

In methodology, defining the procedures to provide authentication using Face Recognition based on LBPH face recognition to the smart academic library allows users to avail the IoT facilities of smart library. The Figure 1 depicts the block diagram for the proposed IoT system. The following steps represent the method to show how the user can check the room occupancy using IoT in smart library.
1) LBPH (face recognition) recognize library user for physical entry into the smart library system

2) Once user face is recognized using LBPH algorithm then user will be authenticated to gain access to IoT application.
3) User queries for a room occupancy to the IoT application at ThingSpeak channel.
4) Up-squared board with the help of ultrasonic sensor checks whether any object/human detected in the study room or not.
5) If the object is detected, then board sends the sensor data of the object/human detection to the ThingSpeak cloud.
6) The user follows the instruction provided by face recognition system to reach room occupancy service IoT application over ThingSpeak channel.
7) IoT application visualize the sensor data in a graph format to authenticated user. The graph represents the distance of object from the sensor so if the distance value is greater than threshold then the room is available otherwise it will be considered as occupied.
8) ThingSpeak channel updates every 5 seconds and Up-squared with help of ultrasonic sensor send the object detection data to ThingSpeak cloud to keep real-time view of the IoT system.

There are certain assumptions made while implementing this project, which are the following:

1) The proposed library in the system was considered as smart library which has an electronic database of users.
2) The face recognition module would not recognize an unknown user. The users must be a part of the library management system. The maximum distance between user and camera should be 30cm and 0° frontal face angle.
3) The library administrator has IoT literacy to operate IoT applications.

a. Face Recognition using Local Binary Pattern Histogram

The original LBP method was presented by Ojala to be used in texture description [12] [13]. The essential technique, marks every pixel with decimal values called LBPs or LBP codes, to portray the nearby structure around of pixel. As showed in figure 2, estimation of the Centre pixel is subtracted.
from the 8 neighbor pixels’ values, if the outcome is negative the twofold binary is 0, generally 1. The count begins from the pixel at the best left corner of the 8-neighborhood and proceeds in clockwise heading. After figuring with all neighbors, an eight-digit paired value is delivered. At the point when this binary value is changed over to decimal, the LBP code of the pixel is created, and put to the directions of pixel in matrix using eqn 1 [14]. Formally, given a pixel at \((X_c, Y_c)\), the resulting LBP can be expressed in decimal form as follows:

\[
LBP_{p,R}(x_c, y_c) = \sum_{p=0}^{p-1} s(i_p - i_c)2^p
\]

where \(i_c\) and \(i_p\) are, respectively, grey-level values of the central pixel and \(P\) surrounding pixels in the circle neighborhood with a radius \(R\), and function \(s(x)\) is defined as

\[
s(x) = \begin{cases} 
1 & \text{if } x \geq 0 \\
0 & \text{if } x < 0 
\end{cases}
\]

However, the problem of LBP is that it cannot cover a large scale image so Ojala [12] revised the method to be flexible for any radius and any number of sampling points and came up with the new method called Extended LBP [13] as shown in figure 3 where \(P\) is the number of neighbors and \(R\) represents the radius of a circle.

The histograms of LBP are used for face recognition and it contains data about the distribution of local micro patterns. Normally, the face image is quite large for LBP calculation, so the image can be spatially divided where the image is in \(m\) small regions, \(R_0, R_1...R_{m-1}\) [15]. The histogram of each region \(j\) \(R\) is defined as

\[
H_{ij} = \sum_{xy} I(f_j(x, y) = i)1(I(x, y) \in R_j)
\]

Where \(n\) = number of different labels produced by LBP operator, \(m\) = number of rectangular blocks of the image and \(I(A) = 1\) or 0 depending on whether \(A\) is true or false.

Yang H. [15] proposed in his paper that we can train and allocate different weights for different face regions as few parts of face (such as the eyes, mouth) contain crucial information for face recognition. Thus, the regional histograms can be concatenated end to end to build a global description of the face.
By using this method, local pattern information with spatial information of the whole image together can be collected. Chi square statistic similarity measure is usually used to calculate the similarity of two histograms of two different faces to decide if two face images are belong to same person which can be defined as follows:

\[
X^2_{ij}(S, M) = \sum_{i,j} W_j \frac{(s_{ij}-M_{ij})^2}{(S_{ij}+M_{ij})}
\]  

where \(i\) represents \(0,1,...,n-1\), \(j\) represents \(0,1,...,m-1\), \(W_j\) is the weight for region \(j\), \(S\) and \(M\) are histograms of the target and query face images respectively.

LBP based face recognition algorithm was used to implement the facial recognition feature for the IoT solution. Face recognition is carried out in 3 stages which are: (1) Database Creation (2) Training face recognizer, (3) Recognition [16]. These 3 stages were implemented using python and OpenCV [17][18].

b. Authentication

Authentication is an extension to a face recognition mechanism. The IoT framework offers access to the user using their face. So, user-verification is required to be able to successfully gain access. According to the authentication feature implementation, when an user appears in front of the system, the system authenticates the user through the face recognition process as discussed earlier but if the user was not recognized by facial recognition procedure, then an functionality is added for the administrator which allows him/her to register the user to the library database by using the LBPH algorithm as well as adding the user manually using SQLiteStudio database manager, else any random user access would be rejected. SQLiteStudio is an advanced and cross-platform database manager with MDI interface, wizard dialogs, many useful tools and customizable interface. SQLite is the SQL-based relational database management system (RDMS), utilized by well-known organizations to provide local data storage on their mobile devices or host machines and currently the world’s most widely-used database. So immediately after the authentication procedure or registration procedure in case of new user, the smart library user is verified to gain access to the “Library room occupancy” IoT application over ThingSpeak channel. Face recognition algorithm (LBPH) and authentication features are implemented using python and SQLiteStudio for user interaction, with the physical world and gateway to the Thingspeak cloud, to avail the access of IoT application.

c. Access to Library Room Occupancy Service

The implementation of this technology utilized commercially available IoT devices such as ultrasonic sensor (HC-SR04) deployed in grid system as described in IoT architecture with single-board computer (UP2). According to the function of smart library, each sensor node assumed to be implemented at study or discuss rooms of the library and collected real-time information to single-board computer (UP2) and IoT gateway [22]. As discussed earlier, UP2 worked as middleware to collect and transmit the ultrasonic sensor values by running python script to detect distance and send that information to assign ThingSpeak channel from python script. To calculate the distance using python program, the difference between two timestamps recorded. The HC-SR04 sensor required a short 10uS pulse to trigger the module, which caused the sensor to start the ranging program (8 ultrasound bursts at 40 kHz) in order to obtain an echo response. To create trigger pulse, trigger pin (TRIG) set high for 10uS then set it low again. After then ECHO became high for the amount of time it took for the pulse to go and come back, so the code therefore measured the amount of time that the ECHO pin stayed high and each signal’s timestamp recorded in the correct order to measure distance [23]. Once the distance or sensor data was captured successfully then urllib2 python library used to send the data to ThingSpeak channel URL using private key of the assigned ThingSpeak channel.
d. **ThingSpeak Channel and Data Output**

As discussed earlier, ThingSpeak is an IoT platform that utilize channels to save data sent from any application or devices using the REST and MQTT APIs. To use these functionalities, ThingSpeak channel was configured first and then data was sent and received to and from the channel. ThingSpeak offers public and private view of the channels. In addition, to create channel and update its feed, update an existing channel, clear channel feed and delete a channel, multiple REST API calls such as GET, POST, PUT and DELETE were used. From REST and MQTT APIs, REST API calls were used to create and update ThingSpeak channels and charts. A web browser or client asks for the server utilizing a REST get back to and the server sends back with a response. Web browsers utilize this interface to retrieve web pages or to send information to remote servers. The REST API was used for this project because it was useful to use REST calls to update or retrieve any historical data, such as data within a defined time range data from a ThingSpeak channel and responsible to create, view, clear, delete channel and create chart to represent the real-time data.

5. **Test and results**

a. **Experiment 1-Face Recognition Accuracy**

For verification of user and evaluation of methodology, we conducted the experiment on sample face image for the different pose variation and set the camera at the approximated angles. For the data acquisition, simple web camera was used and 2 sample images of the same face for 9 different training angles were captured. Thus, the training imaging were natural images without imposition of any constraints neither on the targeted nor their surroundings [19]. Data acquisition of face images was quite challenging as the image taken from different angles was based on rough estimation and it had no control over the pose, expression, illumination and occlusion. In this experiment, the training set comprised rotation angles of 0°, 15°, 30°, 45°, and 60° in left and right directions, while the upward and downward face recognition were presented at rotation angles 10°. Consequently, the test data included 2 samples for the same face image for above mentioned poses of the user.

| Pose Variation of test subject (Degree) | Associated confidence for the predicted face |
|--------------------------------------|-----------------------------------------|
|                                      | Sample 1 | Sample 2 |
| 0° frontal                           | 57.27    | 59.29    |
| 15° Right                            | 75.61    | 79.02    |
| 15° Left                             | 71.00    | 73.37    |
| 30° Right                            | 92.57    | 91.09    |
| 30° Left                             | 102.00   | 93.34    |
| 45° Right                            | 123.00   | 118.09   |
| 60° Right                            | 179.00   | 161.37   |
| 10° Upward                           | 108.27   | 117.37   |
| 10° Downward                         | 105.67   | 107.34   |
The results are displayed in Table 1 and Figure. 4a), which represents the associated confidence for the predicted facial image where the accuracy of the facial recognition decreases with increasing confidence score. As we can see from the results, the confidence increased gradually when the pose variation of test subject or the angles increased. In second experiment, the face recognition was done at a different distance from the camera position but at 0° frontal angle which can be observed from table 2 and Figure. 4. Similar to the previous experiment, 2 sample face images of the same face were captured at 30 cm, 45 cm and 60 cm distance from web-camera to calculate the confidence variable. From this we noticed that the value of confidence increased as the distance between camera and the user increased. The best performance facial recognition is subjective to accuracy of the prediction confidence. Thus, the more the associated confidence is, the less is the accuracy of the recognition. So, according to attained results, the ideal position of the user for the accurate facial recognition was 30 cm away from camera and at 0° frontal pose variation.(refer to Figure 4 b)

| Distance from camera (cms) | Associated confidence for the predicted face |
|---------------------------|---------------------------------------------|
|                           | Sample 1 | Sample 2 |
| 30 cm                     | 69.49    | 70.42    |
| 45 cm                     | 92.29    | 95.99    |
| 60 cm                     | 115.10   | 108.98   |

b. **Experiment 2- Face Recognition Timing Performance**

This experiment tests the timing performance of the LBPH face recognition algorithm to measure the response time of the system. The parameters used for this test were the different rotation angles of the face such as 0°, 15°, 30°, 10° upward and 10° downward pose variations. As seen in table 3, we made experiments for different rotation angles on the same face image to check the robustness of the LBPH recognition algorithm. In addition, these experiments were performed and tested on Intel-based platform running at 2.5 GHz processor with 4 GB ram for all three experiments. The response time mentioned in table for three different experiment were captured real-time for face recognition and the average response was also derived to compare three different experiments. In experiment 1, recognition at 0° frontal face orientation made the best score out of three, especially by far to 30° orientation. But recognition at 10° downward angle also gave good result with 3.60 seconds.

Thus, like the recognition accuracy test, the ratio of response time of the algorithm increased as the
rotation angles increased. But overall LBPH was robust to change in the pose variation as the average difference between response times was very minimum. The results showed that LBPH gave the best timing performance but until a certain rotation angle. So, recognition at 0° frontal face angle makes best timing results.

Table 3. Timing performance Results

| Rotating angles of test subjects (degree) | Timing Performance / Response Time (seconds) |
|----------------------------------------|---------------------------------------------|
|                                        | Experiment 1 | Experiment 2 | Experiment 3 | Average Response Time |
| 0°                                     | 3.35         | 3.53         | 3.40         | 3.42                   |
| 15°                                    | 3.93         | 4.48         | 3.92         | 4.11                   |
| 30°                                    | 5.90         | 5.71         | 4.64         | 5.41                   |
| 10° Upward                             | 5.62         | 4.37         | 4.38         | 4.79                   |
| 10° Downward                           | 3.60         | 3.48         | 4.07         | 3.85                   |

c. **Experiment 3- Test on Public Database**

In this test, we conducted a comprehensive experiment using a large-scale face database to verify the performance of LBPH algorithm and system. Face database was trained using training stage of the LBPH algorithm which was suitable for testing the face recognition. 220 training samples of 11 individuals were used for testing. Basically, in this experiment the effects of varying-lighting, size, and head orientation were investigated using the complete dataset of 220 images. Multiple 11 groups of 20 images were selected and used as the training set. Inside each training set there was one facial image of each person and all were taken under the similar conditions of head orientation, image size and lighting and there was no separate set of test images taken under natural illuminations. The aim for this experiment was to demonstrate that with a sufficient set of training illuminations for each subject, LBPH algorithm indeed worked stably and robust with practical illumination, misalignment, pose, and occlusion [20]. All images in the facial database were then classified as being one of the eleven individual disregards the different poses. The independent variables were different in head orientation, size, illumination and light conditions while conducting final recognition procedure using recognition stage of the algorithm, however, the threshold was effectively infinite, so that no faces were rejected as unknown [21] and the recognition was successful.

d. **Experiment 4- Sensing and Communication Evaluation**

In order to verify the effectiveness of the occupancy sensor detection and integration to ThingSpeak IoT application, few tests and experiments were conducted. This section mainly includes testing methodology used to measure sensor data collection, real-time representation of collected data and demonstration of average execution data and time on ThingSpeak channel. As discussed in section IV, sensor data collected by ultrasonic sensor (HC-SR04) transferred by Up-squared and stored in ThingSpeak private cloud for further processing. To test the proposed human or object detection system in an actual situation. We developed experimental system as shown in figure 5. The test was done using ultrasonic sensor, and the Raspberry-Pi separately to validate the test result on different dates and time. Events were logged each time the object detected, or the sensor detected a change of
occupancy.

Figure 5 shows the data value of the sensor for two cases: (1) Occupied and, (2) Not occupied. However, we made the assumption while experimenting the module that the threshold of the detection system was 100cm so if the distance of the object or human being recognized was less than 100cm then the library room was considered as occupied else it was available. Although, the shortest time delay applied in this test was 3 minutes, it can be controlled by changing sensor mounting position, adjustment of sensor aiming angle and tuning of sensor sensitivity. After the testing of sensing technology and communication to Thingspeak channel completed, sensor data feeds log in JSON, XML, and CSV formats generated from ThingSpeak channel. The resulting logs have all the specific details about individual sensor data entry. Thus, the experimental results indicated that the effectiveness of the occupancy sensor detection and the communication module from IoT devices to ThingSpeak IoT application was promising.

6. Conclusion and future scope

Internet of Things has great potential for libraries and it is an emerging area were several possible services may become available as a result of an increasing inter-connected smart devices and networked environment. If the proposed IoT solution is implemented in the desired scope, it will capture desired results as discussed in section V and make a valuable addition to library services. The smart IoT technology in libraries is in the initial stage and currently evolving, therefore there is a need for librarians to be trained in this technology while the IoT technology is more accepted, adopted and available easily. However, it has certain implementation problems, such as the cost of IoT devices, construction and layout, technical standards and adaptability. However, the results of the IoT application demonstrated in paper, shows that IoT technology has broad development scope in libraries.

In addition, face recognition is a challenging problem in the area of image processing and computer vision when utilized as a security parameter. This paper contained a detailed methodology and implementation for face recognition module. We analyzed that LBPH has preliminary success in face
recognition problem and it achieved the highest accuracy in multiple training sets during experiments as mentioned in section V. Thus, the results of the experiment show that the proposed IoT solution for smart library is consistently efficient to enrich library services and user’s library experience. An IoT based library is going to bring in a plethora of changes to the library management system, specifically the way library connects with its patrons.

a. Future Scope

The approach described in this paper is successful as an initial stage solution and encourages IoT solutions for library and face recognition, however, more research is needed in the following domains:

1) IoT based libraries need to take various issues into consideration such as privacy and security of user’s information, as sharing this data publicly may lead to hacking and increase the cost of investment in IoT technologies in terms of time, money and manpower such as staff training. Furthermore, there is also an increasing decline in the use of physical libraries.

2) The information that is generated by the IoT application over the library system might aid collection developers better understand how patrons interact with physical space in library for service understanding or deeper automation.

3) With regards to facial recognition, size of dataset is to be expanded with multiple variations of illumination, pose, distance, expression and occlusion, so these conditions can be considered while capturing the images of the subjects.

4) Qualitative and quantitative data can be collected to learn what students like to do in the library space and provide deep understanding for library space management.

Furthermore, certain IoT technologies such as proposed in the IoT solution can be implemented to provide an IoT application for the library building to measure the number of visits, what users browsed, which library areas were occupied, and this solution can be implemented using inexpensive and simple sensors that collect the required data of the library usage. Adapting this IoT technology will allow library management to make strategic decisions and develop a more efficient and effective library environment.

References

[1] M. Wójcik, "Internet of Things – potential for libraries", Library Hi Tech, vol. 34, no. 2, pp. 404-420, 2016.

[2] M. Chang, "Building an Internet of Things environment in the Library", in vala 2016, Melbourne convention & exhibition center, 2016, p. 18.

[3] OCLC, Libraries & the Internet of Things, NextSpace, 24, 2015, viewed September 30, 2017, http://www.oclc.org/nextspace

[4] M. Sun, "The Research on the Development of Smart Library", Applied Mechanics and Materials, vol. 571-572, pp. 1184-1188, 2014.

[5] P. Renold and R. Joshi, "An Internet Based RFID Library Management System", in IEEE Conference on Information and Communication Technologies, IEEE, 2013, pp. 932-936

[6] N. Duc and B. Minh, "Your face is NOT your password Face Authentication ByPassing Lenovo – Asus – Toshiba", 2017.

[7] John D. Woodward, Jr., Christopher Horn, Julius Gatune, and Aryn Thomas,“A Look at Facial Recognition"
[8] "Microsoft Azure IoT Suite overview", Docs.microsoft.com, 2017. [Online]. Available: https://docs.microsoft.com/en-us/azure/iot-suite/iot-suiteoverview. [Accessed: 01- July- 2017].

[9] M. Mohammadi, "Building an Internet of Things environment in the Library", in vala2016, Melbourne convention & exhibition center, 2016, p.18.

[10] X. Zeng, S. K. Garg, P. Strazdins, P. P. Jayaraman, D. Georgakopoulos, and R. Ranjan, "IoTsim: A simulator for analysing iot applications," Journal of Systems Architecture, vol. 72, pp. 93-107, 2017.

[11] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Future generation computer systems, vol. 29, pp. 1645-1660, 2013.

[12] T. Ojala, M. Pietikainen and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," Proceedings of 12th International Conference on Pattern Recognition, Jerusalem, 1994, pp. 582-585 vol.1. doi: 10.1109/ICPR.1994.576366.

[13] T. Ojala, M. Pietikainen and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971-987, Jul 2002. doi: 10.1109/TPAMI.2002.1017623.

[14] D. Huang, C. Shan, M. Ardalibian, Y. Wang and L. Chen, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 41, no. 6, pp. 765-781, Nov. 2011. doi: 10.1109/TSMCC.2011.2118750.

[15] H. Yang and Y. Wang, "A LBP-based Face Recognition Method with Hamming Distance Constraint," Fourth International Conference on Image and Graphics (ICIG 2007), Sichuan, 2007, pp. 645-649. doi: 10.1109/ICIG.2007.144.

[16] A. kar, "Face Recognition Archives - The Codacus", The Codacus, 2017.[Online].Available:https://thecodacus.com/category/opencv/facerecognition/#.We9fzVuCzIU. [Accessed: 08- Sep- 2017].

[17] "Getting Started with Images — OpenCV 3.0.0-dev documentation", Docs.opencv.org, 2017. [Online]. Available: https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_gui/py_image_display/py_image_display.html. [Accessed: 02- Nov- 2017].

[18] "FaceRecognizer — OpenCV 2.4.13.4 documentation", Docs.opencv.org, 2017.[Online].Available: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html#create_facefacerecognizer. [Accessed: 14- Oct- 2017].

[19] A. Singh, "Comparison of face Recognition Algorithms on Dummy Faces", The International journal of Multimedia & Its Applications, vol. 4, no. 4, pp.121-135, 2012.

[20] A. Wagner, J. Wright, A. Ganesh, Zihan Zhou and Y. Ma, "Towards a practical face recognition system: Robust registration and illumination by sparse representation," 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, 2009, pp. 597-604. doi: 10.1109/CVPR.2009.5206654.

[21] M. Turk and A. Pentland, "Eigenfaces for Recognition," in Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, Jan. 1991. doi: 10.1162/jocn.1991.3.1.71.

[22] M. Sun, "The Research on the Development of Smart Library", Applied Mechanics and Materials, vol. 571-572, pp. 1184-1188, 2014.

[23] "HC-SR04 Ultrasonic Range Sensor on the Raspberry Pi", Cases for your Raspberry Pi, 2017. [Online]. Available: https://www.modmypi.com/blog/hcsr04-ultrasonic-range-sensor-on-the-raspberry-pi. [Accessed: 17- Aug- 2017].