

Software engineering model based smart indoor localization system using deep-learning

Zainab Mohammed Resan, Muayad Sadik Croock

Computer Engineering Department, University of Technology-Iraq, Baghdad, Iraq

Article Info

Article history:

Received Oct 12, 2019

Revised Feb 12, 2020

Accepted Mar 18, 2020

Keywords:

CNN

Deep learning

GPS

Indoor localization

Raspberry Pi

Robotic car

Software engineering

ABSTRACT

During the last few years, the allocation of objects or persons inside a specific building is highly required. It is well known that the global positioning system (GPS) cannot be adopted in indoor environment due to the lack of signals. Therefore, it is important to discover a new way that works inside. The proposed system uses the deep learning techniques to classify places based on capturing images. The proposed system contains two parts: software part and hardware part. The software part is built based on software engineering model to increase the reliability, flexibility, and scalability. In addition, this part, the dataset is collected using the Raspberry Pi III camera as training and validating data set. This dataset is used as an input to the proposed deep learning model. In the hardware part, Raspberry Pi III is used for loading the proposed model and producing prediction results and a camera that is used to collect the images dataset. Two wheels' car is adopted as an object for introducing indoor localization project. The obtained training accuracy is 99.6% for training dataset and 100% for validating dataset.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Muayad Sadik Croock,
Department of Computer Engineering,
University of Technology-Iraq,
Alsinaa Street, Baghdad, Iraq.
Email: Muayad.S.Croock@uotechnology.edu.iq

1. INTRODUCTION

Localization systems are recently appeared to provide the information of location of different objects, such as people, animals and things. Location finding is normally acquired by using global positioning systems (GPS) for outdoor environments. Since GPS has a lack of the sight line to satellite and a lot of other obstacles, it is not working for indoor environments [1]. The localization is very important for the paralyzed people, and their families as it eases the live for them and displaces their concerns. In addition, it saves time of searching for the disabled persons if they were lost for any reason.

In [2], the authors studied finger printing-based indoor localization in commodity 5-GHz Wi-Fi networks. They proposed a system called BiLoc, which uses bi-modality deep learning for localization in the indoor environment using off-the-shelf Wi-Fi devices. The experimental results validated the superior performance of BiLoc over several benchmark schemes. In [3], the authors designed an indoor localization system based on crowd sourcing, which can construct the map of WiFi radio using its data, the map of indoor location is turned to indicative map, the authors show results that their proposed network can minimize the variety problems caused by changing environment. In [4], a convolutional neural network (CNN) model for indoor localization in multi-floor buildings was produced using WiFi received signal strength values. These values have been taken from the access points in wireless LAN. The results for this

model showed accuracy for training model about 100%. In [5], the authors used pervasive Wi-Fi and attractive field data for indoor limitation. Exploratory outcomes demonstrated that profound network models consolidating attractive field and Wi-Fi finger printings improved indoor localization precision. The training phase of deep positioning was computationally intensive, the testing phase was fast and suitable for real time indoor localization. In [6], the authors designed WiFi deep, which was a Wi-Fi -based indoor fingerprinting localization system that can achieve robust and high accuracy tracking in the presence of device heterogeneity. The system employed a model regularization to enable the network to generalize and avoid over-fitting, leading to a more robust and stable models. The results showed that the WI deep comes with a localization accuracy better than the state-of-the-art systems by at least 53% and 29.8% in the large and small environments respectively. In [7], they proposed a model that learns profound learning scene acknowledgment technique. This depends on limitation improvement utilizing the technique for exchange learning on inception V3 network. Model feature information was added to assist in scene recognition [8]. In [9], The authors utilized an advanced mobile phone-based movement acknowledgment for indoor restriction utilizing a convolutional neural system. These exercises can be utilized as the tourist spots for indoor localization. These exercises could be utilized as the tourist spots for indoor confinement. Another convolutional neural system has been intended to become familiar with the best possible features consequently.

In this article, an indoor object localization model is proposed based on deep learning [10] and deep CNN [11]. The proposed model contains two levels. First level, or software level, is used for collecting the images dataset, done using raspberry pi III camera. This camera is fixed on a simple robot of two wheels' car. The dataset is collected and prepared by gathering and pre-handling images information. It uses as input for the proposed model. The model is trained until it reaches to a satisfying accuracy for both training and validating dataset. Then the model is saved. In the second level, or can be called as hardware level, is used for gathering and controlling the electric robot car that carries on the raspberry pi III and its camera. This camera is not only used for preparing the dataset, it is also used to take a real time image when a certain condition happened or when an authorized user is asking about an object's place. In this case, the raspberry pi III captures the image and send it to the proposed inside the Raspberry Pi. Then, the proposed system waits for the response. The prediction of the model with certain accuracy is supposed to be the prediction of the place itself [5, 12] and [13].

2. PROPOSED SOFTWARE ENGINEERING MODEL

It is well known that the provided algorithms are required to be more reliable and have all the ability of being expanding and flexible. In this work, a software engineering model is proposed as a base stricture for the utilized algorithm. This is to ensure the reliability, scalability and flexibility of the proposed algorithm. The mentioned algorithm is concepts of performing the suggested method of detecting the affected areas in the brain. Figure 1 shows the block diagram of proposed model that explains stages of method as a work flow.

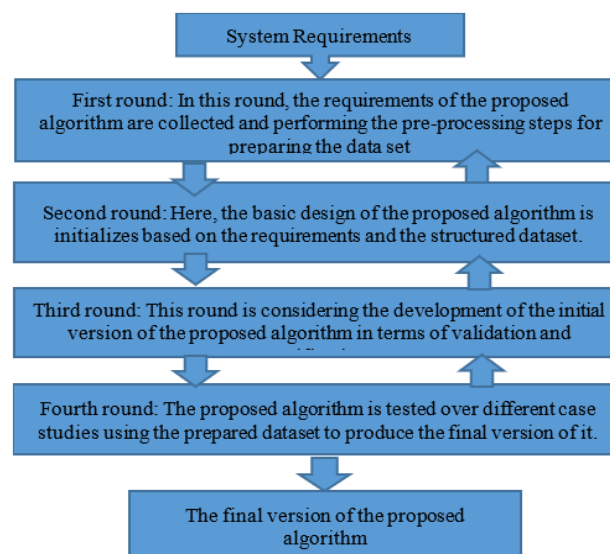


Figure 1. Block diagram of the designed software engineering model

It is shown from Figure 1 that the system requirements are collected by the first round for preparing the structure of the designed algorithm. At the second round, the initial design of the suggested algorithm is performed based on the system requirements. Different levels of developments have been done in the third round depending on the initial design of the proposed algorithm in iteration way. The last version of the proposed algorithm is tested in the last round for ensuring the reliability, expandability and flexibility of the it.

3. PROPOSED SYSTEM

As mentioned earlier, the proposed system clearly contains two different phases, designed based on the proposed software engineering model. The first one, can be called as the offline phase, which contains all the programming algorithms starting with creating the dataset and processing it, ending with saving a well-trained model. The second phase, or the online phase, in which, a real time images are taken, security where localization can significantly improve security conditions the world over. Client versatility examples and communication can be utilized to distinguish conceivable dangers that may present security risks. Similarly, in war zone, the military can follow its advantages through a restriction framework that can improve the general task and increment the odds of fruitful activity. The final classification is then assigned to its estimated location. As shown in Figure 2 [2].

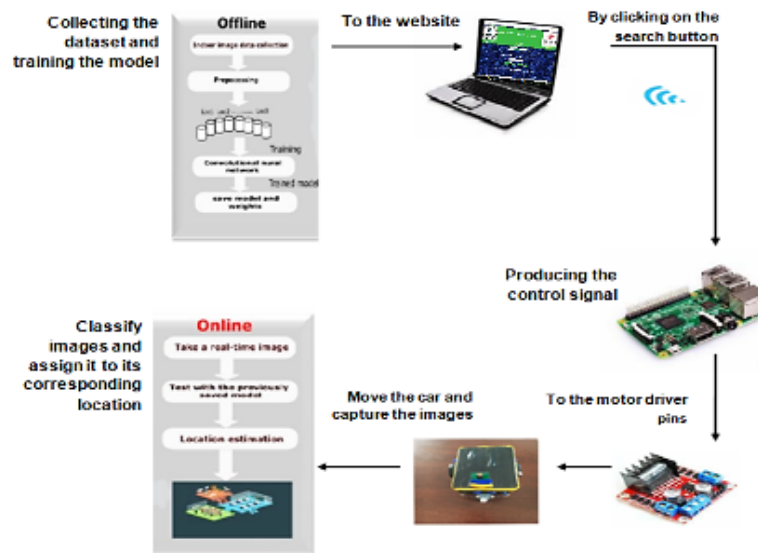


Figure 2. Proposed system block diagram [2]

3.1. Proposed methodology

In this section, the dataset is collected and prepared. It also introduces the CNN model algorithm and training procedure. Moreover, this section provides the website design that is based on the proposed algorithm. At the other side, the object localization system is introduced with the related proposed algorithms and flowcharts. The proposed overall system is started with the software part and followed by the hardware part as shown in Figure 3. The following steps describe the method used to build the model briefly:

- Starting the collection of a dataset for training the model. This dataset must cover all the desired areas that is wanted to be added to the classification model.
- Preparing this dataset by adding the necessary operations, such as cropping the images to be all of the same size. Moreover, adding a flipped copy of the same images for increasing their number in each class.
- Designing the CNN layers of the model with Keras library. Then, the system trains the model until reaching the desired accuracy and save them in the Raspberry Pi for later use.
- The software part ends with creating a simple website that contains the controlling buttons on the hardware parts. This website is built by using the HTML, PHP, CSS and python scripts.
- The first step in the hardware part contains gathering and building a simple prototype of a robotic two wheels' car with its battery and motor driver.
- Combining the Raspberry Pi and related accessories that include camera, LEDs, breadboard, voltage step regulator, two ultrasonic sensors. All these components are connected together and placed in a black box for arrangement.

- The Raspberry Pi is the core of controlling the hardware part. It is connected by WiFi to the website. When the search button is clicked in the web, a certain procedure is followed by moving the car and capturing real time images. This procedure is introduced in details in the next sections.
- The last step is finding an appropriate classification accuracy and assigning it to its class that is related to its corresponding classified indoor location.

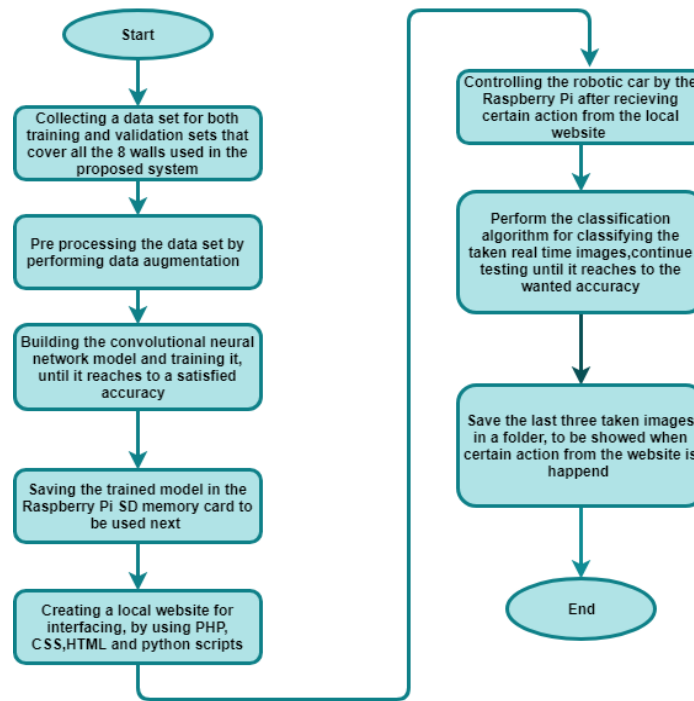


Figure 3. The proposed system methodology flowchart

3.1.1. Dataset collecting

The dataset is chosen very carefully, with notifying the angle of capturing the images and the number of images that covers the included areas that are needed to be classified. In these images, different situations and different positions are taken in consideration, as shown in Table 1. In this part, the data set collection and choosing are discussed. Moreover, as well as the preprocessing operations, such as cutting, zooming, and shearing are also introduced. The camera is located about 20 cm from ground on the robotic car with vertical angle of 15 degrees. After all datasets are collected, each class is assigned to one room. This is to reduce the complexity as it is not sensory to include all walls in the dataset because some images in the collected dataset are taken in a way that covers more than one corner inside the room.

Table 1. Overview of the collected dataset

Location (Wall) No.	Corresponding Room Name	Number of Training dataset	Number of Validating dataset	Size Of Images (width x height)
Location 1	Meeting Room	264	60	480 x 270
Location 2	Meeting Room	268	61	480 x 270
Location 3	Eating Room	255	59	480 x 270
Location 4	Eating Room	260	60	480 x 270
Location 5	Admin Room	208	54	480 x 270
Location 6	Admin Room	202	60	480 x 270
Location 7	Admin Room	298	67	480 x 270
Location 8	Eating Room	267	61	480 x 270

3.1.2. Dataset preparing

The next step is to prepare these datasets to be ready for use in the training model. The data augmentation is used as a tool that is available in any deep learning toolbox. It is done with Keras library

using a class called image data generator [14]. This class applies a random invariance, such as images cropping, scaling, shearing, and flipping as shown in Table 2.

Table 2. Overview of the used data augmentation techniques

Name	Value
Scaling	(1./255)
Zooming	20%
Flipping	Horizontal
Crops	224*224

3.1.3. Model training

The model is trained from scratch, because there is no pre-trained model used for initialization. The main source of data is the images available with Raspberry Pi camera. The proposed CNN with deep learning is designed to train a number of images and test others [15] and [16]. Each entered image to the CNN model passes through many levels of convolution layers. Some of them contains a filters and pooling, while the others include fully connected layers. Finally, such an image passes through a soft max filter [17]. The following steps illustrate the building of the CNN model:

- In the first convolutional layer, 20 filters of size 5x5 pixels are applied to the input image of size 224x224, followed by a rectified linear unit (ReLU). A max pooling layer is used for taking the maximal value of 2x2 regions with strides of 2x2.
- The output of the previous layer is then processed by the second convolutional layer. It contains 50 filters of a size 5x5 pixels. Again, that is followed by a ReLU and a max pooling layer.
- This convolutional layer's output is passed to a fully connected layer and contains 500 neurons, followed by a ReLU.
- At last, the yield of the last completely associated layer is nourished to a soft max layer that appoints a likelihood for each class of 8 classes.

Then it assigns the class number to its corresponding location in one of the three rooms mentioned earlier, with a probability goes from 0 to 100%. As shown in Figure 4.

Training the CNN model is the next step, see Table 3, it takes a time ranges from 88 to 94 second for each epoch. This is because the deep learning model training requires a high ability GPU. The proposed model is trained online in Google Colab, which gives a free GPU on a free cloud service for developing applications of deep learning [18] and [19]. This is done by loading the dataset to the Google drive for opening the Google Colab and making the model. Training tries are done with different number of classes from 4 to 8, in both laptop and online on Google Colab, but there is no need to mention the results of these tries. The important thing to know is that, it has been reached to a point in which only small number of epochs is enough to train this model with satisfying results. Therefore, only 5 epochs are taken each of which produces different accuracy and loss metrics for both training and validating dataset.

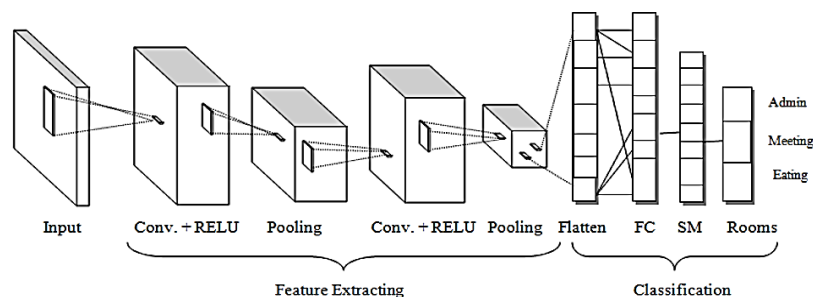


Figure 4. CNN architecture of the proposed model

Table 3. Model training results corresponding to each epoch

Epochs	Time in seconds	Training		Validation	
		Accuracy	Loss	Accuracy	Loss
1	94s	0.7923	0.7252	0.9146	0.2176
2	89s	0.9836	0.0785	1.0000	0.0301
3	90s	0.9960	0.0276	1.0000	0.0180
4	88s	0.9965	0.0170	1.0000	0.0096
5	90s	0.9965	0.0152	1.0000	0.0145

3.1.4. Website design

After the model had been trained and reached to a satisfying accuracy results, which was 99.6% for training dataset and 100% for validation dataset. This model is saved to the RPi III memory to be used in the hardware level. The saved model cannot be used directly for predictions, because it is saved as a python script. So, an interfacing mechanism must be added for the user to be able to contact with the model in some way. Figure 5 shows a flowchart for the designed website, which contains the steps that must be followed to operate this system.

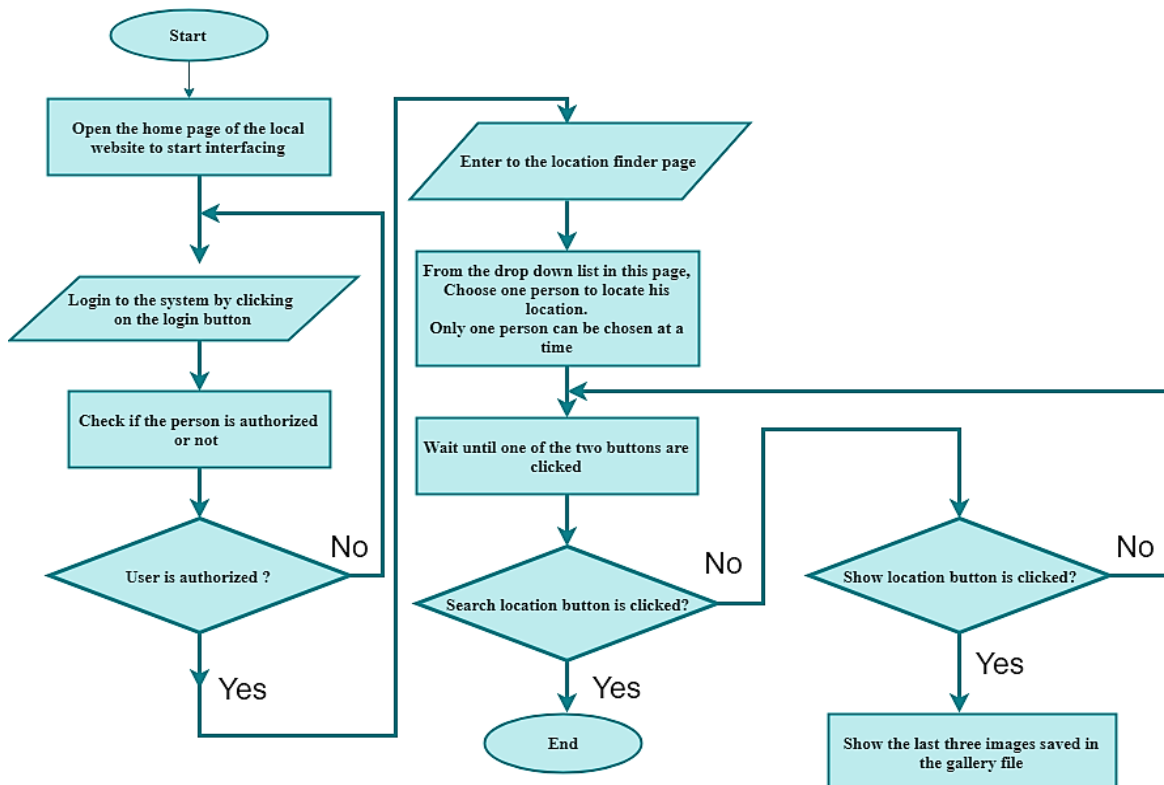


Figure 5. Website flowcharts

The following steps are described in the next lines:

- A simple web page is created inside the RPi. For security reasons, this web page contains a login button at the starting home page.
- Without login, the normal user cannot see more than simple description on the project and the copy rights and the year of modification.
- When the user fills up the login form, and he/she is authorized to enter to this web. The web then moves the user to the indoor localizer page.
- The indoor localization page contains number of persons listed, and followed by two buttons; the search and showing buttons.
- The user should select one object at a time to search for its place. In this paper, only one object is available but for the future work, this project can cover large number of them.
- After choosing the person's name, the user should click on search location button, which contains the hardware controlling and model prediction results.
- The other button, location button, displays the captured images that had been interred to the model for testing reasons.

3.2. Hardware integration level

This section clarifies how the equipment parts are gathered, associated and utilized together as a total working framework for indoor item restriction robot. It contains two levels, the initial segment is connected for structure the vehicle segments, while the subsequent level is in charge of the controlling system for the vehicle and the rest segments.

3.2.1. Car parts gathering and arrangement

The principle segments utilized for setting up the mechanical vehicle are appeared in Figure 6. These parts are Raspberry Pi III model B, the L298N motor driver, the ultrasonic sensor, the voltage steps down regulator, the Li-Po battery, and the RPi camera [20-25].

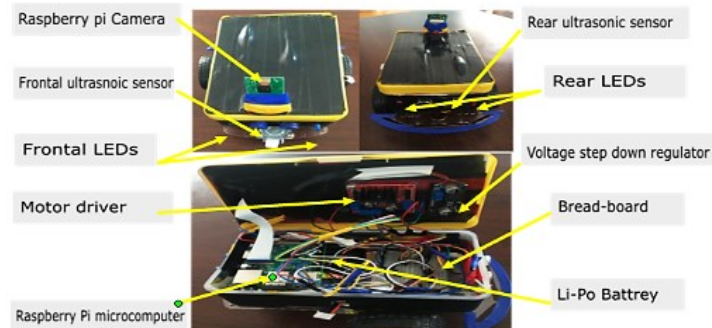


Figure 6. The hardware requirement for car module

3.2.2 Robotic car movement and obstacles detection

Two ultrasonic sensors, fixed at the front and back of the car, are used to detect the obstacles in the car's passage way. The detection is accomplished by measuring the distance between the car and the faced obstacles. It is important to note that the software algorithms are designed based on the proposed software engineering model. The proposed algorithm of car movement and obstacles detection is shown as a flowchart in Figure 7 and the next lines describe it in details:

- For forward movement, the RPi fully stops the car, if the measured distance is smaller than or equal to (100 cm). The car is kept waiting for 5 seconds (secs).
- After 5 secs, again it checks if the measured distance is increased. This means that the obstacles are moved, then the Raspberry Pi orders the car to drive again, until the measured distance is smaller than or equal to (100 cm).
- In case of the 5 seconds are passed and the distance still the same, this means that this obstacle may be a wall or any stable object.
- Now, the car is stopped, and it starts the procedure of capturing and classifying the images.
- If the measured distance is smaller than (80 cm), this means that the robotic car is too close to the wall, is to take an image in a clear way the car must drive back at least for 20 cm, which can be equipped by moving the car backward for (1 sec).
- For backward movement, The Raspberry Pi fully stops the car, if the desired time for backward movement is passed, or if the measured distance is below than or equal to (30 cm).

3.2.3. Classifying images procedure

After the robotic car has been reached to the wall successfully, the Raspberry Pi starts in capturing images. This procedure is shown in Figure 8 as a flowchart that explains the proposed algorithm. The image classification algorithm is proposed based on the software engineering model. The following lines are presented to describe this algorithm in details:

- The first step in this procedure is to load the previously saved model in the RPi as a python script, by a special PHP formula in the website. After the user clicked on the search location button, and the robotic car found a wall (or as it thinks, it is a wall).
- The RPi orders the camera to take the first image and save it in gallery file inside it.
- Then it turns the car to the left with a 15 degree for changing the image angle, and again the camera takes another image and save it with the first one.
- Again, the RPi turns the car back to its original location by turning it to the right with the same degree (15).
- Then it moves the car to the right with the same mentioned degree, for taking the last image in another different corner.
- At this point the system has three different images for distinct corners. These images enter to the loaded model as real time test images.

- Each image is passed through the loaded model and reach to a certain classification accuracy. The obtained three accuracies from three images are compared with each other. The maximum accuracy of these accuracies is chosen as the actual result of the classification.
- In the proposed algorithm, a desired accuracy is considered to be 70%.
- If the resulting image classification accuracy is equal or higher than this desired accuracy, then the classification is accepted and the image is an actual wall. Then this image should be assigned to the corresponding room and takes it as the final result. Otherwise, the RPi ordered the car
- to turn to another wall. This is done by moving it to the left with 90 degrees. Then the procedure is started from the beginning, starting from robotic car movement and obstacles detection.

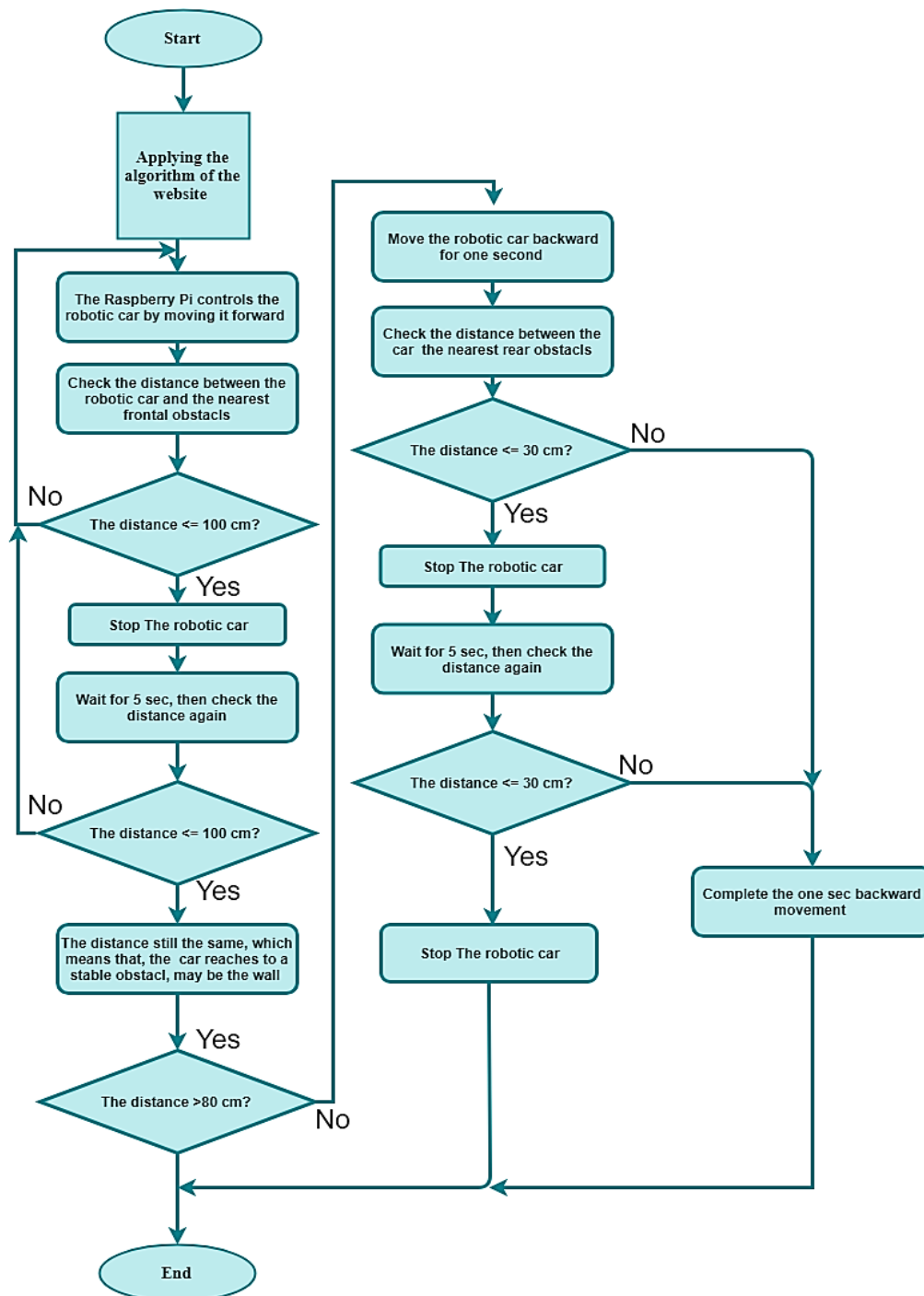


Figure 7. Robotic car movement algorithm

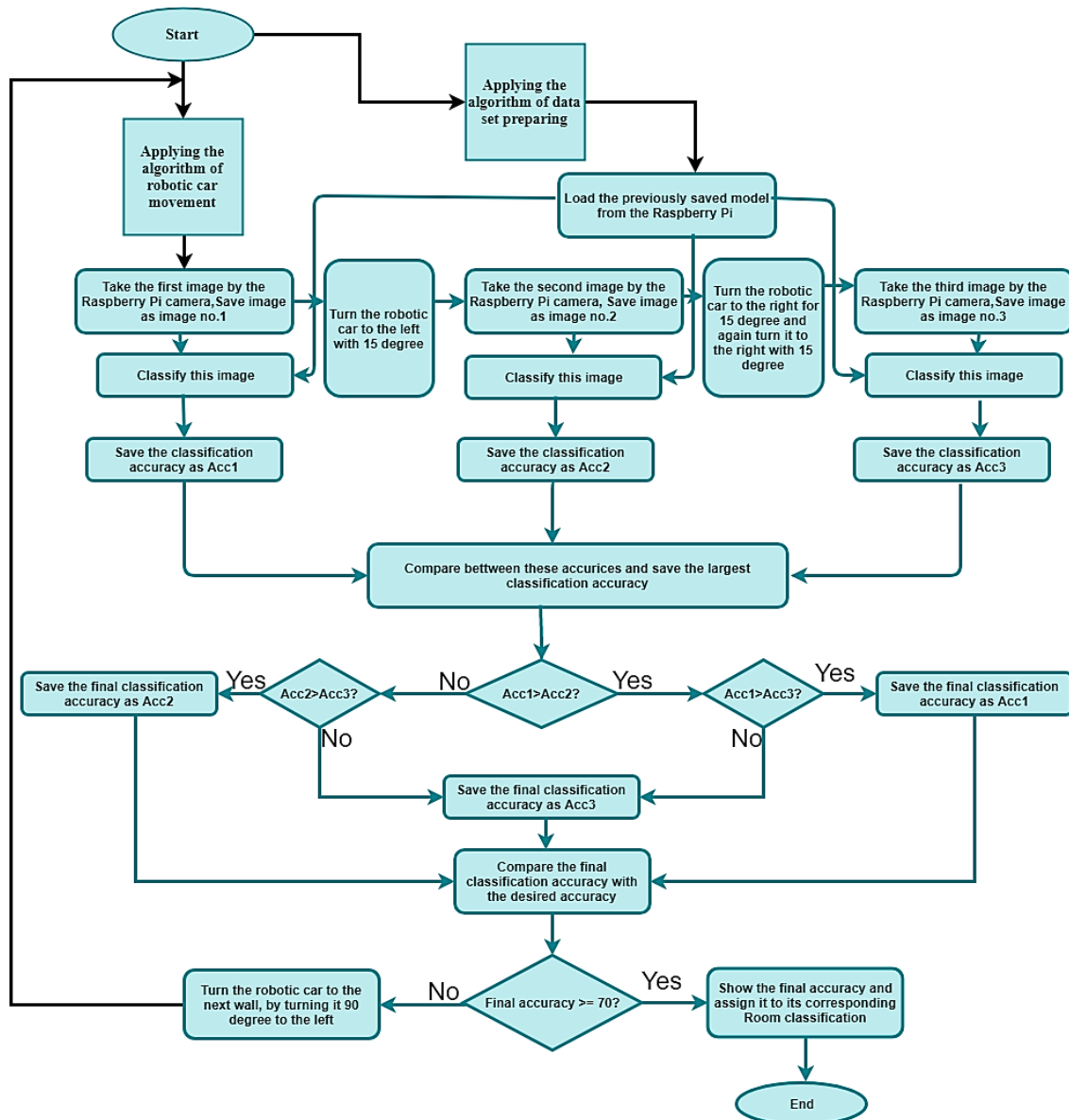


Figure 8. Capturing and classifying images algorithm

4. THE EXPERIMENTAL RESULTS

To determine the efficiency of the proposed system, many experiments are introduced on different case studies. As previously mentioned, the system contains a website, in which, the user interfaces with the proposed algorithms. In this section, different case studies of different situations and different locations are discussed in details. The first step in all these case studies is to login to the system through the login page, for authority confirming, so this step is introduced first followed by the case studies.

4.1. Authorization checking

The user should be authorized to be able to enter to the page of localization in the website. Otherwise, the user cannot do anything with website except seeing the information that are shown in the home page. Therefore, and as a beginning in the localization procedure, any experiment or case study in the proposed system passes through the following lines:

- In the starting of the website, a login button is appeared. The user should click on it to view the login form. It has been noticed that there is no sign in button. For security and privacy purposes, no one can sign in to the system without authorization.

- After filling the user name and password with an authorized account, and then clicking on the login button. If the account is authorized, the user should see the localization page, else the website will return to the home page with login error message.
- In this step, the user must select one person from the introduced list, in the proposed system only one person is active. But in the future, other persons will be active also. It should be noticed that only one person should be selected at any time.
- After choosing the name of the person to search for, the user should click on the search location button. Then the procedure of controlling the robotic car by the Raspberry Pi, checking for obstacles, and capturing the images that passes through different algorithms too. The evaluation of them will be introduced in the following sections with different case studies.
- The final results are shown as a text in the website.
- If the user clicked on the show location button, the last three taken images are shown in the gallery webpage.

4.2. Case study one

The first experiment, which can be named as the normal case study with distance less than or equal to 100 cm is introduced. Here, after the user selects the required person to search for, the following steps are followed:

- a. The Raspberry Pi orders the robotic car to move forward. The two blue LEDs are on, indicate the forward moving, as shown in Figure 9 (a).
- b. The car moves forward and checks the distance, when it became less than or equal to 100 cm and more than 80 cm, as the algorithm says, the Raspberry Pi orders the car to stop and wait for 5 sec. Then, again the distance is measured, if it changes, the car continues to move forward and wait the distance to become less than or equal to 100cm as shown in Figure 9 (b). Otherwise and as shown in Figure 10, the Raspberry Pi starts capturing three test images in three different angles, with 15 degrees between each corner. Then the car returns to its original location.
- c. These images are sent to the website to be used inside the program of classification. The classification procedure takes about 3-5 sec. The results are shown in the web page as a text according to Figure 11.



Figure 9. (a) The car is moving forward, (b) The nearest wall is on 90 cm distance



Figure 10. The procedure of capturing images

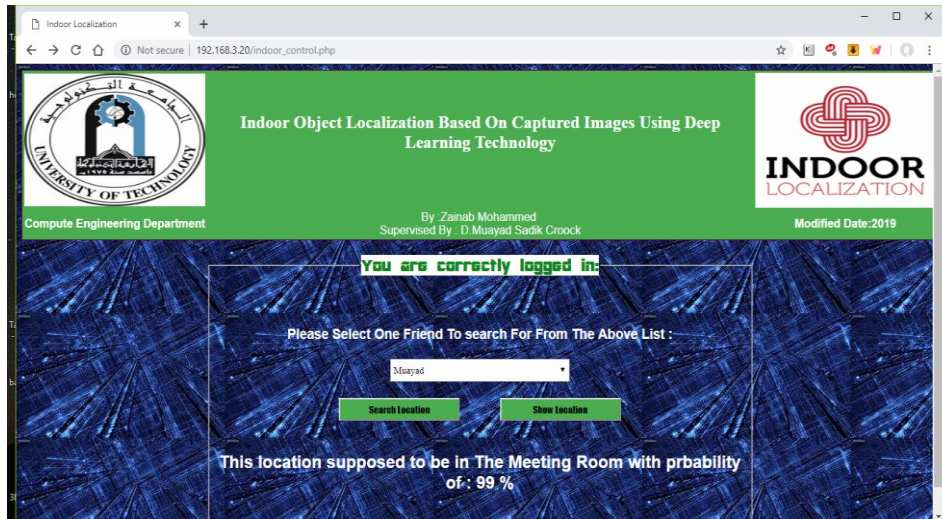


Figure 11. The final localization results of the first case study

4.3. Case study two

The second experiment, which can be named as the closest distance case study with distance less than 80 cm, is introduced. After the user selects the required person to search for, the following steps are followed:

- The Raspberry Pi orders the car (prototype) to move backward for one second, to increase the distance a little bit for not being too close to the wall. The two rear Red LEDs are on, indicating for start moving backward.
- The Raspberry Pi orders the car to stop and start capturing three test images in three different angles.
- These images are sent to the website to be used inside the program of classification. The classification procedure takes 3-5 sec of time. The results are shown in the web as a text.

4.4. Case study three

The third experiment can be named as the forward and backward obstacles detection. First, the forward obstacles will be discussed. As mentioned in the first experiment, (For forward obstacles detection) if the distance is less than or equal to 100 cm, the car waits for 5 sec, then the following steps are occurred:

- Check the distance, if it is increased, which means that the obstacle is moved, the car continues until it reaches again to a distance less than or equal to 100.
- Otherwise, if the distance is still less than 100cm, which means that the obstacles are not moving, the car stops and completes the capturing and classifications procedures. This is because the obstacle maybe the wall itself.
- The same procedure is done for the backward moving. When the car is moving backward and the distance is less than 30 cm, it will wait for 5 secs, then the following steps are occurred:
 - The RPi checks the distance, if it is increased, which means that the obstacle is moved, the car continues until it finishes the desired backward moving time.
 - Else, the distance is still less than or equal 30cm, which means that the obstacles are not moving, the car stops and completes the capturing and classifications procedures.

4.5. Case study four

This experiment can be called the wrong classification results handling experiment. This case study is the same as the normal experiment of the case study one, except that the probability of the classification is under the desired probability, which is 70%, as mentioned in chapter three. In addition, this case study considers an image distortion, that caused by unmoved obstacles, which indeed gives a wrong classification probability. In this case, the Raspberry Pi orders the car to turn to the left with a 90 degree as shown in Figure 12 (a). This makes the car moves to the next wall and then the total procedure of case study one is repeated, moving, checking for obstacles, as mentioned early. Then the car is capturing three images as shown in Figure 12, and classifying them until the probability is improved. The results of this experiment have shown that the classified images are taken in the Meeting room with probability of 99%. It has been noticed that only the final results are displayed on the website as shown in Figure 12 (b).

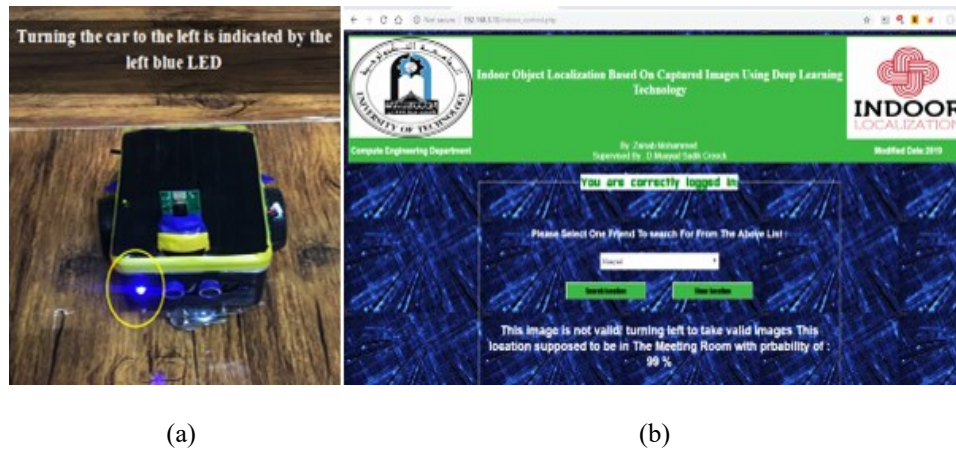


Figure 12. (a) Moving the car to the next wall, (b) The final classification results of the case study four

4.6. Case study five

This case takes the probability of putting new objects, which were not trained in the trained dataset. This experiment shows that the designed system can classify the walls to its corresponding room even if new objects are added like new chair, table, or new curtains. The same procedure of section B is executed here, with the addition of some obstacles as mentioned in section D. Moreover, a new curtain is added to the wall as shown in Figure 13 (a). This experiment is introduced in the admin room, and the classification results have shown that the probability of the images to be in the admin room was 82%. This proves that the proposed system has a great ability to classify any building as shown in Figure 13 (b).

The results of implementing the proposed system for different room classifications are listed in Table 4. It has been shown that many predictions are right. Many experiments had been introduced for this project, covering the eight walls, each experiment gave a different probability result. The results have shown that from every ten tested locations only one location is false. This means that the project false predictions are less than or equal to 10% which is acceptable loss percentage for a system starts with dataset from scratch.

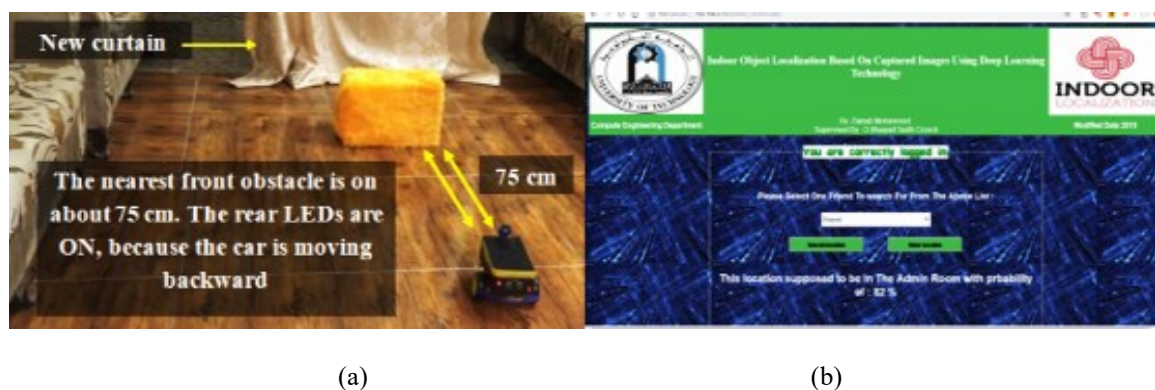


Figure 13. (a) Adding a new object, (b) The final classification results

Table 4. The final probability results for different experiments

Experiments	Origin location	Estimated location	Probability
1	Admin Room	Admin Room	99%
2	Admin Room	Admin Room	82%
3	Eating Room	Eating Room	88%
4	Eating Room	Eating Room	91%
5	Meeting Room	Meeting Room	99%
6	Meeting Room	Meeting Room	86%

5. CONCLUSION

An affordable indoor localization system based on deep learning was introduced and implemented to classify locations inside a building. The proposed system consisted of two parts: the classification algorithms and website building part (software part as well as the mini-robotic car (object) part (hardware part). The proposed algorithms were designed in deep learning technology with Keras library and CNN. A training model was created and trained, then the model was saved inside the Raspberry Pi SD card as a python script. The website was designed with HTML and CSS with the help of PHP for inserting the python scripts easily. Different experiments were introduced in order to test the proposed system. The results of these experiments have shown that it is easier to control the car by using only the Raspberry Pi control signals with high performance in term of accuracy and precision.

REFERENCES

- [1] Maryland NRCS, "Introduction to Global Positioning Systems (GPS)," USDA, August 2007.
- [2] Xuyu Wang, Lingjun Gao and Shiwen Mao, "BiLoc: Bi-Modal Deep Learning for Indoor Localization with Commodity 5GHz WiFi," *IEEE Access Special Section on Cooperative and Intelligent Sensing*, vol. 5, pp. 4209–4220, 2017.
- [3] Baoding Zhou, Qingquan Li, Qingzhou Mao, and Wei Tu, "A Robust Crowd Sourcing-Based Indoor Localization System," *Sensors*, vol. 17, no. 4, pp. 1-16, 2017
- [4] Mai Ibrahim, Marwan Torki and Mustafa ElNainay, "CNN based Indoor Localization using RSS Time-Series," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1044-1049, 2018.
- [5] Wei Zhang, Rahul Sengupta, John Fodero and Xiaolin Li, "Deep Positioning: Intelligent Fusion of Pervasive Magnetic Field and WiFi Fingerprinting for Smartphone Indoor Localization via Deep Learning," *16th IEEE International Conference on Machine Learning and Applications*, pp. 1-5, 2017.
- [6] Baoding Zhou, Jun Yang and Qingquan Li, "Smartphone-Based Activity Recognition for Indoor Localization Using a Convolutional Neural Network," *Sensors*, vol. 19, no. 3, pp. 1-15, 2019.
- [7] Wei Guo, Ran Wu, Yanhua Chen and Xinyan Zhu, "Deep Learning Scene Recognition Method Based on Localization Enhancement," *Sensors*, Vol. 18, no. 10, pp. 1-20, 2018.
- [8] Tang P., Wang H., Kwong S., "G-MS2F: GoogLeNet Based Multi-Stage Feature Fusion of Deep CNN for Scene Recognition," *Neurocomputing*, vol 225, pp. 188–197, 2017.
- [9] Moustafa Abbas, Moustafa Elhamshary, Hamada Rizk, Moustafa Youssef and Marwan Torki, "WiDeep: WiFi-Based Accurate and Robust Indoor Localization System using Deep Learning," *2019 IEEE International Conference on Pervasive Computing and Communications (Percom)*, pp. 1-10, 2019.
- [10] B. Javidi, "Image Recognition and Classification: Algorithms, Systems, and Applications," *CRC press*, 2002.
- [11] Yuan Y., Mou L., Lu X., "Scene Recognition by Manifold Regularized Deep Learning Architecture," *IEEE Transaction on Neural Network and Learning Systems*, vol. 26, no. 10, pp. 2222–2233, 2015.
- [12] J. Xiong, K. Sundaresan, and K. Jamieson, "ToneTrack: Leveraging Frequency-Agile Radios for Time-Based Indoor Wireless Localization," *21st Annual International Conference on Mobile Computing and Networking*, pp. 1-13, 2015.
- [13] J. Haverinen and A. Kemppainen, "Global Indoor Self-Localization Based on the Ambient Magnetic Field," *Robotic Autonomous System*, vol. 57, no. 10, pp. 1028-1035, 2009.
- [14] The PiHut, "Raspberry pi Camera," 2019. [Online]. Available: [https:// www.thepihut.com /products/raspberry-pi-camera-module](https://www.thepihut.com/products/raspberry-pi-camera-module). Accessed: 19 May 2019.
- [15] Nguyen L. D., Lin D., Lin Z., Cao J., "Deep CNNs for Microscopic Image Classification by Exploiting Transfer Learning and Feature Concatenation," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2018.
- [16] Zhou B., Lapedriza A., Xiao J., Torralba A., Oliva A., "Learning Deep Features for Scene Recognition Using Places Database," *27th International Conference on Neural Information Processing Systems*, pp. 487–495, 2014.
- [17] François Chollet, "Deep Learning with Python," *Manning*, 2018.
- [18] Jordi Mas and Carles Matue, "Introduction to Web Application Development," *Eureca Media*, 2015.
- [19] Martin Thoma, "Analysis and Optimization of Convolutional Neural Network Architectures", arXiv:1707.09725, 2017.
- [20] Tan and Clinton, "Embedded system application development for the raspberry pi", Nanyang Technological University, 2016.
- [21] Yunus Çelik, Mahmut Altun, and Mahit Güneş, "Color Based Moving Object Tracking with an Active Camera Using Motion Information," *IEEE Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1-4, 2017.
- [22] Liuliu Yin, Fang Wang, Sen Han, Yuchen Li, Hao Sun, Qingjie Lu, Cheng Yang, and Quanzhao Wang, "Application of Drive Circuit Based on L298N in Direct Current Motor Speed Control System," *Proceedings Volume 10153, Advanced Laser Manufacturing Technology*, 2016. doi: <https://doi.org/10.1117/12.2246555>.
- [23] Mohit Rane and Kalpan Mehta, "Potential Applications of Ultrasonic Sensor," *International Journal of Engineering and Computer Science*, vol. 6, no. 3, pp. 20583-20586, 2017.
- [24] Jang, Youngcheol, and Eunok Kwak, "Lithium Polymer Battery" U.S. Patent No. 9,065,083, 2015.
- [25] Muayad Sadik Croock, Salih Al-Qaraawi and Rawan Ali Taban, " Gaze Direction based Mobile Application for Quadriplegia Wheelchair Control System," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 5, pp. 415-426, 2018.