FOLLOWER ROBOT IN AN INTEGRATED AMBIENT INTELLIGENCE SYSTEM

A Thesis

by

ASHA NAIR

BS, Pune University, 2011

Submitted in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

in

COMPUTER SCIENCE

Texas A&M University-Corpus Christi
Corpus Christi, Texas

December 2019

FOLLOWER ROBOT IN AN INTEGRATED AMBIENT INTELLIGENCE SYSTEM

A Thesis

by

ASHA NAIR

This thesis meets the standards for scope and quality of
Texas A&M University-Corpus Christi and is hereby approved.

Dr. Scott King,PhD
Chair

Dr Luis Garcia, PhD
Committee Member

Dr Dulal Kar, PhD
Committee Member

December 2019

ABSTRACT

The purpose of this project is to create a digital supervision system by using a combination of motion detection, face recognition with sound-based interaction, people counter, and follower robot. It can be customized for homes and offices. The system can be used as a means of access control and estimating the number of people in the room, thereby proving very useful in real-life situations to manage security. The system responds spontaneously and accurately, which is attributed to the use of simple method calls, which increases system efficiency. There are several sensors and hardware components used, including webcams, vision-based sensors, motion sensors. The robotic system proposed is implemented using an Arduino UNO board and has eight working sensors that are used for making the robot follow unknown people and return to its home position. The experiments have proven accuracy in face recognition, people counter and efficient tracking by the robot to follow a person by avoiding the obstacles.

DEDICATION

I would like to dedicate my thesis to my husband, Ketan Solanki, whose unyielding love, support, and encouragement have inspired me to pursue and finish the research.

ACKNOWLEDGEMENTS

Foremost, I would like to express my sincere gratitude to my advisor Dr. Scott King for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis and throughout my graduation.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Luis Garcia and Dr. Dulal Kar, for their encouragement, insightful comments, and suggestions. My sincere thanks also go to Dr. Mamta Yadav for promptly providing me all the resources required for the completion of this project. A special thanks to my fellow labmates Kuldeep, Ogwo, Dekwaan, Evan, Leana, and Alexis, who have been kind and supportive in various ways.

I would like to thank my husband Ketan Solanki, my best friend Meghashree Baheti, my brothers Venkat Rathan, Hemanth Sai, and Vijay Teja for all their help and sleepless nights to help me succeed in my research. Thank you for the profound love and support. I owe you all more than ever. Finally, I would like to thank my parents and my in-laws. Their perpetual love made this possible, holding everything in place.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES PAGE

LIST OF TABLES

TABLES PAGE

CHAPTER I: INTRODUCTION

1.1 Motivation

In today's world, where technology is the primary driver of the globe, we feel its presence in every segment of our lives. Environments such as digital homes, digital labs, and digital workspaces support people in their daily life. For example, the home security systems and workspace surveillance system. Home security systems consist of integrated devices that work together with a central controller for detecting intruders and burglars. Workspace surveillance systems use multiple cameras for monitoring employee activities or protecting office assets.

We can create such environments where a system is trained to recognize the users and their behavior using Ambient intelligence(AMi). [26]. The algorithms are implemented to use the training information to program the system to take action to meet the user's needs. From home security system to automatic car unlock systems, it aims to automate day to day human tasks by making use of intelligent machines. These systems are usually sensitive to the presence of people and respond as per the situation. These systems can be personalized, adaptive, and anticipatory. The ambient intelligence integrates different systems and uses environmental sensors to sense a personal presence. In parallel, ambient intelligence is used as the security systems in many places to detect any unknown activities or people entering secured areas. It helps us to stay safe and notify us of any such activities. At times people entering the building may find it challenging to figure out the building's layout. In such cases, robots can detect people and interact with them to help them to navigate. Systems that can detect people, count the number of people, and trace the location of people are useful for surveillance and security in large gatherings where human management is impossible to control large the number of people. Expanding the same thought, we have created a follower robot in an integrated ambient intelligence system that interacts with people and can understand the presence of human beings and also distinguish between them. The goal of our research is to implement a Follower Robot in an Integrated Ambient Intelligence System for the iCore Lab at Texas A&M University-Corpus Christi to create a smart environment. The system consists of four main components; a Face Recognition System, Motion Detection, Follower Robot, and People Counter. The Face Recognition System is in charge of capturing laboratory member's faces when motion is detected and greeting them while people counter system counts the number of people entering and leaving the lab. The follower robot follows any unknown person that enters the lab.

1.2 Contribution

There are two main contributions to this research.

- An integrated environment with a follower robot that can be used for creating smart labs, smart homes, and smart offices which can be used for the purpose of monitoring and access control. The components of the system can be used together or individually for creating various synergized systems.
- A simplified way of object detection and object tracking, along with the classification of the object which can be used for other applications.

1.3 Overall System

This system provides a new dimension by integrating various independent technologies to create a digitized lab experience that will take access control and intruder detection to the next level. The people counter algorithm is used to ascertain the number of people in the lab at a given time.

In figure 1.1 consider the rectangular frame as the lab. When a person enters the lab, the people counter algorithm uses the overhead camera to detect the motion and increase the current count of the number of people by one or more. In parallel, motion detection and facial recognition algorithm make use of another camera which is placed in the room. It works as an access control mechanism for people moving in and out of the lab. In order to capture an unknown person's (not registered with the lab) movement in the lab, there is a robot placed a few steps away from the entrance. A signal is sent to the robot when an unknown person is detected. The robot then traces the movement of the person. The person is required to hold the red and blue blocks while moving around the lab. Thus, creating a system that tracks the number of people in and out of the lab at any given time, providing access to the lab by facial recognition and also intruder detection and movement with the help of the robot.There are 3 components in the system. Face recognition component, people counter component and follower robot component.

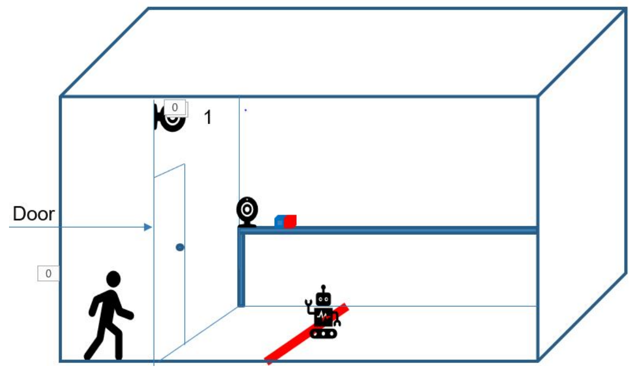

Figure 1.1: The overall system

There are three components in the system. face recognition, people counter and follower robot. All the components will be further explained in the following chapters. When a person enters the lab, the people counter algorithm is used for updating the count of people entering and exiting. At the same time, the motion is detected, and then the face recognition algorithm is processed. The follower robot has three modes, waiting mode, follower mode, go home mode.

1.4 Prior Work

Several contributions have been made in this area that makes use of ambient intelligence. Dat Do implemented the architecture of an integrated ambient intelligence system that was published in 2017, as shown in Fig 1.2 [17]. An intelligent ambient system was created using components such as body tracking, face recognition, controller, monitoring device, and interaction modules. Various devices such as RGB cameras and Kinect depth sensors were used to obtain human information for training the system. The controller used components like face recognition, body detection, and tracking to create location-based sound interactions with the user. The Monitoring system was implemented for logging and tracking of people in the environment. Such integration of various components helps in creating various interactive environments.

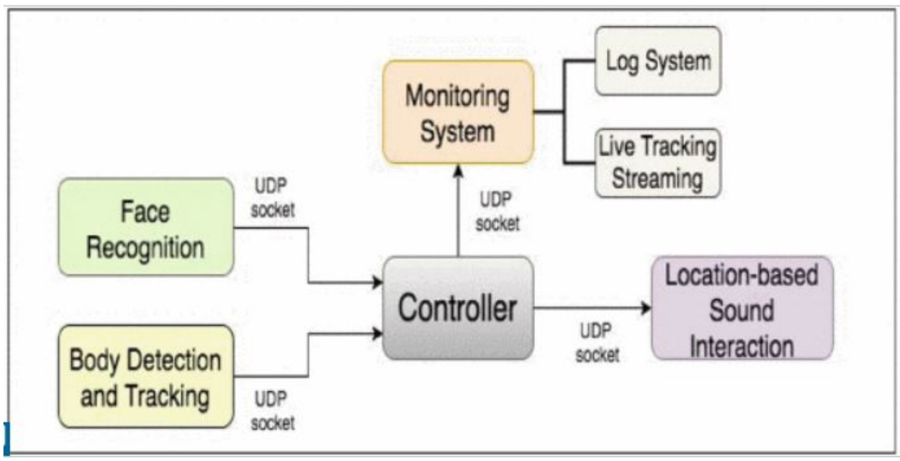

Figure 1.2: The architecture of the proposed IAMIS [17]

For creating our integrated system, we are using a combination of face recognition, people counter, and follower robot. Components used in our system are existing independent systems. Our system aims to use individual components and integrate these technologies to create a digitized environment. As the system contains three components, the prior work is discussed for each component in the below sections. The algorithm developed for each component aims to be computationally light with the highest accuracy in order to build an efficient and robust system.‘

Face Recognition

Face recognition can be done using two types of methods: Conventional methods and Deep learning methods.

Conventional Methods: There are several algorithms that follow the conventional and traditional methods for face recognition, such as fisher faces algorithm and Eigenfaces algorithm [34]. The advantage of using these methods is the simplicity of implementation. As they are primitive algorithms, they are highly sensitive to changes in brightness, expressions, aging, etc[6].

Deep Learning Methods: The process of deep learning refers to identifying the contours and facial points of a person's face, based on a defined architecture of 128 essential points and then saving this output to be used by machine learning for face recognition[35]. These 128 points are useful for solving the posing and projecting problems that occur in the face recognition process. As the next step to this process, it requires integration with Machine learning, which is essentially comparing the previous output of deep learning to a set of trained images and hence helping to achieve face recognition.[30] Face recognition is a step-by-step process and can be implemented using python and OpenFace[7]. Two approaches using deep learning which have been very accurate, are DeepFace and FaceNet.

1. DeepFace: Taigman et al., researched this algorithm.[42], it has an accuracy of 97.35% and a speed of 0.33 seconds per image using a single-core 2.2GHz CPU. The 3D modeling system is used for aligning the face input, and this input is then passed through a deep neural network. The deep neural network is used for the training of images[8].

2. FaceNet: Schroff et al.[38], researched this algorithm. It has an accuracy of 99.63%. In this approach, Euclidean distances are calculated to find the face distances between two images. Deep convolutional networks are used for generating 128 points and L2 distances. L2 distances are further used for comparing faces, and 128 points are used for solving the posing and projecting of faces. If the L2 distances are longer, then the faces do not match else the face belongs to the same person[20].

People Counter

There are several traditional methods used for counting the number of people entering and exiting a room. Some of the traditional approaches use mechanical counters, light beams, differential weighing systems, light-sensitive carpets, and vision-based systems [11] [18].

Figure 1.3 shows the methodology for the paper presented by Aditya Shukla in 2017 for pedestrian flow control using image processing [31]. Figure 1.3 shows the working of the system where the detection of the object is marked using a green box and the value of up-count and down-count, i.e., the number of people entering and leaving the lab, are updated in the video stream in the top right corner. In this scenario, the object is detected using contours from frames captured using an overhead camera. The position of the contours is estimated in the given frame to find direction. There are two lines that are drawn in the frame named 1-line and 2-line. If the object crosses the 1-line first and 2-line later, then the object has moved in the down direction. If the object crosses the 2-line first and then the 1-line, then the object has moved in the up direction. Every time the object crosses both the lines, the appropriate counter is updated. The accuracy of this particular method was 85%. The system has a limitation of not recognizing people. Any object moving in the frame is considered as a person, and the counter is updated.[14]

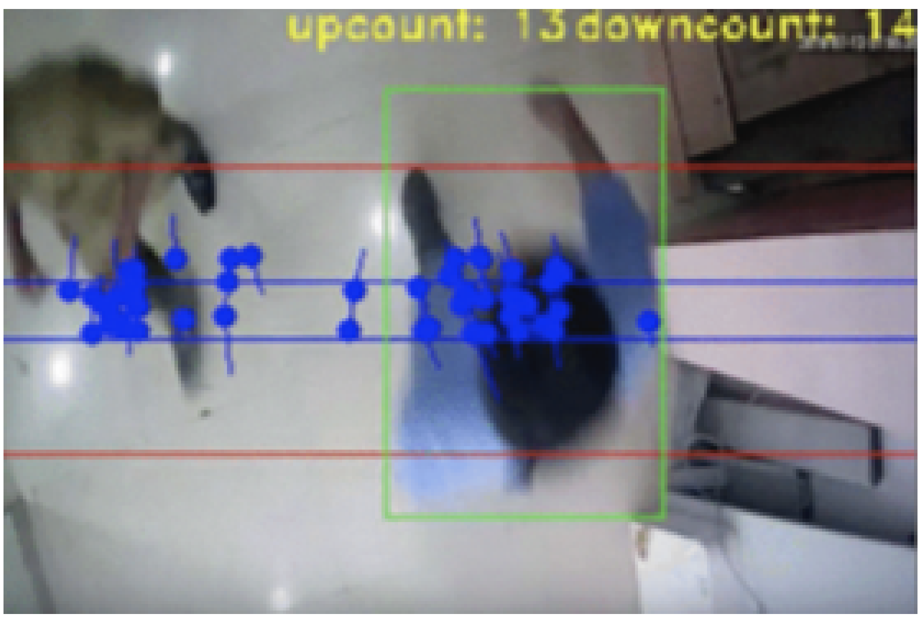

Figure 1.3: Pedestrian flow counter using image processing [31]

Another such work directional people counter based on the head tracking method by Jorge Garcia is shown in figure 1.4 below. This paper was published in 2017. In this paper, the object detection is used for detecting circular objects in the frame. The number of the circular object is then counted for determining the number of people in the frame. When any objects move, the Euclidean distances between all objects are calculated to identify the same objects in different frames.

The accuracy of the system is 91%. The number of people counted will be accurate if there is no other circular object in the frame. If any other kind of object enters the frame such as a ball or any animal, then it will be considered as a person. Hence the number of false detection will be significant.[21]

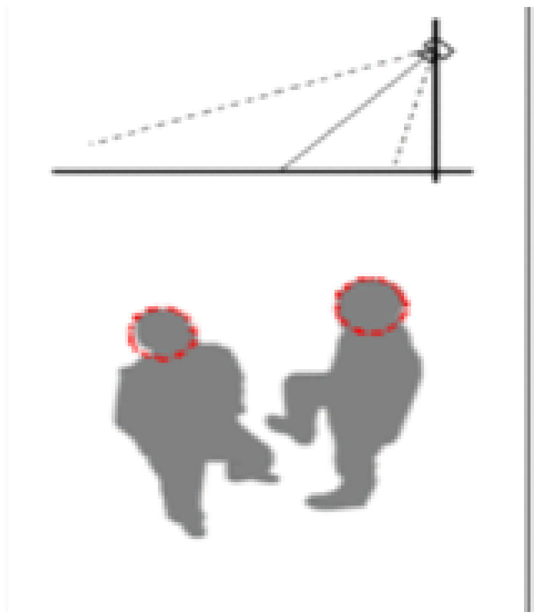

Figure 1.4: Directional People counter based on head tracking [21]

In our work, we have combined the concepts used in the above papers to come up with a more robust and accurate object tracking mechanism. The first paper had a line detection system that is used in our system. However, we are using the coordinates of a single line in order to determine the direction of the person versus the use of 2 lines in the previous work.

The second work defined a method for identifying missing objects in the frame, the objects in every frame are checked and if there is any object that is not close to objects in the previous frame, a new object is registered. We have made use of this concept into our system since it is useful for registering and de-registering objects as per updated frames in the video stream[12].

Follower Robot

Significant advancements have been made in the area of an object following by a mobile robot. A few of the methods by which robots can follow objects are explained as follows:

Object following method

In the object following method, the concept of image processing is used[23]. Attributes for the target can be the shape, color, length, or width or height of the object. The shape and the color of the object is generally used for detecting a specific pattern of the target for the robot to follow[48]

Leader following method

This algorithm uses a camera that is used for image processing[41]. The camera is mounted on either on a servo motor or pan-tilt or both. The servo motor is used for rotating the camera in the X-axis direction, and the pan-tilt is used for moving the camera in the Y-axis direction. The robot usually follows one particular object.[32] It generally does not communicate with the object and tries to find the object in the horizontal direction by moving the camera using servo motors or in the vertical direction by using the pan-tilt mechanism.[48]

Human following

The human following method usually uses image processing for identifying the individual by making use of the individual's color of the cloth and texture.[13]. The target should be within a specified range of the robot, and the robot follows the human around. If the robot loses the target, then it tries to find its target within a specified angle or range and tries to reconnect with the target in order to follow it again. [48]

Vision-based sensor method

The vision-based sensor method uses images captured by the camera to determine the presence, orientation, and dimensions of the object.[19]. The lights and the controller are all controlled by a single unit, which makes the operation simple and effective.[36] They provide added functionality like multi-point inspections and detection of the target in inconsistent positions.[47] Whenever the target is detected, the vision-based sensor can also return the x,y coordinates or dimensions related to the target. Our system adopts the vision-based sensor approach. [48]

In the system proposed by Satya Savithri in 2017 in the paper named person follower robotic system, the robot tends to read the letters printed on a paper, as shown in figure 1.5. If the robot

Figure 1.5: Person follower robotic system [37]

reads the letter HL, then it moves forward. If the robot reads the letter R, it moves towards the right, and if the robot reads the letter L, then it moves towards the left. This particular system is used for handicap people assuming that the motion of the person is at a certain speed. If the speed of the person moving increases, then the performance of the robot decreases, which is a limitation in the system. However, the robot is working on the Arduino board and is not using any other processor on top of it. Hence there is no latency in the system. We have made use of the same concept for implementing the algorithm. The implementation is done directly on the Arduino UNO board using C language, and the sensors are connected in a similar way for faster execution without any latency. As our system is not using the concept of reading letters for following the target, and the object follower uses vision-based sensors for following the target, the limitations in this system do not affect our system. [37]. Dhiman Sarker followed the vision-based approach for robot following in a 2017 study [39]. The concept of optimal flow is used for determining the deviation of the person as the motion is detected, see Figure 1.5. As per deviation, the angular displacement is converted to a certain angle.

This particular angle is used for rotating the robot in the direction of the person. As the entire system consists of a Raspberry PI, which is connected to the Arduino board, and the computationally expensive optimal flow is used for detecting the deviation, the system has some latency. Due to this system, sometimes due to this lag, the robot tends to miss the target at a time. From the above, we have taken the concept of rotating the robot as per deviation and used it in our system to rotate the robot as per the X-axis value of the target. [39]

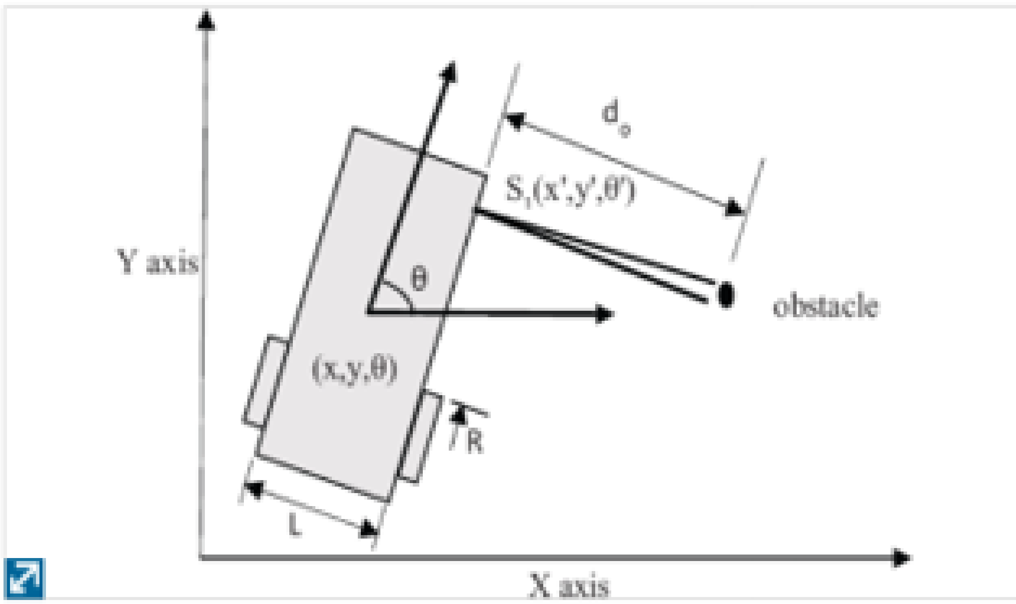

Figure 1.6: vision-based intelligent object follower robot [39]

In Vision based sensor cart follower developed by M.F Ahmad in 2017 for the wheelchair by using a microcontroller which made use of the vision-based sensor[1]. The sensor used for the color tracking technique was pixyCMUcam5 for a wheelchair luggage follower. The methodology consisted of three parts: object identification, finding cart direction, obstacle avoidance. For object identification, two color patterns were used to identify the target of interest. For tuning the direction, servo motors were used, and the servo motors were positioned in the middle of the linkage rod of the cart, which was connected to the front caster wheels.

For obstacle avoidance, the ultrasonic sensor was connected to the Arduino microcontroller. One of the ultrasonic sensors was mounted in the middle section of the cart, and the other two ultrasonic sensors were placed in the front section of the cart, but on the left and right side in order to avoid the obstacles coming from both sides.[2] When the method would detect any obstacle from either direction, the cart would either stop or turn in another direction, depending upon the position of the obstacle. After re-positioning itself, the cart would recalculate to find the target and try to follow by moving servo motors in specific directions. We have made use of the advanced version of pixyCMUcam5 into our system for the purpose of object identification.[1]

CHAPTER II: SYSTEM DESIGN

This chapter details the system design. Figure 2.1 below details the hardware components used within the system. Several sensors, such as vision-based sensors, servo motors, 2.4G wireless serial, line follower sensor, color sensor, infrared sensor, an ultrasonic sensor, are used to detect human activity. The webcams are used for motion detection and for capturing pictures that are used for facial recognition. If the person is not recognized, then infrared signals are sent to the robot. The webcams are also used for updating the count of people who enter and leave the room. This way, webcams are used for capturing human activity, and the signals are sent to the robot and controller for further processing.

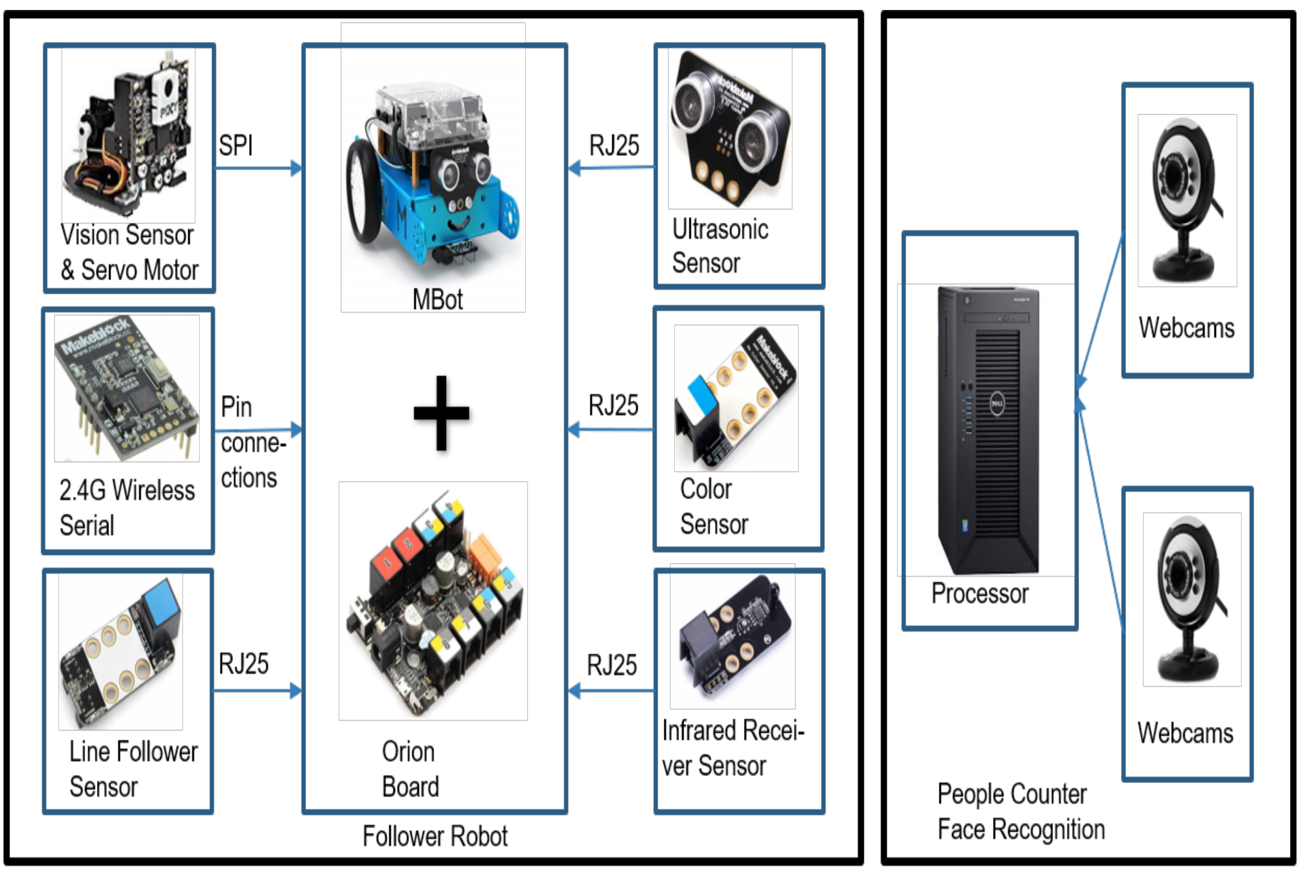

Figure 2.1: Hardware used in the system

Figure 2.2 describes the integrated software components used within our system. The working mechanism of this component is such when signals are received from the hardware component, the presence of a person is suggested, and several modules are triggered. The face recognition and people counter modules are contained within the same logical block. The component view of the system is as follows:

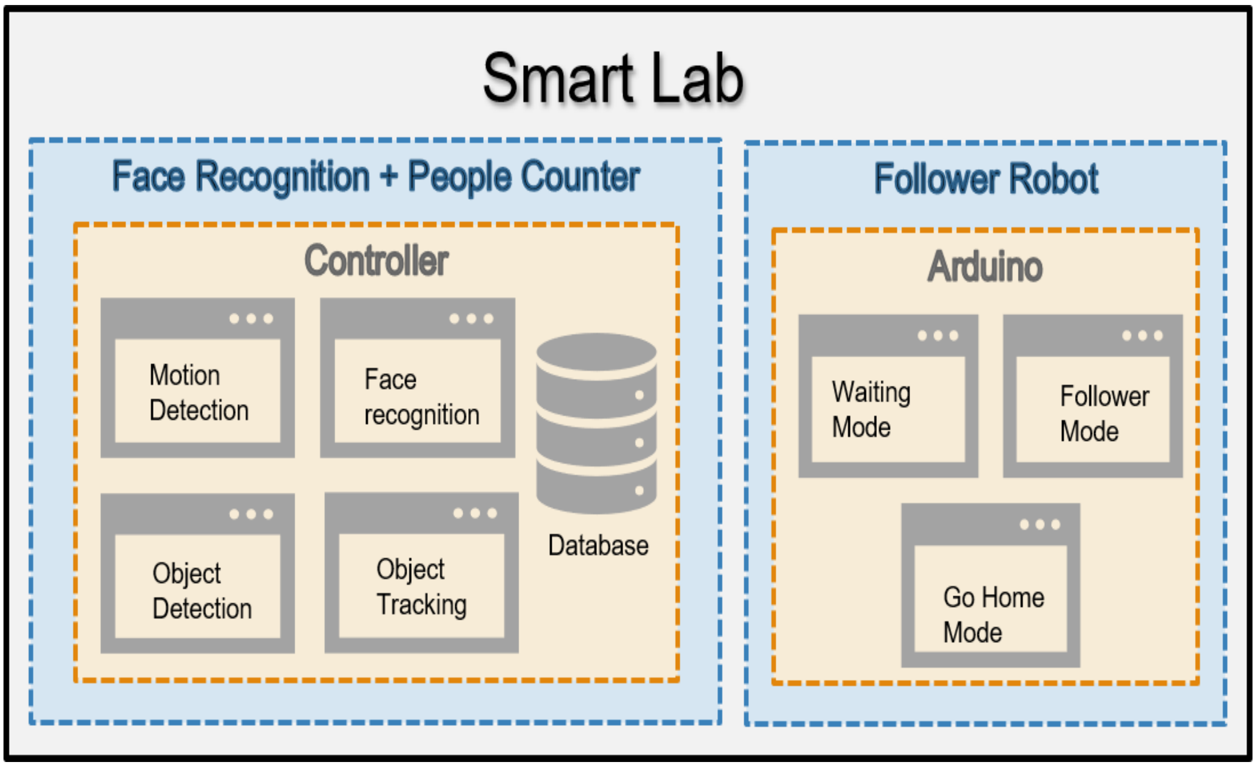

Figure 2.2: Software used in the system

The face recognition module makes use of motion detection and face recognition components, whereas the people counter module makes use of object detection and object tracking module while the database contains the trained data-set. The follower robot makes use of the Arduino block, which is triggered using the face recognition module if it detects any unknown person. The Arduino block also contains the follower mode block and go home mode block.

The facial recognition module receives the frames from the webcam and detects motion using Euclidean distances between alternate frames[25]. If a motion is detected, then the face recognition algorithm is triggered. It then greets the individual by its name upon finding a match. If the face is not recognized, a signal is sent to the robot for triggering follower mode. Also, upon receiving such a signal, the people counter is incremented.

The follower robot component gets triggered by signals sent from the controller when the face recognition module detects an unknown person, which then makes the robot follow the individual based on certain predefined identification blocks. The follower robot works intelligently and goes back to its home position when the person leaves the lab by making use of go home module. There are three major components in the system:

1. Face Recognition
2. People Counter
3. Follower Robot

2.1 Face Recognition

Face recognition identifies the person who enters the lab. It is required for sending a signal to the robot in case an unknown person enters the lab. Webcams are used for Face recognition. The algorithm processes a continuous stream of video and captures the frame of the video whenever motion is detected. If there is a face detected in the frame, then the face recognition procedure is triggered. If a match is found, the person is greeted with ”hello”, and their name. The unknown person is greeted with ”hello” and a signal is sent to the robot for triggering the follower mode. The face recognition is used in conjunction with the trained data-set.

Data-set

A data-set is primarily a set of training images and registered data images that are used for face recognition. This data-set is used in the face recognition process. The detected face is matched to the trained faces in the data-set to obtain the result. If the data-set is large enough with several images of the same person from multiple angles, the algorithm learns and becomes more accurate in terms of face recognition.

Figure 2.3 shows a sample data-set of one individual. We have used five data-sets in this system of different individuals like former USA president Barrack Obama, current USA president Donald Trump and Bollywood actor Salman Khan and myself. Each data-set contains roughly around 200-300 images.

Figure 2.3: Example data-set for face recognition

These images are used for creating encoded values of faces. These encoded values will then be used in the process of face matching to predict the identity of the person. Encoded values have a corresponding name value for every entry, which is further used for greeting the person.

Implementation of the face recognition module

Figure 2.4 shows the steps involved in the implementation of the face recognition module. The first step in face recognition is preprocessing, which involves the creation of encodings. It is followed by motion detection. If the motion is detected, the frames are captured for face detection. Once the face is detected, the posing and projecting of the faces are resolved by editing the images. The encodings of the captured frame are created, which is finally used for finding the name of the person.

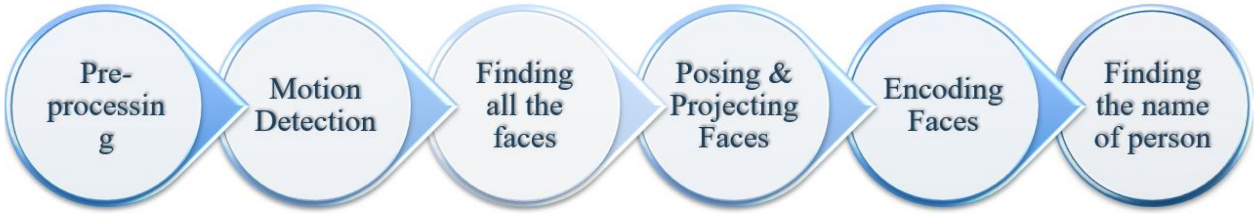

Figure 2.4: Steps involved in face recognition

Pre-processing

For pre-processing, the images from the training data set are loaded from the database, and encodings of the faces of those images are done. Two arrays are created, one for face encodings and the other of face names. Each entry in the face encoding array will have a corresponding entry of a name in the face-name array. These images are then converted into gray-scale images, and histogram equalization is done. The purpose of converting the images into gray-scale images is for reducing the processing time and increasing the accuracy of face recognition in the detection of the face.

Motion detection

For motion detection, the Pythagorean distance is calculated for every alternate frame for comparison. For example, frame one and frame three are compared, and their distance is calculated. An absolute threshold is predetermined. If the standard deviation is greater than the standard threshold (default value 10), the motion detection flag is changed to true, and the frame is captured for finding all the faces. [9]

Finding all the faces

Histogram of oriented gradients(HOG) is used for face detection. As discussed in Chapter 1, the results of the HOG were most accurate and efficient in the system developed by Dat Do. Hence HOG was used in our system for face recognition. The purpose of HOG is to detect objects using the feature descriptor of computer vision.[43] The image is converted to a gray-scale. For every pixel in the image current pixel is compared with the adjacent pixel to check which is darker. Arrows are then used for indicating which pixel is darker, and this process is then repeated for every single pixel. This helps in identifying faces in different under different brightness conditions. After drawing all the arrows, blocks of six 3×3 frames are created in the image to determine the basic pattern. A single arrow is added for the entire block. This basic pattern helps in determining eyes and the mouth area of the face, as shown in figure 2.5. There are several pre-trained faces provided by OpenCV, which can be used for determining whether the deduced basic pattern is that of a human face or not [9]. If a face is detected, the image is further processed to solve the posing and projecting of faces.

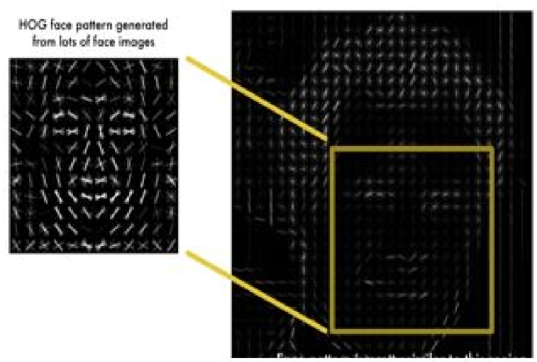

Figure 2.5: HOG [9]

Posing and projecting of faces

After face detection, for different posing and projecting of faces, 68 specific points of the face are identified. It is used for figuring out the eyes and mouth area of the image, as shown in figure 2.6. The image to be tested can be rotated, scaled, or sheared to get 68 points to match. Once the 68 points are identified, the encoding of the face is generated. [9]

Figure 2.6: 68 points [9]

Encoding faces:

After identifying the 68 points, the encoding of the face is generated for matching the face. These encodings are also generated for all the trained images in the data-set. It is used for matching the test face with the trained face. For encoding faces, a deep convolutional neural network is trained to generate 128 measurements for each face image. Three images are loaded for generating these 128 points. The first being that of the person whose face we want to train. The second image is of the same person whose image we want to train the system. Whereas the third image belongs to someone different. 128 points measurements are then calculated for each of the three images. The first and third image measurements will vary since they belong to two different people, Whereas the measurement of the first and second images will be similar since it belongs to the same person. By repeating this process, millions and millions of times, the image of the neural network learn to generate 128 points for each person accurately.

OpenFace has already implemented this method, and these trained networks are published for generating 128 points. For generating encoding, we have to run our face image through their network to generate 128 points. For preprocessing, the images in the data-set are also run through the same network for generating 128 points. These 128 points will be later used for matching.

Figure 2.7 shows an example of the encodings generated by the OpenFace trained network. These encodings are further passed through a machine-learning algorithm to find the name of the person.

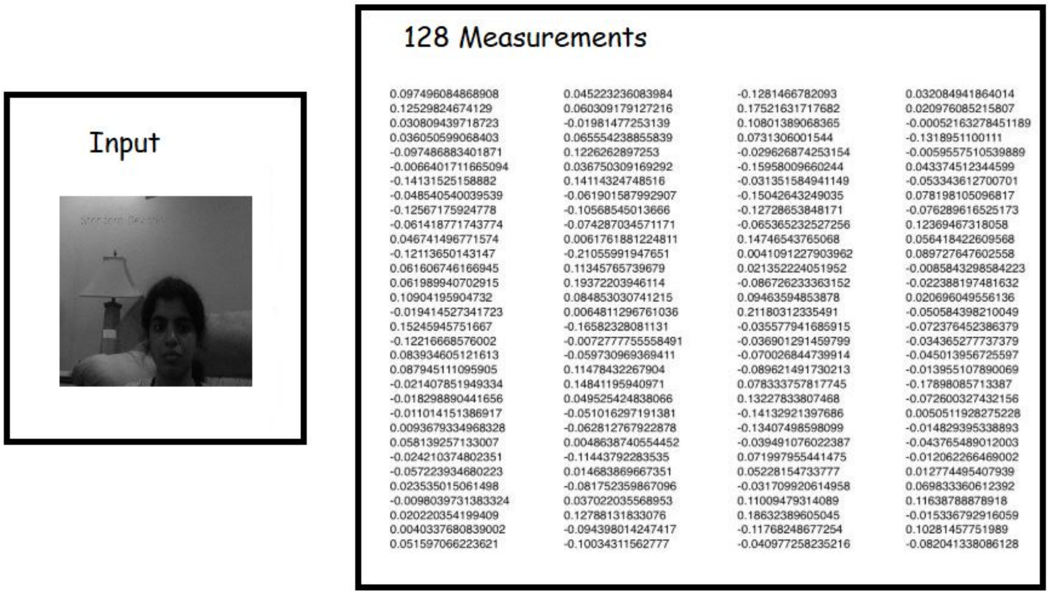

Figure 2.7: Example of encodings of a trained faces

Finding the name of the person:

After generating encoding, the value of the encodings is used for finding a match for the test image in the trained data-set. For finding the name of a person, a trained classifier can be used for taking in the measurement of the new test image and telling which person is the closest match in the trained data-set[45]. The result of the classifier will be a value from 0 to 1. This value is then compared to a predetermined threshold. If the value is higher than the threshold, then a match is found else not. In our system, we have made use of the support vector machine (SVM) classifier[10]. The SVM is a learning algorithm that classifies data for analysis. Basically, it assigns new examples to one category or another[40]. A prediction is made if it belongs to the category. The prediction returns a value between 0 and 1[15].

If the value is higher than the threshold, then a match is found. The face encoding index value is returned. This index can be used to find the face name from the face name array. The name of the person is then concatenated with ”hello” and using text to speech feature of the python the computer greets the person otherwise the unknown person is simply greeted with a ”hello” and a serial message is sent to the robot using wireless serial sensor in order to trigger the robot to follow the visitor.

As shown in figure 2.8, two frames are shown. The first frame shows the image when the face is recognized. A red rectangle is drawn around the detected face. It is labeled with the name associated with the detected face. The second frame shows the image that indicates deviation. If the value of the standard threshold(which is 6 in the image), goes above 10, motion is detected.

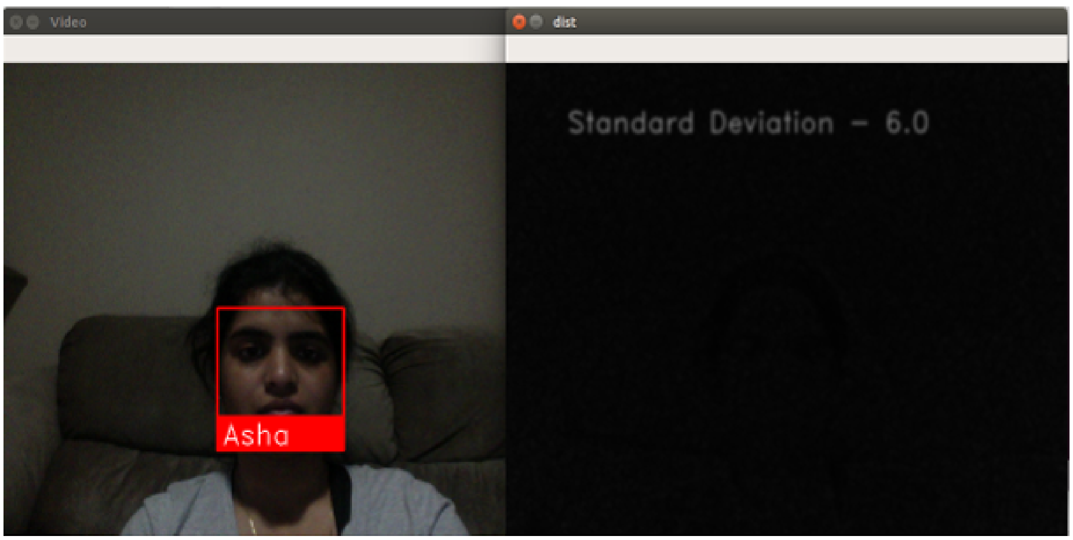

Figure 2.8: Face recognition screen

This algorithm works even when the person has used accessories like a cap or sunglasses. As shown in figure 2.9, the face is recognized even when the person is wearing sunglasses and a cap even though the trained data-set did not have images with the accessories.

Figure 2.9: Face recognition screen

2.2 People Counter

The people counter module has three steps, as shown in figure 2.10. For every Nth frame, the object is detected. This object is tracked until (N-1)th frame. Again, at the Nth frame, the count of people is updated and new objects are detected. This is done until the video stream is stopped. The value of the N varies as per the speed of the camera and the processor used.

Figure 2.10: Steps in people counter

Object Detection

SSD and MobileNet are used for object detection. SSD is used for detecting objects and MobileNet is used for getting the position of the object. In this phase, new objects are detected using frames obtained from the webcam. In this phase, we also validate if we missed any object that was identified in the object tracking phase. The object detection algorithm returns the bounding boxes of the object detected, along with the x and y coordinates of the object with reference to the frame [44]. Object detection algorithm for every alternate frame can become computationally expensive [33]. Hence in our system, we have used it after N frames(default value 25). The default value of the periodicity can be reduced if the machine used for processing is more powerful, and the speed of the camera is higher.[39]

In figure 2.11, two objects are identified. One object is classified as a person and marked using a blue rectangle and, another object is classified as a car and marked using a red rectangle. The rectangles are bounding box which show the area occupied by the object in the given frame. The values of each rectangle are denoted using x and y coordinates with reference to the entire image. These x and y coordinate values can be used for determining the area and the position of the object in the given frame. A confidence value is returned with the classification of the object. The confidence value can be used to prevent false detection.

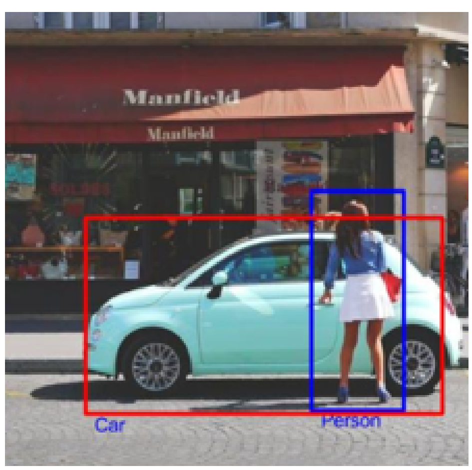

Figure 2.11: Object detection using SSD and MobileNet [39]

In figure 2.12, the first frame shows the screenshot from a video, where a chair is pushed from one end to another. The chair is not detected as an object, and the counter is not updated. The second frame shows the screenshot from a video, where a person on the chair and the chair is pushed again from one end to another. In this case, the person is recognized as an object, and the counter is updated.

Figure 2.12: Object detection using SSD and MobileNet [39]

Object Tracking

After object detection, object tracking is used for tracking the movement of the detected objects in the frame. Tracking is done till $(N-1)th$ frame. Thereafter on the Nth frame, the object detection method is executed. This process keeps running in a loop.

The input to the algorithm can be static or dynamic. The algorithm can run on the live video stream from a webcam or using a previous video. To use a recorded video file, the location of the video file can be provided as an input parameter for the method to execute. The output can then be saved as the video file by providing the name of the file, which can be passed as a parameter. The skip frame value can be changed by passing it as a parameter. In our system, we have defaulted the value to 25 based on analysis done, which are described in the following chapters.

We have made use of the OpenCV method for image processing, the *dlib* library is used for object tracking, and the deep learning method is used for object detection algorithm. The object detection algorithm is used for creating the bounding boxes.[22] These boxes are then used to define the centers for every object in the frame, as shown in figure 2.13. Then the Euclidean distances between new centers of the new objects in the current frame and old objects in the previous frame are calculated. [37].

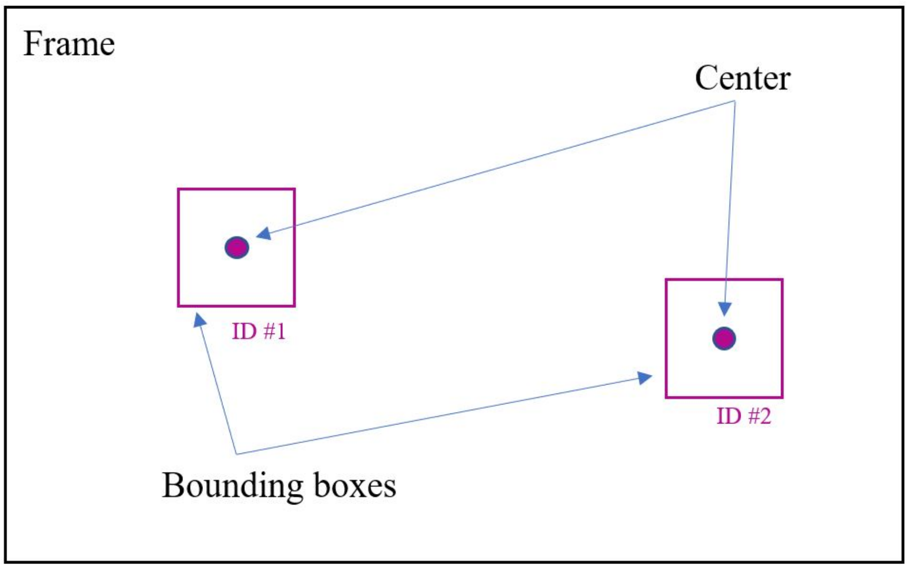

Figure 2.13: The centers of the objects in the frame

In a given frame, we can have one or more objects. For simplicity, suppose we have a previous frame named frame A, which has two objects and a current frame-$frame_B$ that has three objects. We now have to find the distance between the objects of the $frame_A$ and $frame_B$. While two objects of the $frame_A$ objects will find their corresponding distance with the objects in $frame_B$. A third object of $frame_B$ will remain unassociated. This third object will then be given a new object ID, and the other two objects will be given the same object id as that of $frame_A$ as shown in figure 2.14. If any object is lost for more than 40 frames, then it is deregistered or marked as lost.

The difference between the previous centers and new centers helps us identify the distance and the direction of the object that has entered the frame. As shown in figure 2.15, the distance and the direction between the centers can be easily predicted by using this method.

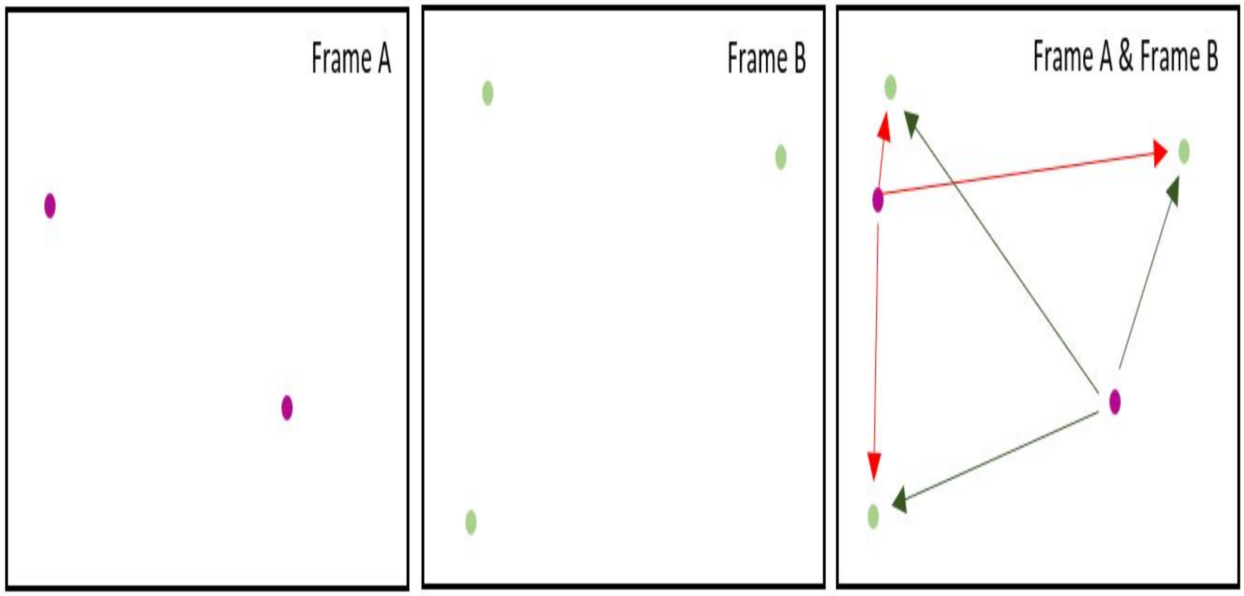

Figure 2.14: Finding Euclidean distance between old and new objects

The object can be classified into various classes using `Single Shot Detector`. Since the object

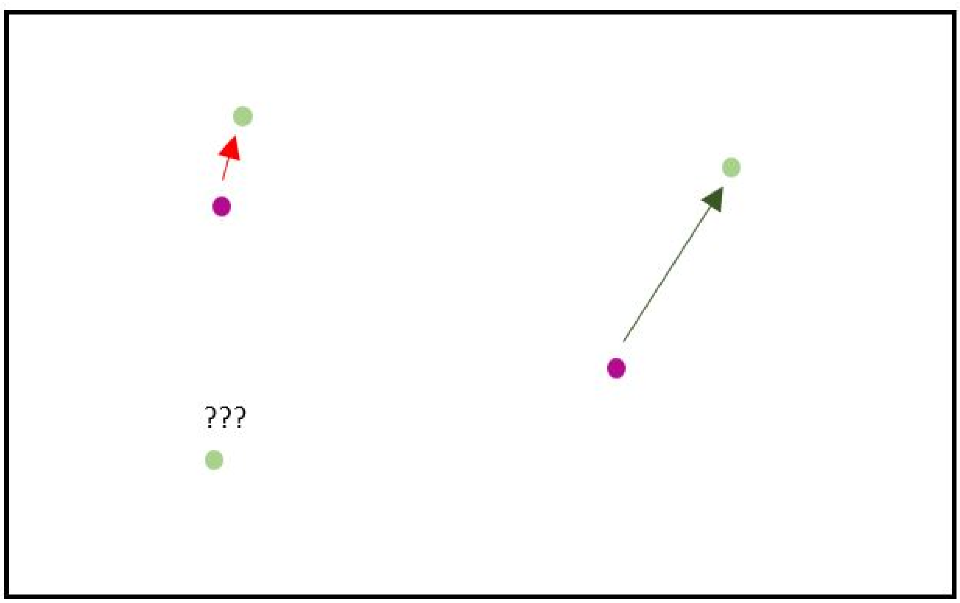

Figure 2.15: Registering a new object and finding the direction

that we are trying to identify in the frame is that of a person, here we use the person identifier from the Single shot detector model. Once the object is detected, its position of the center is saved, and the loop of calculating the distance and determining the direction keeps executing.

Find direction and Update counter

A horizontal line is drawn on the frames at Height/2 and Width/2 of the frame. When an object crosses any coordinate of the line, the counter of the up or down is incremented. The y coordinate value of the object is used for finding the direction. The y coordinate value for all previous frames of that particular object is used for calculating the mean value. If the value is negative, the object has moved up. If the value is positive, the object has moved down. The up and down values are displayed in the frame.

As shown in figure 2.16, two IDs are found in the image. The position of the center of each of the

Figure 2.16: People counter system

objects is compared to the yellow line in the frame, the down counter and up counter is updated depending on the direction of movement of the object. The values displayed in the frame are as follows:

1. The ID value shows the number of objects that have been identified.

2. The status has two values, detecting and tracking. When the object detection algorithm executes, the value displayed is detection. When the object tracking algorithm executes, the value displayed is tracking.

3. The yellow line is the visual horizontal division marking the center of the frame

4. The up and down values indicate the number of people who entered and left the lab

2.3 Follower Robot

The follower robot module follows an unknown person around the room, tracking their movements. The robot receives a wireless serial message from the controller to indicate the detection of an unknown person. A pre-condition to the follower robot is that the person needs to carry the blue and red blocks that serve as signals to the robot to help him follow the person. This block can be replaced by a tag, hat, or jacket in the future. The robot follows the colored block for user-defined minutes and again goes back to its home position after following. The robot can be forced to return to its home position by using a remote to change the mode.

The follower robot is made up of vision sensors, line follower sensors, color sensors, ultrasonic sensors, and infrared receiver sensor, as shown in figure 2.17. These sensors make use of two types of connections, i.e., RJ25 and SPI cable. The ports used are port 3, port4, port 6, port 7, and port 8. The port 1 is used as an input port, and port 5 is used as an output port by the Pixy CMUCam5 sensor. There are 2 DC motor-25 used, which is connected to the caster wheels. Figure 2.18 shows the RJ25 connection and SPI connection.

Waiting mode

By default mode the robot is in waiting mode. The robot returns to waiting mode after go home mode. Figure 2.19 below shows the working of the waiting mode of the follower robot component.

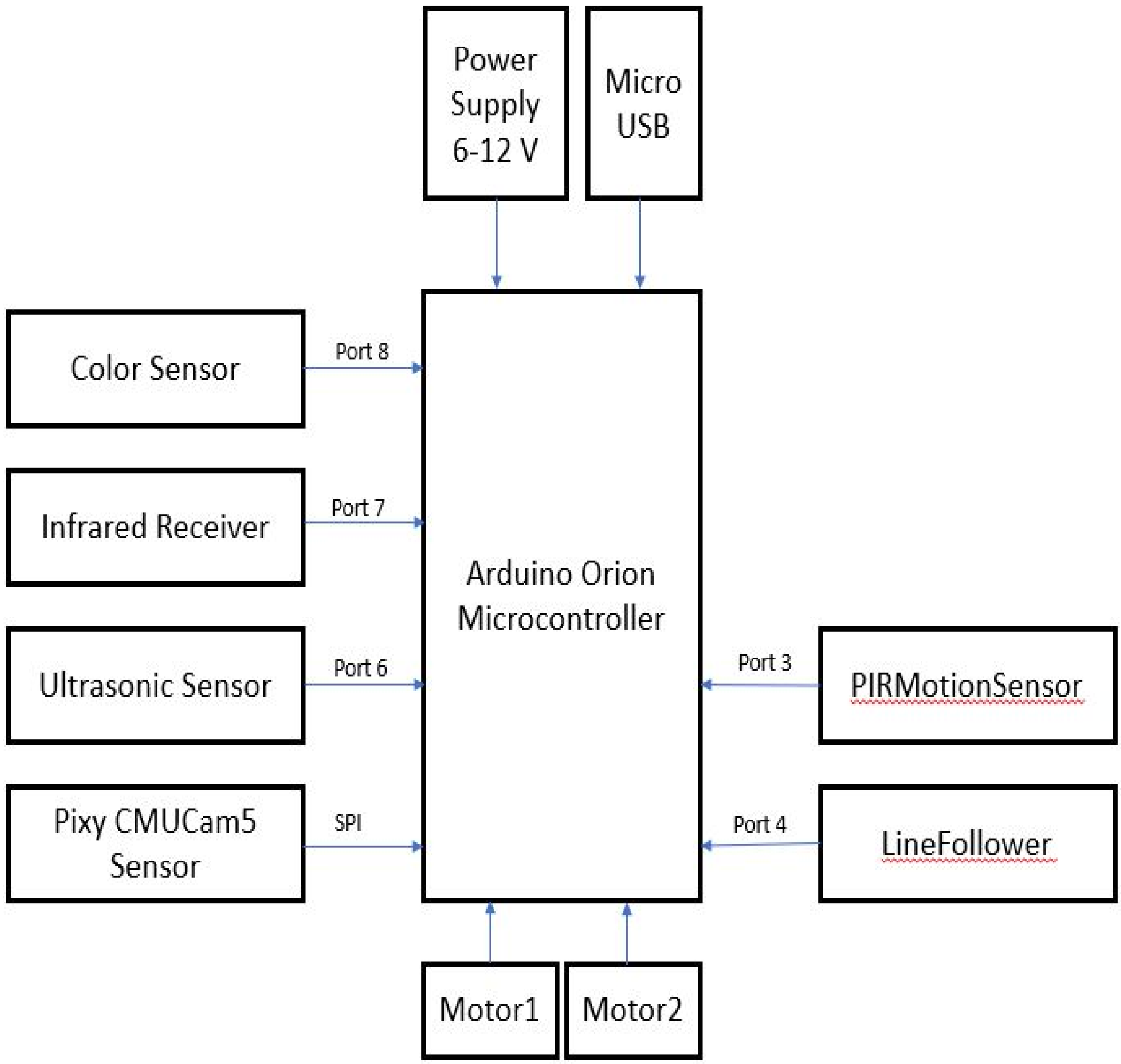

Figure 2.17: Component view of the system

When the robot is in waiting mode, it keeps waiting until it receives a signal for changing the mode to follower mode. This signal is sent when the face recognition component detects an unknown person.

When the face recognition component detects an unknown person, it sends a serial message to the Arduino board, which is connected to the computer using wireless sensors. This Arduino

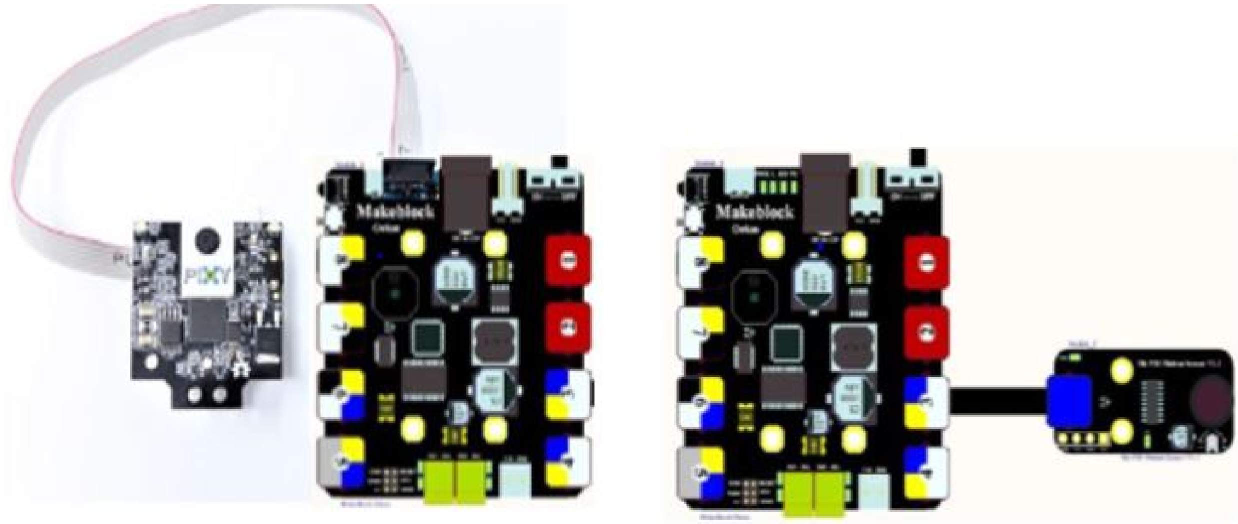

Figure 2.18: Types of connections

board then sends infrared signals to the robot, which changes its mode to follower mode.

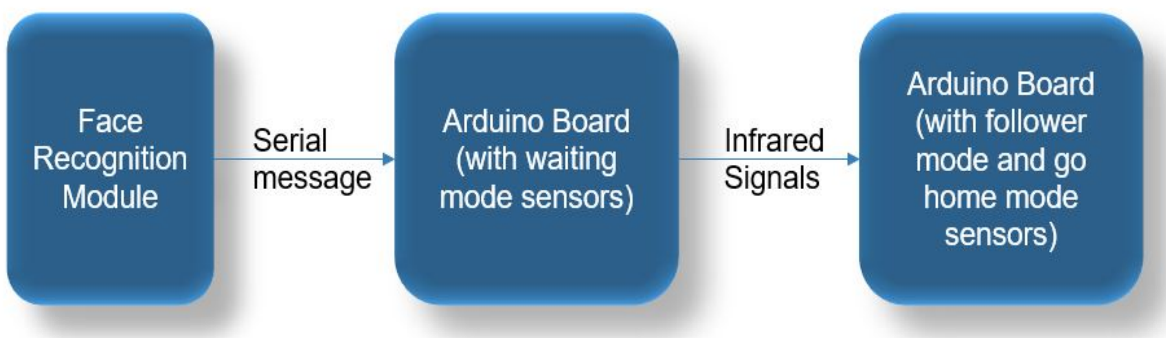

Figure 2.19: Waiting mode

The algorithm used for the implementation of motion detection is as shown in figure 2.20.

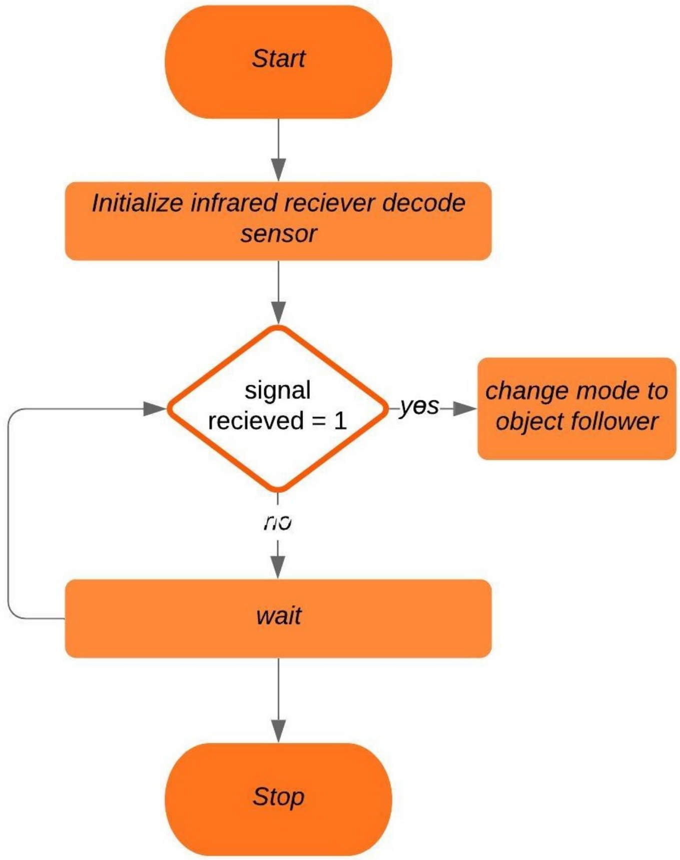

Figure 2.20: Waiting mode algorithm

The algorithm begins with the initialization of the infrared receiver decode sensor. If the infrared signal received is 1, then the mode is changed to object follower. If the signal is received is not 1, then the robot waits until its sensor receives 1 signal.

Follower Mode

Follower mode is triggered when an unknown person is detected by the system. For follower mode, we make use of the pixy CMU which acts as a vision sensor. The field of view of the Pixy CMUCam5 camera is approximately 60°. The Pixy CMUCam5 CMU camera has a PixyMon software which is used for training objects[29]. The trained objects are called signatures. Each signature is either a color or a color code (CC). For every Pixy CMUCam5 camera, approximately seven signatures and seven color codes can be saved. The limitation of the color signature is that it cannot be combined with another color to create a new signature. If a color code is used, then it can be used in the combined to form a new signature. For example, the color codes can be used for combining two colors and creating a new signature. For the signature CC 1 and signature CC 2 together, the Pixy CMUCam5 camera is going to detect the object as signature CC 12.

In figure 2.21, the S=12 is the signature value that is detected. In this sensor, the signature CC 1 is saved as blue, and the signature CC 2 is saved as red. When both the colors are detected together, the signature value derived is 12. The φ value is used for denoting the angle at which is the target is rotated. The signature range is to adjust the inclusiveness of the signature. If the value of the signature range is 0, then the color of the target has to be precise as the trained signature. The minimum brightness value is the value of the environment brightness required for detection, and the camera brightness value is for adjusting the brightness of the camera.

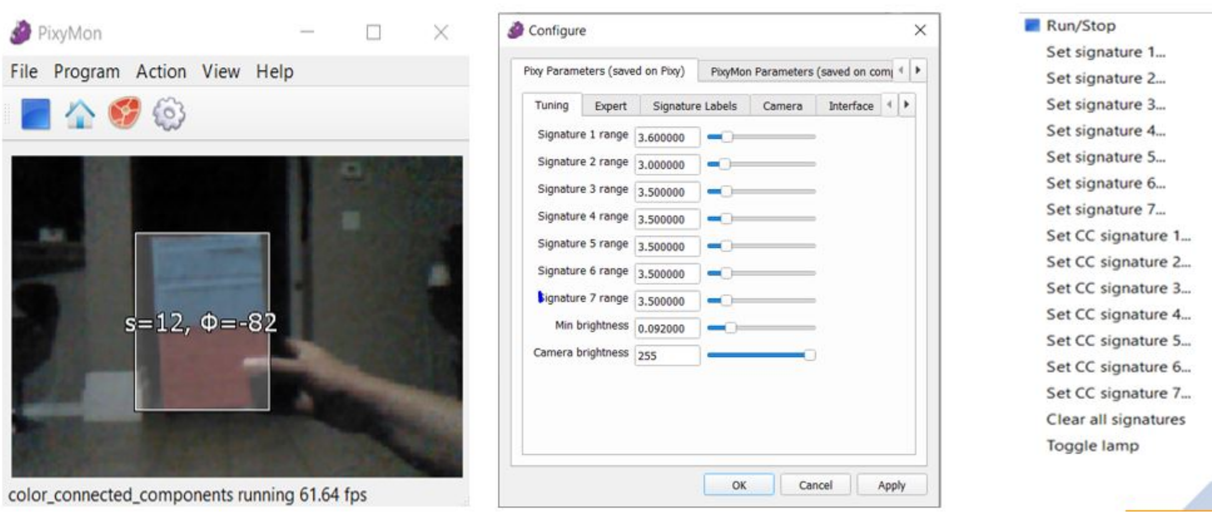

Figure 2.21: PixyMon software

The block array contains all the signatures that are identified as targets in the given frame. This block array is returned when the target is detected by the vision-based sensor Pixy CMUCam5 cmu camera in the given frame. For every signature, the x and y value of the target is returned, which indicates the bounding boxes of the object with reference to the frame. These x and y values are used in the algorithm for calculating the area of the object.

The algorithm for object following is shown in figure 2.22:

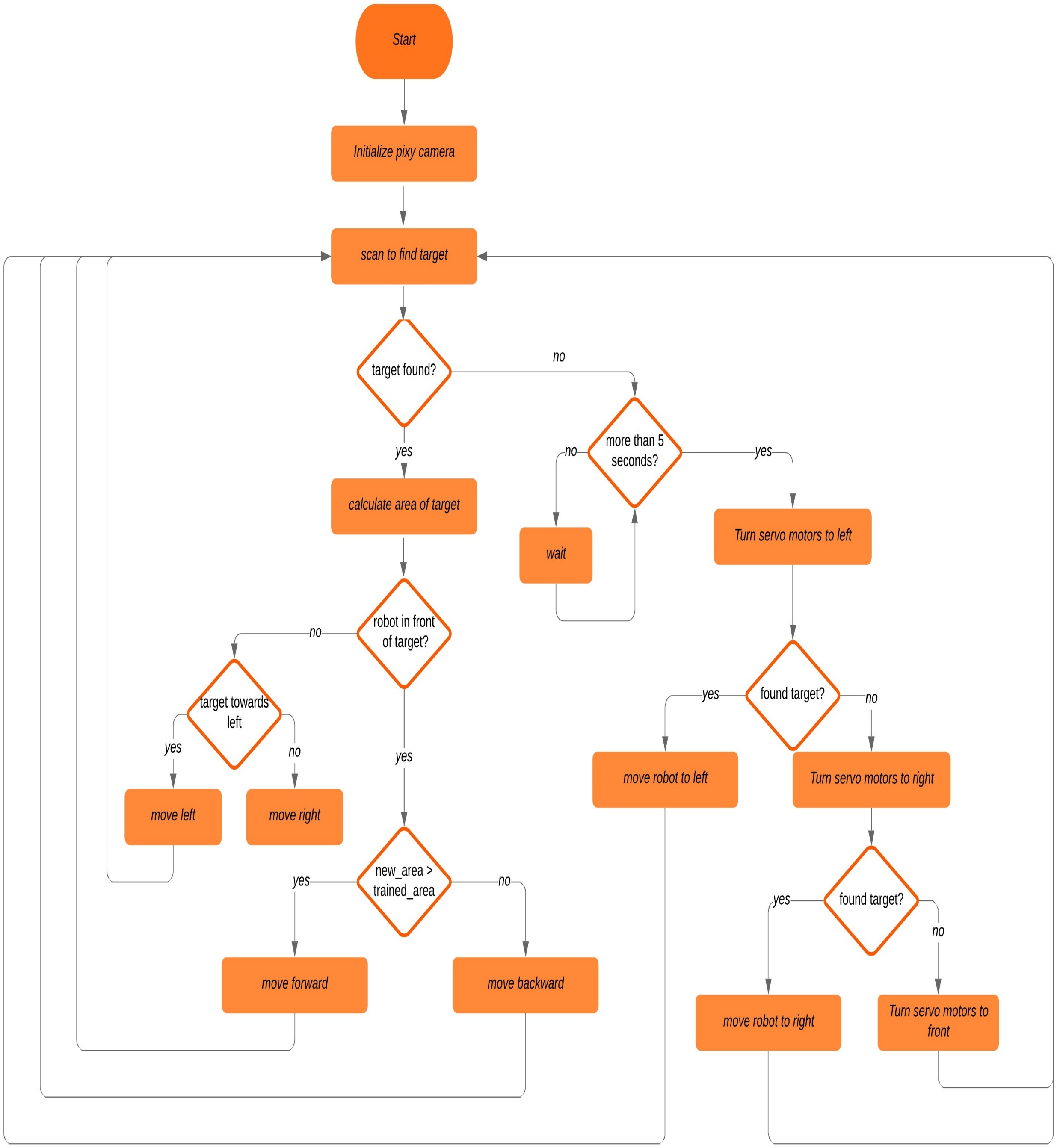

Figure 2.22: Object Follower algorithm

When the Pixy CMUCam5 camera is trained, each object is saved as a signature. Every signature has x and y values. When the robot is in the follower mode, the Pixy CMUCam5 camera is initialized and, it scans to find the target. When the target is found, the pixy CMU camera returns the x and y value, which is used to calculate the area of the target. If the area of the target is larger than the area of the trained object, then it moves backward. If the area of the found target is smaller than the area of the trained object, it moves forward. If the area of the object is smaller, and the object has the X value, which is lesser than the X minimum (which is the center of the frame), then it moves towards the left. If the object has an X value, which is higher than the X minimum, then it moves towards the right.

In that event, if any obstacle is detected at any point, then it will move backwards. During the process of moving towards the object, if the target is lost from the vision of the robot, the robot waits for five seconds. After five seconds, the servo motors move towards left. If the target is found at the left of the robot, then the robot moves towards the left and again scans for the target repeating the procedure above. If the target is not found, then the servo motor turns towards the right. If the target is on the right side of the robot, then the robot turns towards the right to scan for the target again and follow it. In case the target is not found at left or right, the servo motor keeps repeating the procedure every 5 seconds until it finds the target. This is shown in figure 2.23. The first image shows the servo motors turned towards left, and the second image shows the servo motors in the center, and the third image shows the servo motors turned towards the right.

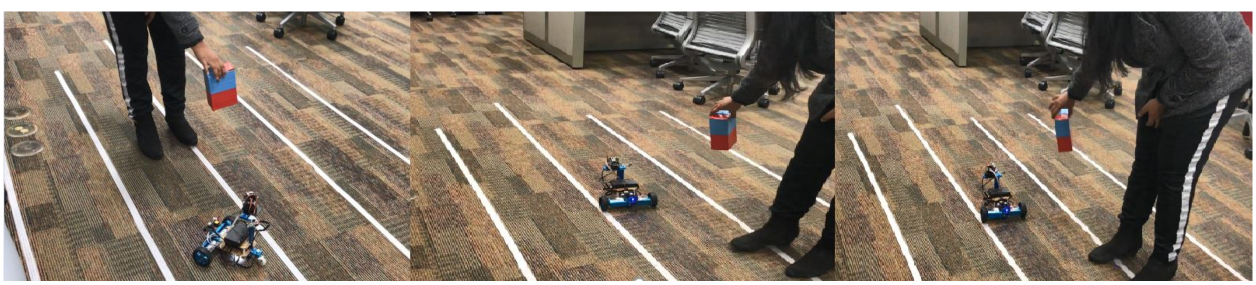

Figure 2.23: Rotation of servo motors

The combination of the Pixy CMUCam5 camera and the servo motor increases the efficiency of the entire system. The field of view of the Pixy CMUCam5 camera is approximately 60°. When the servo motor is added to the camera, the effective field of view becomes 180°. This helps to find the target even when it is not directly in front of the camera without rotating the robot. 0°indicates the left side of the robot and, 180°indicates the right side of the robot.

The follower mode ends after 15 minutes.

Go home mode

Once the robot is done following, it returns to its home position using the line follower. The line follower in this system is done using white lines. The line follower sensors have two sensors. The sensors are S1 and S2. If S1 and S2 detect the black line, then its value is IN. If it does not detect the black line, then its value is OUT[3]. In the lab, the carpet is black, and the lines are white. When both sensors have the value as OUT, it indicates the white line is found. Whenever the white line is found, it moves forward. If S1 is OUT and S2 is IN, then the robot moves towards the left. If S2 is OUT and S1 is IN, then the robot moves right. The robot uses color sensors to track the red color. Whenever it reaches the red color, it indicates that the home position is found, and it needs to stop. This is shown in figure.2.24

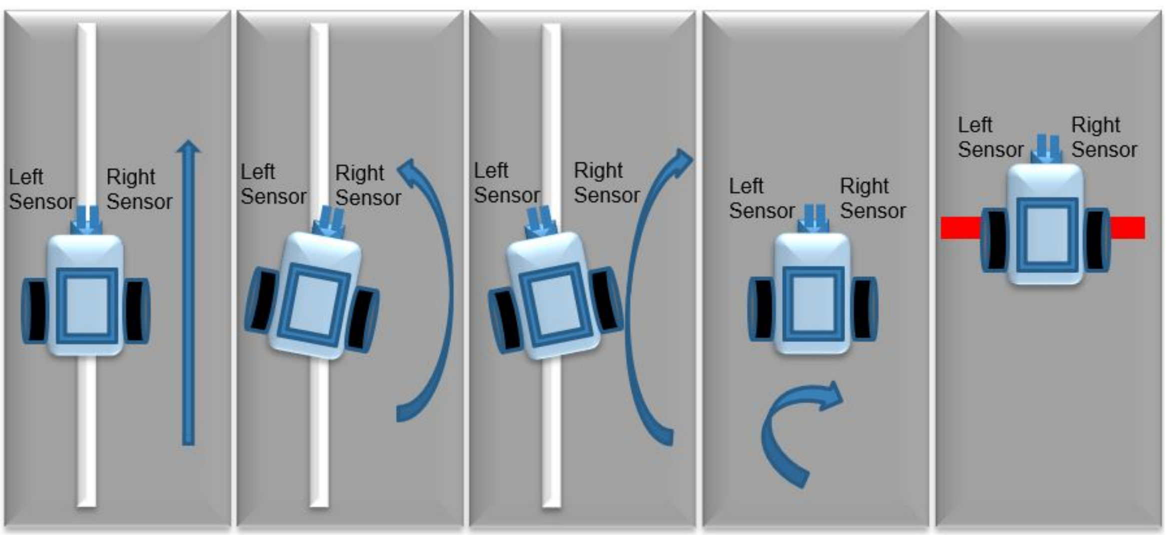

Figure 2.24: Line Follower

The algorithm for line follower module is as shown in figure 2.25:

The line follower algorithm uses the sensor1 and sensor2 values to determine the direction in which the robot has to move as well as it has the obstacle detection and avoidance implementation which ensures that if an object is within a range of 9 cm then the robot tries to move in the left direction in order to avoid the object.

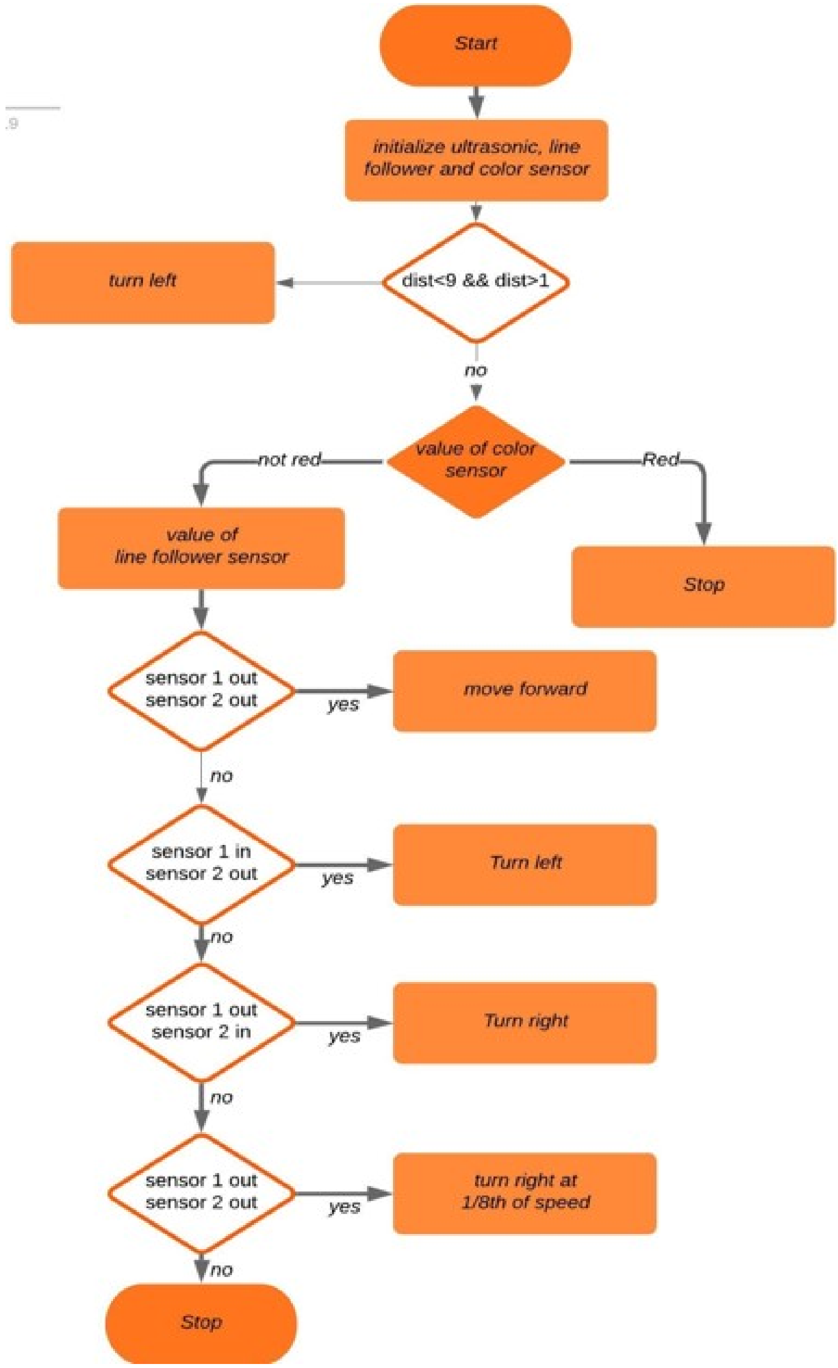

Figure 2.25: Line Follower algorithm

CHAPTER III: EVALUATION AND RESULT

To check the working of the system and to find the accuracy of the components, the system was analyzed under various conditions and by using different parameters.

3.1 Face Recognition

Analysis of face recognition algorithm accuracy with different threshold values

We have compiled the results of the use of the above algorithms on a few data-sets as detailed herein. We have categorized our results into two sections, each using a different threshold for face match i.e., 0.4, and 0.6. The face recognition algorithm returns a value in the range between 0 and 1, depending on how close face match is obtained from the input image and the trained data-set. A threshold is used to indicate that the match is positive or negative. This threshold can be used for determining the accuracy of the system.

To test our system, we made use of five different data-sets.

1. Data-Set 1: It consists of 200 images. Approximately 100 images are of former USA president Barack Obama, and 100 images are of myself.

2. Data-set 2: consists of 200 images. 100 images are of former USA president Barack Obama, and the other 100 images are of current USA president Trump.

3. Data-set 3: It consists of 300 images. 100 images are of former USA president Barack Obama, 100 images are of current USA president Donald Trump, and 100 images of myself.

4. Data-set 4: It consists of 300 images. 100 images are of former USA president Barack Obama, 100 images are of current president Trump, 100 images are of Salman Khan.

5. Data-set 5: consists of 300 images. The images are of all four people, i.e., former USA president Barack Obama, current USA president Trump, and Salman Khan and myself in any random number.

Test No	Number of Images	Positive	Negative	Accuracy %	Error%
1	200	197	12	94.26	5.74
2	200	195	15	92.86	7.14
3	300	286	22	92.87	7.13
4	300	294	16	94.84	5.16
5	300	295	13	95.78	4.22

Table 3.1: Analysis of face recognition algorithm accuracy with a threshold 0.4

Test No	Number of Images	True Positive	False Positive	Accuracy %	Error%
1	200	192	8	96.00	4.00
2	200	189	9	95.45	4.55
3	300	288	17	94.43	5.57
4	300	286	14	95.33	4.67
5	300	284	11	96.27	3.73

Table 3.2: Analysis of face recognition algorithm accuracy with threshold 0.6

The results of the experiments are shown in table 3.1, which uses the threshold of 0.4 and table 3.2, which uses the threshold of 0.6.

Accuracy measurement:

On analyzing the result sets at different thresholds, we observed that the results set for the experiment with a threshold value set to 0.4 gave us the highest accuracy, i.e., 98.5% but also the highest error percentage that was found in this case, i.e., 7.5%.

The result set for the experiment with the threshold value set to 0.6 gave the accuracy of 96%, and the error was reduced to 5.6%.

Time Measurement:

When the threshold is 0.4, the time required to recognize a face is approximately 1.9 seconds. When the threshold is 0.6, the time required to recognize a face is approximately 1.6 seconds. In the prior work, the best time that was recorded in the paper named an infrastructure for interactive environments was approximately 4 seconds for one iteration when the confidence value is set to 90%. This confidence value corresponds to the threshold value of approximately 0.6 in our system.

Hence we can observe that the time taken for an iteration is reduced by approximately two seconds using our model with the threshold of 0.6 and the threshold of 0.4.

3.2 People Counter

The system is tested for finding the Nth frame and the appropriate camera angle to get maximum accuracy.

Finding the Nth frame

We have made use of the Webcam Logitech V-UAS14 for body detection and tracking. This camera is mounted at a distance of approximately 13 feet above the ground, and its speed is 30 fps. The position of the camera is perpendicular to the wall to capture the top-view of the frame.

Object detection is a computationally expensive algorithm, and hence it cannot be executed for every second frame. The object detection is done at Nth frame and object tracking is done for every frame until (N-1)th. The algorithm was executed by using various values to determine the optimal value of the Nth frame that can be used to process object detection and ensure that the system is running with maximum accuracy.

Two videos were tested by sampling several Nth frame values, and the accuracy was calculated. The first video had the majority of human figures, while the second video had a combination of human figures and other objects such as shopping bags, baby carriages, etc. Each video was 6 minutes long. The results are :

For the first video consisting of human figures only, we plotted the frame number along the X-axis and the associated error % along the Y-axis. It can be seen from the graph in figure 3.28 that the values suitable are 25, 45, 60, and 75 gives maximum accuracy where the error % is 0, as seen in figure 3.1.

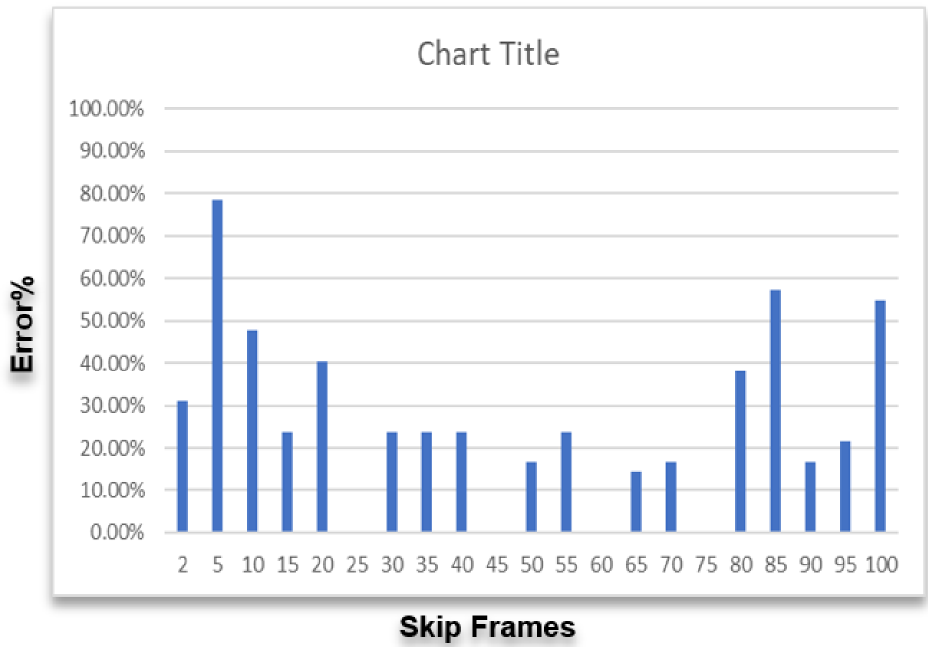

Figure 3.1: Graph of accuracy for video 1

For the second video consisting of human figures and other objects, we plotted a similar graph and concluded that the most accurate result was achieved when the frame value was set to 25, as seen in figure 3.2.

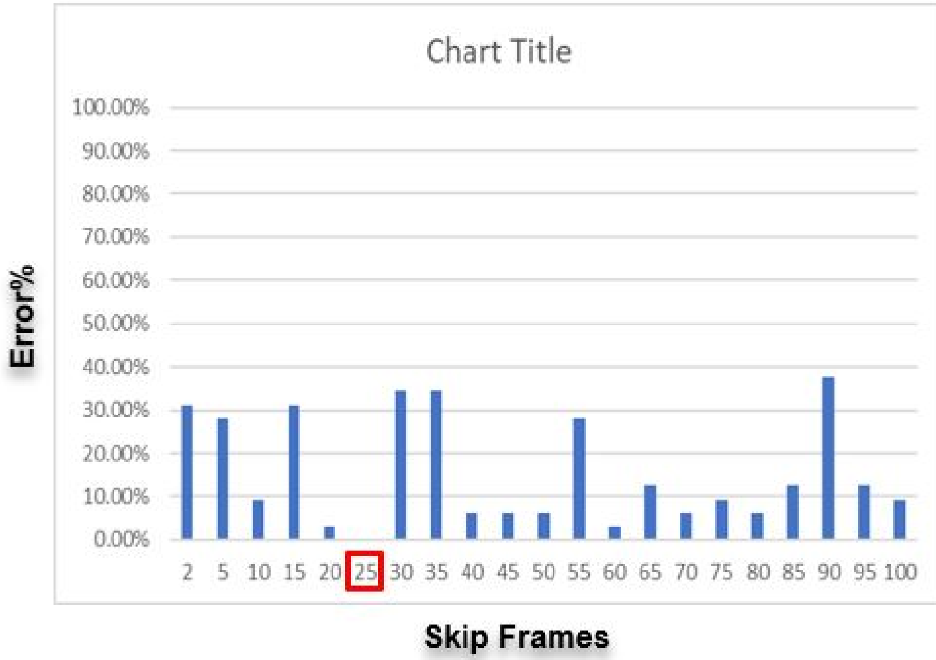

Figure 3.2: Graph of accuracy for video 2

Based on the observation of these two videos, we sampled the live video stream in our system using different frame values between 2 and 100. Figure 3.3 shows the result of 4 live videos used for finding the Nth frame. The live videos were 6 minutes long. Video 1 had one person in one frame. Video 2 had two people in one frame. Video 3 had four people in one frame, and video 4 had four people with various objects like chairs, ladders, etc., in one frame. We concluded that for the live video, when the Nth frame value was set to 25, the result was 98% accurate.

Finding the appropriate angle of the camera

For the people counter module, for accurate performance, the camera angle needs to ascertain. In order to do so, we conducted an experiment that was conducted with 2 people and 4 people in the frame. The camera was set at an angle of 2°and 8°.The count of people entering and leaving the lab is counted and compared with the up count, and down count returned by the algorithm for every input video to determine accuracy. values are The maximum accuracy was found when the camera was at an angle of 2°as can be seen in table 3.3.

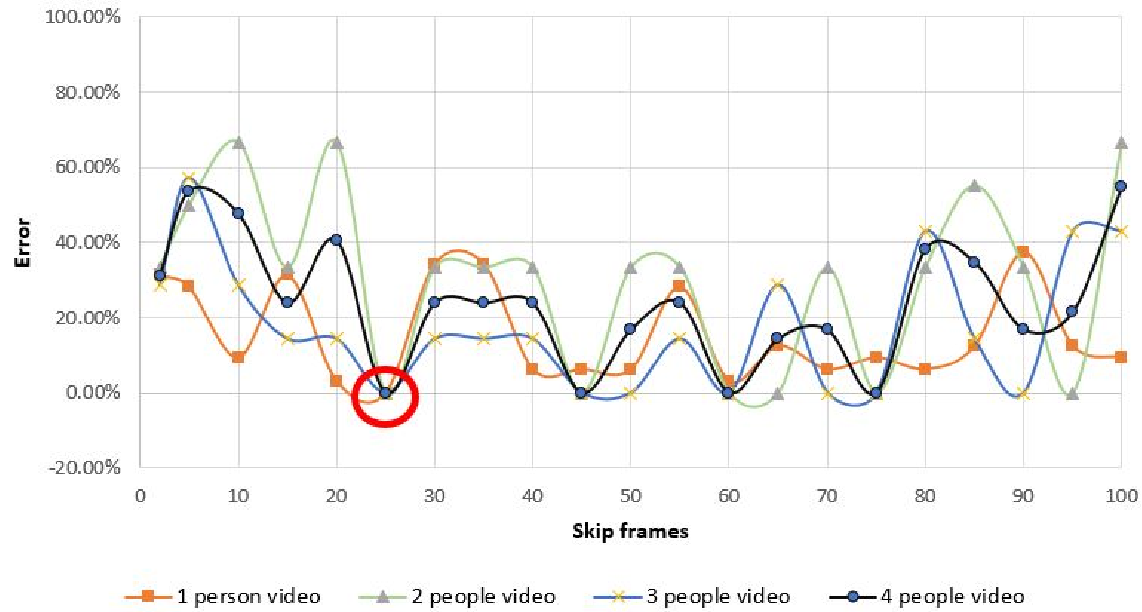

Figure 3.3: Graph of accuracy for all videos

Angle	Number of People in the frame	Accuracy
2°	2 people	98.83%
2°	4 people	91.23%
8°	2 people	89.33%
8°	4 people	83.54%

Table 3.3: Camera Angle Performance

3.3 Follower Robot

The follower robot is tested for its performance with objects of different size and the minimum brightness at which the robot works. The performance is evaluated by analyzing the behavior of the robot with varied conditions.

Performance is shown between two targets with different size

The distance of the object and the size of the object matters in the follower robot algorithm. We experimented with the impact of size and distance on the performance of the robot and analyzed the behavior as detailed below.

The experiment was conducted using two objects of one object's area was $8.75 inch^2$, and the other object's area was $24 inch^2$, as shown in figure 3.4. The objects are as shown in the below figure.

Figure 3.4: objects of different size

The result of the analysis is as follows, we have made use of the below equation to provide the relative ratio of the size of the object (area) and the distance (maximum distance):

$$Ratio = \frac{Area}{Maximum\ distance} \tag{2}$$

Results suggest that if the area of the object is higher, then the distance between the object and the robot can be higher, and if the area of the object is smaller, the distance between the object and the robot has to be smaller.

This means that if the size of the object is larger, then the robot can follow it to a farther distance. The free flow distance mentioned in the table refers to the distance after which the robot will move

Area	Free flow	Maximum distance	Ratio Area : maximum distance	Ratio Area : free flow
$8.75inch^2$	53 inch	95 inch	10.86	6.06
$24inch^2$	106 inch	145 inch	6.04	4.42

Table 3.4: Performance shown between two targets with different size

freely, i.e., suppose the distance between the object and the robot is 95 inches, and the size of the object is 8.75 inches the robot will move towards the object in a move-pause-move style till 53 inches of distance after which it will follow the object freely as it will be able to detect the object clearly after that point.

Comparison of the effect between bright and dark surroundings

The PixyMon software of the camera has the option of setting the brightness for the camera, as shown in figure 3.5. In order to find the effect of bright and dark surroundings on the performance of the robot, the minimum working brightness required by the robot was needed to be ascertained. We found this value by decreasing the brightness value in the PixyMon software by one and observing the performance. The maximum value of brightness is 255, at which the target looks

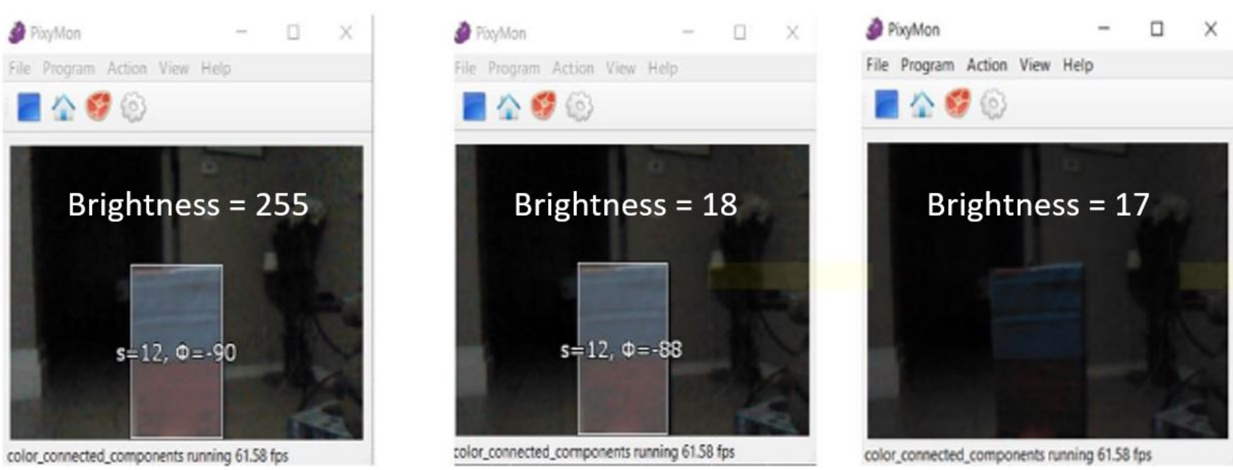

Figure 3.5: Field of view

as shown in the first figure below. The second figure is that when the brightness is reduced to 18, the target is still visible, and the object tracking and following will be performed. The last image refers to brightness level 17. At this point, the target is not visible, and the object detection and following cannot be performed.

Thus concluding that a minimum brightness of 18 is required for the smooth functioning of the robot.

CHAPTER IV: FUTURE WORK

The object detection and tracking are 98% accurate, and 2% error is for those people who are working close to each other, and object IDs get swapped. Using different object detector techniques can resolve this particular problem. The area of the blobs detected by *SSD* along with the smallest euclidean distance can solve the 2% error for assigning object ID. This way, the number of false detection will get reduced. There are specific object detection techniques like *YOLO* and *R-CNN* that are efficient for multi-object detection and tracking along with *SSD* and *MobileNets*. The combination of these algorithms can be applied at different Nth frame rates to ensure accurate tracking.

The obstacle avoidance in the current system is done just for detecting any obstacles in the front of the robot. Multiple ultrasonic sensors can be used in the left, right, and back of the robot to avoid obstacles from all sides. Taking different types of actions as per the signal received from the sensors ensures the robot is more efficient in tracking and obstacle avoidance. Figure 4.1 shows a similar approach that was used in the wheelchair follower system

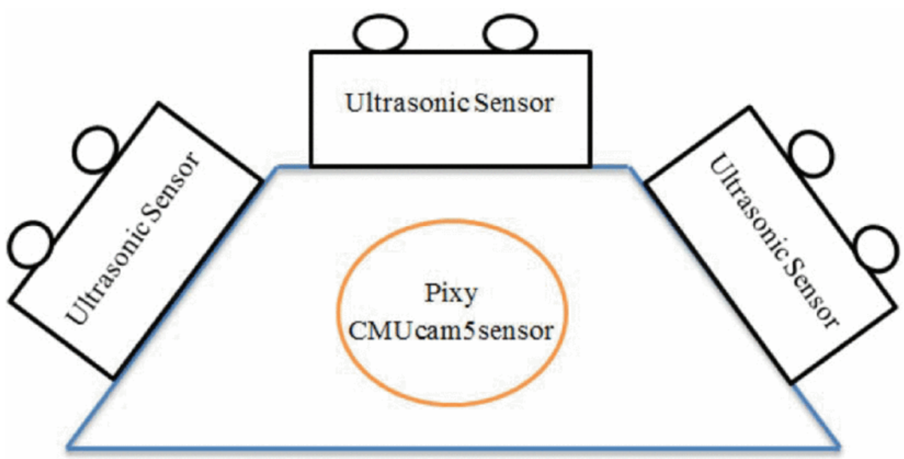

Figure 4.1: Pixy CMUCam5 with ultrasonic semsors
[1]

With the launch of Pixy CMUCam5 3 in 2020, which supports face signatures, the follower robot can be enhanced to use this and recognize faces along with following and tracking. These signatures will be similar to the CC signatures that are currently used in the system. Using the same algorithm, unknown people can be followed in the lab.

The servo motors currently increase the field of view in the horizontal direction from 60°to 180°. By adding a pan-tilt mechanism with the servo motors, the field of view of the current Pixy CMUCam5 camera can be increased in the vertical direction. Certain times if there are light fixtures in the ceiling, it affects the working of the vision sensor. The pan-tilt mechanism can be used to move the camera in a vertical direction if the target is not found. This mechanism can be implemented using the servo motors attached to the system.

The go-home mode currently is using a line follower system. A docking station with infrared lights can be used for replacing the current line follower system. The infrared beacon lights can be used to determine the position on the robot as per the docking station, which can help the robot return to its home position.

If enough memory is added to the Arduino board, the activity of unregistered people can be recorded, which can further be used for surveillance and security measures.

The sound-based interaction currently is used for greeting the person entering the lab. These commands can be used for taking actions as per the voice commands that are given by the user. For example, if the person is detected as a known person, he/she can have a designated workstation. The sound-based interaction commands can be used to turn on his/her workstation. Many such sound-based interaction voice commands can be integrated into our system.

For face recognition system, adding an administrative layer which provides features to add more images to the training data-set and to delete faces of people who no longer have access to the lab is required. This layer will ensure that only people with access to the lab are followed by the robot.

Such enhancements to this system can further increase the utility of this system, which can then be customized to be used in several layouts like home, intelligence/security departments and other similar environments. .

CHAPTER V: CONCLUSION

Follower robot in an integrated ambient intelligence system brings the digitized experience of access control and tracking into the lab. The three major components are facial recognition, people counter and follower robot which integrate to create a seamless system. This system detects motion if any person enters the lab. It recognizes the person and greets the person. Simultaneously, it tracks the number of people in a given room at any point in time. In case of entry of any unregistered users, the follower robot follows the person.

The system has made use of the best algorithms that are computationally light overall and hence provide the less computation time for a given data set. The system also is pioneering the space of interactive ambient intelligence systems by linking face recognition, the identification of the person by name, and then greeting them with a hello *name*. If the person is not recognized, he/she is greeted just with a hello, and a signal is sent to the follower robot.

This infrastructure can be applied in various interactive environments such as smart homes, smart home offices, smart labs, or surveillance systems. It has an improved processing time for face recognition using deep learning when compared to the prior work done in the same system. It has a simplified algorithm for object detection and object tracking along with the classification of the type of an object for people counting using only the webcams, which overcomes the flaws of all the previous work that has been mentioned. The simplified implementation can be used for developing tracking and detection algorithms and for other types of objects as well. Through analysis, we found that the computationally heavy object detection step in the algorithm can be used in the 25th frame, which makes the entire people counter algorithm faster

It has a simplified algorithm for the object following robot, which is implemented using the Arduino UNO board and generic libraries. It uses five different sensors for following the unregistered users. The infrared receiver receives a signal from the face recognition system if an unregistered user enters the lab. Using the vision sensor, we capture the video stream, which helps to follow the target, and the ultrasonic sensors help to avoid obstacles while tracking the target. The line follower and the color sensors are used for go-home mode, which allows the robot to return to its home position. Thus this particular algorithm can be used for any robot that has an Arduino board and is not specific to any brand or a particular type of robot.

The most significant advantage of this system is that the entire system can be decoupled, and different components can be used together or individually to create further synergistic systems. In future work, we are going to continue improving performances of face recognition, follower robot, enhance the people counter algorithm accuracy, and also enrich interactions.

REFERENCES

[1] M. F. Ahmad et al. “Colour tracking technique by using pixy CMUcam5 for wheelchair luggage follower”. In: *2017 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. Nov. 2017, pp. 186–191. DOI: 10.1109/ICCSCE.2017.8284402.

[2] M. F. Ahmad et al. “Visual based sensor cart follower for wheelchair by using microcontroller”. In: *2015 IEEE International Conference on Control System, Computing and Engineering (ICCSCE)*. Nov. 2015, pp. 123–128. DOI: 10.1109/ICCSCE.2015.7482170.

[3] F. R. Ali and A. T. Rashid. “Design and implementation of static and dynamic objects store systems using line follower robots”. In: *2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*. Mar. 2018, pp. 37–42. DOI: 10.1109/ICASEA.2018.8370952.

[4] E. J. Almazán and G. A. Jones. “Tracking People across Multiple Non-overlapping RGB-D Sensors”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. June 2013, pp. 831–837. DOI: 10.1109/CVPRW.2013.124.

[5] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. “OpenFace: A general-purpose face recognition library with mobile applications”. In: 2016.

[6] T. Baltrušaitis, P. Robinson, and L. Morency. “OpenFace: An open source facial behavior analysis toolkit”. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2016, pp. 1–10. DOI: 10.1109/WACV.2016.7477553.

[7] T. Baltrusaitis et al. “OpenFace 2.0: Facial Behavior Analysis Toolkit”. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. May 2018, pp. 59–66. DOI: 10.1109/FG.2018.00019.

[8] N. R. Barai, M. Kyan, and D. Androutsos. “Human visual system inspired saliency guided edge preserving tone-mapping for high dynamic range imaging”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. Sept. 2017, pp. 1017–1021. DOI: 10.1109/ICIP.2017.8296435.

[9] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. “Eigenfaces vs. Fisherfaces: recognition using class specific linear projection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7 (July 1997), pp. 711–720. ISSN: 0162-8828. DOI: 10.1109/34.598228.

[10] A. C. Braun, U. Weidner, and S. Hinz. “Support vector machines, import vector machines and relevance vector machines for hyperspectral classification — A comparison”. In: *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*. June 2011, pp. 1–4. DOI: 10.1109/WHISPERS.2011.6080861.

[11] A. Burbano, S. Bouaziz, and M. Vasiliu. “3D-sensing Distributed Embedded System for People Tracking and Counting”. In: *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*. Dec. 2015, pp. 470–475. DOI: 10.1109/CSCI.2015.76.

[12] A. B. Chan and N. Vasconcelos. “Counting People With Low-Level Features and Bayesian Regression”. In: *IEEE Transactions on Image Processing* 21.4 (Apr. 2012), pp. 2160–2177. ISSN: 1941-0042. DOI: 10.1109/TIP.2011.2172800.

[13] C. Chen, C. Chou, and F. Lian. “Trajectory planning for human host tracking and following of slave mobile robot on service-related tasks”. In: *2011 IEEE International Conference on Robotics and Biomimetics*. Dec. 2011, pp. 2419–2420. DOI: 10.1109/ROBIO.2011.6181666.

[14] Chin-Lung Chen, C. Chou, and F. Lian. “Active pedestrian following using laser range finder”. In: *2011 IEEE International Conference on Information and Automation*. June 2011, pp. 690–695. DOI: 10.1109/ICINFA.2011.5949082.

[15] D. Comaniciu and V. Ramesh. “Mean shift and optimal prediction for efficient object tracking”. In: *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*. Vol. 3. Sept. 2000, 70–73 vol.3. DOI: 10.1109/ICIP.2000.899297.

[16] N. Dalal and B. Triggs. “Histograms of oriented gradients for human detection”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. June 2005, 886–893 vol. 1. DOI: 10.1109/CVPR.2005.177.

[17] D. Do et al. “Implementation of an Integrated Ambient Intelligence System”. In: *2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC)*. Dec. 2017, pp. 1410–1416. DOI: 10.1109/ISPA/IUCC.2017.00214.

[18] G. Englebienne, T. van Oosterhout, and B. Krose. “Tracking in sparse multi-camera setups using stereo vision”. In: *2009 Third ACM/IEEE International Conference on Distributed Smart Cameras (ICDSC)*. Aug. 2009, pp. 1–6. DOI: 10.1109/ICDSC.2009.5289371.

[19] A. Faragasso, A. Yamashita, and H. Asama. “Versatile vision-based touch sensor for autonomous robots”. In: *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. June 2017, pp. 282–283. DOI: 10.1109/URAI.2017.7992730.

[20] P. F. Felzenszwalb et al. “Object Detection with Discriminatively Trained Part-Based Models”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.9 (Sept. 2010), pp. 1627–1645. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2009.167.

[21] J. García et al. “Directional People Counter Based on Head Tracking”. In: *IEEE Transactions on Industrial Electronics* 60.9 (Sept. 2013), pp. 3991–4000. DOI: 10.1109/TIE.2012.2206330.

[22] M. Harville. “Stereo person tracking with short and long term plan-view appearance models of shape and color”. In: *IEEE Conference on Advanced Video and Signal Based Surveillance, 2005*. Sept. 2005, pp. 522–527. DOI: 10.1109/AVSS.2005.1577323.

[23] P. Huang et al. “Object following based on SURF for mobile robots”. In: *The 1st IEEE Global Conference on Consumer Electronics 2012*. Oct. 2012, pp. 382–386. DOI: 10.1109/GCCE.2012.6379634.

[24] O. H. Jafari, D. Mitzel, and B. Leibe. “Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras”. In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. May 2014, pp. 5636–5643. DOI: 10.1109/ICRA.2014.6907688.

[25] Jun Yang et al. “A research of calculating the shortest distance in euclidean network”. In: *2009 IEEE International Conference on Network Infrastructure and Digital Content*. Nov. 2009, pp. 973–975. DOI: 10.1109/ICNIDC.2009.5360837.

[26] S. Kanagarajan and S. Ramakrishnan. “Development of ontologies for modelling user behaviour in Ambient Intelligence environment”. In: *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*. Dec. 2015, pp. 1–6. DOI: 10.1109/ICCIC.2015.7435736.

[27] V. Kazemi and J. Sullivan. “One millisecond face alignment with an ensemble of regression trees”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[28] B. Leibe et al. “Coupled Object Detection and Tracking from Static Cameras and Moving Vehicles”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.10 (Oct. 2008), pp. 1683–1698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2008.170.

[29] J. Li, Y. Ho, and J. Huang. “Line Tracking with Pixy Cameras on a Wheeled Robot Prototype”. In: *2018 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*. May 2018, pp. 1–2. DOI: 10.1109/ICCE-China.2018.8448948.

[30] Y. Li et al. “Simultaneous Facial Feature Tracking and Facial Expression Recognition”. In: *IEEE Transactions on Image Processing* 22.7 (July 2013), pp. 2559–2573. ISSN: 1941-0042. DOI: 10.1109/TIP.2013.2253477.

[31] K. Madhira and A. Shukla. “Pedestrian flow counter using image processing”. In: *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. Aug. 2017, pp. 1911–1915. DOI: 10.1109/ICECDS.2017.8389782.

[32] G. L. Mariottini et al. “Vision-Based Localization for Leader–Follower Formation Control”. In: *IEEE Transactions on Robotics* 25.6 (Dec. 2009), pp. 1431–1438. ISSN: 1941-0468. DOI: 10.1109/TRO.2009.2032975.

[33] Y. Nagumo and A. Ohya. “Human following behavior of an autonomous mobile robot using light-emitting device”. In: *Proceedings 10th IEEE International Workshop on Robot and Human Interactive Communication. ROMAN 2001 (Cat. No.01TH8591)*. Sept. 2001, pp. 225–230. DOI: 10.1109/ROMAN.2001.981906.

[34] E. Naz, U. Farooq, and T. Naz. “Analysis of Principal Component Analysis-Based and Fisher Discriminant Analysis-Based Face Recognition Algorithms”. In: *2006 International Conference on Emerging Technologies*. Nov. 2006, pp. 121–127. DOI: 10.1109/ICET.2006.335932.

[35] X. Qu et al. “A Fast Face Recognition System Based on Deep Learning”. In: *2018 11th International Symposium on Computational Intelligence and Design (ISCID)*. Vol. 01. Dec. 2018, pp. 289–292. DOI: 10.1109/ISCID.2018.00072.

[36] A. Rowe, C. Rosenberg, and I. Nourbakhsh. “A Second Generation Low Cost Embedded Color Vision System”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05) - Workshops*. Sept. 2005, pp. 136–136. DOI: 10.1109/CVPR.2005.396.

[37] D. Sanjay, T. S. Savithri, and P. R. Kumar. “Person follower robotic system”. In: *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*. July 2014, pp. 1324–1327. DOI: 10.1109/ICCICCT.2014.6993166.

[38] F. Schroff, D. Kalenichenko, and J. Philbin. “FaceNet: A unified embedding for face recognition and clustering”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015, pp. 815–823. DOI: 10.1109/CVPR.2015.7298682.

[39] M. S. Sefat, D. K. Sarker, and M. Shahjahan. “Design and implementation of a vision based intelligent object follower robot”. In: *2014 9th International Forum on Strategic Technology (IFOST)*. Oct. 2014, pp. 425–428. DOI: 10.1109/IFOST.2014.6991154.

[40] Z. Shaik and V. Asari. “A Robust Method for Multiple Face Tracking Using Kalman Filter”. In: *36th Applied Imagery Pattern Recognition Workshop (aipr 2007)*. Oct. 2007, pp. 125–130. DOI: 10.1109/AIPR.2007.21.

[41] M. N. Soorki, H. A. Talebi, and S. K. Y. Nikravesh. “Robust leader-following formation control of multiple mobile robots using Lyapunov redesign”. In: *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*. Nov. 2011, pp. 277–282. DOI: 10.1109/IECON.2011.6119324.

[42] Y. Taigman et al. “DeepFace: Closing the Gap to Human-Level Performance in Face Verification”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. June 2014, pp. 1701–1708. DOI: 10.1109/CVPR.2014.220.

[43] B. S. Totada and S. D. Cabrera. “Detection of people from time-of-flight depth images using a cell-tracking methodology.” In: *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. Dec. 2018, pp. 084–089. DOI: 10.1109/ISSPIT.2018.8642733.

[44] H. Tsunakawa et al. “Contrastive Relevance Propagation for Interpreting Predictions by a Single-Shot Object Detector”. In: *2019 International Joint Conference on Neural Networks (IJCNN)*. July 2019, pp. 1–9. DOI: 10.1109/IJCNN.2019.8851770.

[45] P. Vadakkepat et al. “Multimodal Approach to Human-Face Detection and Tracking”. In: *IEEE Transactions on Industrial Electronics* 55.3 (Mar. 2008), pp. 1385–1393. ISSN: 1557-9948. DOI: 10.1109/TIE.2007.903993.

[46] W. Wang and J. Chang. “Implementation of a mobile robot for people following”. In: *2012 International Conference on System Science and Engineering (ICSSE)*. June 2012, pp. 112–116. DOI: 10.1109/ICSSE.2012.6257159.

[47] D. Wolf and G. S. Sukhatme. “Online simultaneous localization and mapping in dynamic environments”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*. Vol. 2. Apr. 2004, 1301–1307 Vol.2. DOI: 10.1109/ROBOT.2004.1308004.

[48] Youngwoo Yoon, Hosub Yoon, and Jaehong Kim. “Person following with a RGB-D camera for mobile robots”. In: *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. Nov. 2012, pp. 191–191. DOI: 10.1109/URAI.2012.6462968.