IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 3, June 2018
ISSN (Online): 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Human Action Recognition System for Automation Application

[1] Sai Kailash; [2] Purav M Shah; [3] Sai Karthik; [4] B Madhukar; [5] K Vijaya

[1, 2, 3, 4] Electronics and Communication Engineering (ECE) Department, B.M.S College of Engineering, Bengaluru, Karnataka - 560019, India

[5] Electronics and Communication Engineering (ECE) Department, B.M.S College of Engineering Bengaluru, Karnataka - 560019, India

Abstract - An increasing demand for comfort in personal life has motivated in-depth research in home automation systems. Several automation techniques in existence utilize sensors and actuators, but the use of video processing systems in automation systems would benefit the physically challenged, especially the visually challenged, for using the automation system effectively. This project's central theme lies in automating the process of human gesture recognition through visual-based recognition systems. While traditional approaches operate on 2-D images and use computationally intensive algorithms and high dimensional features for activity recognition, recently, the introduction of RGB depth cameras has motivated the development of a recognition system with lower-dimensional features; the system uses less complex algorithms and a faster method. The automation system developed here uses a visual recognition system implemented using Matlab. The recognition system uses a Microsoft Kinect Sensor as a video capture device and a Fuzzy Inference system for making decisions. The automation system was developed using a hardware setup consisting of a microcontroller unit and the devices to be controlled. The complete system consisting of a video capture unit, action recognition system, and an automation system was implemented. The proposed action recognition system was designed to recognize two different user gestures and control two different devices. The implemented system shows real-time performance suitable for smart home automation systems.

Keywords – *Matlab, Kinect, Automation, Recognition*

1. Introduction

Human gesture recognition is an interesting and challenging research topic in the computer vision area. The research is motivated by the increasing use of Human Action Recognition Systems in applications such as video surveillance systems, home care for older people, Human-Computer/Robot Interaction (HCI/HRI), video retrieval, virtual reality, computer gaming, and many other fields [1].

An action usually refers to a sequence of primitive movements carried out by a single object, that is, an atomic movement that can be described at limb level, such as a step while walking. However, an activity consists of a number of sequential actions. For example, dancing consists of successive repetitions of several actions, i.e., walking, jumping, and waving hands. Actions can be placed on a lower level than activities [2]. Human action recognition finds its application in various fields.

1. Surveillance system
   - Homeland security
   - Crime prevention
   - Traffic surveillance

2. Human-computer interaction
   - Extracting statistics for sports

3. Health care systems
   - Patient rehabilitation process
   - Monitoring elderly behavior

Automation systems utilizing various technologies exist today, including different sensors, applications, IoT devices, GSM, and speech signals have been seen around us. The visual-based approaches were initially used for monitoring applications, and recently, utilization of the visual system for automation is gaining importance due to their accuracy and usability.

2. Literature Survey

Over the past decade, a great deal of work has been done on recognizing human activities. However, the problem is still open and provides a significant challenge to the researchers, and more rigorous research is needed to come around it. An overview of the various action recognition methods and available well-known action datasets are provided in Taha et al. [5]. Previous research in gesture recognition was based on color or grayscale intensity images. These images are obtained from traditional RGB cameras, where each pixel's value represents the intensity of incoming light. It contains rich texture and color information, which is very useful for image processing. However, it is susceptible to illumination changes.

Recently, vision technologies can capture distance information from the real world, which cannot be obtained directly from an intensity image. These images are obtained from depth cameras, where the value of each pixel represents the calibrated distance between camera and scene. An advantage of using these sensors is that they give depth at every pixel, so the object's shape can be measured. While using depth images, computer vision tasks like background subtraction and contour detection become easier. There are many signs of progress, and improvements have been made with the use of depth information.

Based on the paragraphs above, there are two main approaches for human behavior recognition: RGB video-based approach in [3] and depth map-based approach in Ye et al. [5], Chen et al. [6]. This section focuses only on reviewing the state-of-the-art techniques that investigate the applicability and benefit of depth sensors for action recognition, especially skeleton-based approaches. The use of the different data provided by the RGB-D devices for human action recognition goes from employing only the depth data, or only the skeleton data extracted from the depth data, to the fusion of both the depth and the skeleton data. Existing skeleton-based human action recognition approaches (Vemulapalli et al. [7]) can be broadly grouped into two main categories: joint-based approaches and body part-based approaches. Joint-based approaches consider the human skeleton as a set of points, whereas body part-based approaches consider the human skeleton a connected set of rigid segments. Approaches that use joint angles can be classified as body part-based approaches since joint angles measure the geometry between directly connected pairs of body parts.

Jalal et al. [8] present a depth-based lifelogging human activity recognition system to recognize the daily activities of older adults and turn these environments into an intelligent living space. Initially, a depth imaging sensor is used to capture depth silhouettes. Based on these silhouettes, human skeletons with joint information are produced, which are further used for activity recognition and generating their life logs. The lifelogging system is divided into two processes. Firstly, the training system includes data collection using a depth camera, feature extraction, and training for each activity via Hidden Markov Models. Secondly, after training, the recognition engine starts to recognize the learning activities and produces life logs.

Gasparrini et al. [9] propose an automatic fall detection method using the Kinect depth sensor in the top-view configuration. Their approach allows detecting a fall event without relying on wearable sensors and by exploiting privacy-preserving depth data only. Starting from suitably preprocessed depth information, the system can recognize and separate the still objects from the human subjects within the scene using an ad-hoc discrimination algorithm. Several human subjects may be monitored through a solution that allows simultaneous tracking. Once a person is detected, he is followed by a tracking algorithm between different frames. The use of a reference depth frame, containing the setup of the scene, allows one to extract a human subject, even when he/she is interacting with other objects, such as chairs or desks.

Althloothia et al. [10] present two sets of features for human activity recognition using a sequence of RGB-D images: shape representation and kinematic structure. The shape features are extracted using the depth information in the frequency domain via spherical harmonics representation. The other features include the motion of the 3D joint positions (i.e., the end of the distal limb segments) in the human body. Both sets of features are fused using the Multiple Kernel Learning (MKL) technique at the kernel level for human activity recognition.

Wang et al. [11] present an Actionlet Ensemble Model for human action recognition with depth cameras. An actionlet is a particular conjunction of the features for a subset of the joints, indicating the features' structure. As there is an enormous number of possible actionlets, the authors propose a data mining solution to discover discriminative actionlets. An action is then represented as an Actionlet Ensemble, which is a linear combination of the actionlets, and their discriminative weights are learned via a multiple kernel learning method.

Ofli et al. [12] propose a skeletal motion feature representation of human actions, called Sequence of the Most Informative Joints (SMIJ). Specifically, in the SMIJ representation, a given action sequence is divided into a number of temporal segments. Within each segment, the joints that are deemed to be the most informative are selected. The sequence of such joints is then used to represent an action. One of the limitations of the SMIJ representation that remains to be addressed is its insensitivity to discriminate different planar motions around the same joint. The joint angles are computed between two connected body segments in 3D spherical coordinates, thus capturing only a coarse representation of the body configuration.

Kavita et al. [13] presented a review of multiple techniques involved in human motion detection and behavior recognition. A review on vision-based Hand Gesture Recognition and Applications was presented by Choudhary et al. [15]. A Gesture-based HCI for windows mouse control in PowerPoint presentations was presented by Osunkoya et al. [16]. The Kinect sensor's skeletal tracking ability was utilized here with the depth map and color image obtained by the sensor, which enables the user to operate Windows 7 Operating System and explore its functionality with no physical contact with a peripheral device such as a mouse. The predefined gestures recognized by the device allow the simulation of different commands and mouse behaviors.

A review on current hand gesture recognition systems was presented by Khan and Ibraheem [17]. The review presented the key issues of hand gestures, results of hand gesture methods, databases, and a comparison between main gesture recognition phases. The authors Cai et al. [18] presented an Infrared Human posture recognition for Smart home monitoring using Hidden Markov Models (HMM). The proposed model could identify the different body movement postures by observing the human posture sequence, matching identification, and classification process. The results show that the proposed method was feasible and effective for human posture recognition.

The author, David et al., presented a low-cost, flexible, and environmentally friendly home automation system [19]. It employs an embedded micro – web server in Arduino Mega 2560 microcontroller, with IP connectivity to remotely access and control devices and appliances. These devices can be controlled through a web application or via Bluetooth using an Android-based smartphone app.

A Smart Home Automation system based on IoT was presented by Imran et al. [20]. The system consists of a computer server with an internet connection, a modem connecting the server to the external network, an Arduino microcontroller with a hardwired application connected to the devices. The prototype system supports two-level devices that only need to be switched on or off.

3. Proposed Work

This section details the design aspects of the proposed Human Action Recognition System. The aspects include the model of the proposed system, the operation flow of the proposed system design, and their algorithmic implementations.

3.1 Action Recognition System

The system-level view of the proposed gesture recognition based home automation system is presented here with the aid of a block diagram, as shown in figure 1. The figure shows only a high-level view of the system design, and specific details about each block will be explained in the following section.

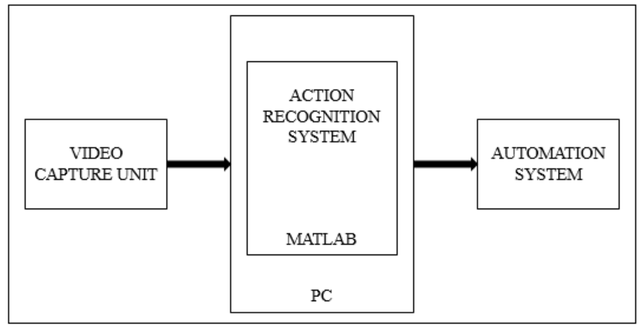

Figure 1: System block diagram

The system captures videos frame by frame using a video camera and passes the color and depth information to the action recognition system. This action recognition system is implemented as a software code that runs on a host machine. The software code uses the video frames, depth information, and color information obtained from the video sensor and constructs a skeletal image of the object. From the skeletal image coordinates, the object's movement can be tracked by finding the difference of the coordinates between the point of interest and a reference point. The motion-tracked can then be used to make a decision using suitable decision logic. The proposed system uses a Fuzzy logic-based decisive approach to make suitable decisions. The decision obtained from the action recognition system is then sent to a microcontroller unit to control suitable devices based on the decision made.

3.1.1 Video Capture Unit

The video capture unit is a block used in the proposed automation system to capture the video frames from the users making the gestures with the intention of controlling the devices. This unit provides the recognition system with the necessary information to identify a particular gesture. A Kinect sensor is used as a video capture unit to provide the required video information.

3.1.1.1 Kinect Sensor

The Kinect sensor is a high-resolution depth and RGB sensing device mainly used for object tracking, object detection, human activity analysis, hand gesture analysis, 3D mapping, and face expression detection. It primarily consists of an infrared projector, a color camera, and the IR camera.

The Kinect sensor's output is the RGB and depth information that will be fed as input to the Action Recognition System.

3.1.2 Action Recognition System

An action recognition system is implemented as a software system. This system is implemented as a Matlab code that runs on a host PC. The recognition system uses the depth information obtained from the Kinect sensor to track the motion and identify the gesture. The depth information is used to track the skeletal structure of the video frame. The skeletal structure coordinates are then obtained from which the right hand, left hand, and shoulder coordinates of the skeletal structure are obtained. Since the system will be designed for hand gesture recognition, the distance features of the right hand are calculated. The distance features are computed for the right hand as the difference between the (x,y,z) coordinates of the right hand and (x,y,z) coordinates of the shoulder. These features are fed as an input to the Fuzzy Inference System, making decisions based on certain threshold levels and taking suitable actions for the made decision. The system's actions can be in the form of displaying specific text or sending control information to other units for device control.

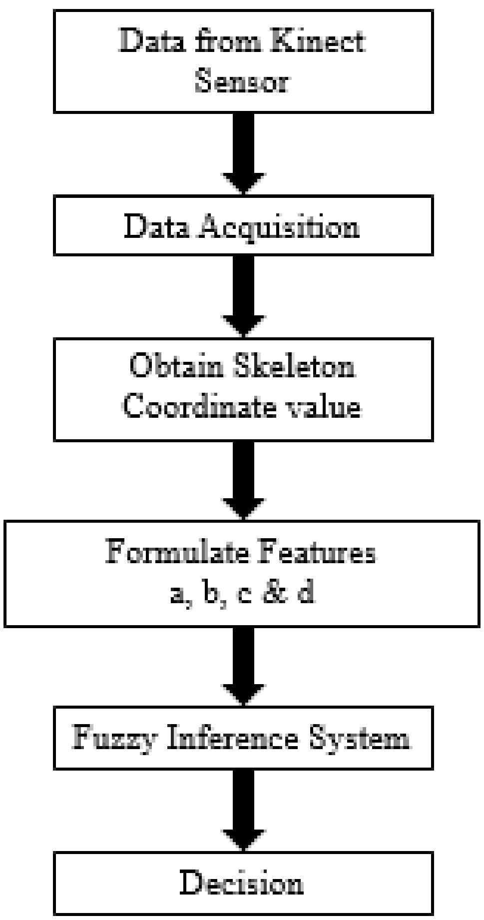

Figure 2: Action Recognition System

3.1.2.1 Data Acquisition from Kinect Sensor

The application must also be able to acquire the capabilities (such as the depth stream, the color stream, skeleton stream) needed from the Kinect sensor, which is essential for the successful implementation of the proposed system. This is done by enabling through coding the *ColorStream* component of the Kinect, which provides the RGB video stream; the *DepthStream*, which provides the 3D representation of the image in front of the sensor and enabling the *SkeletonStream*, which is used for acquiring the skeleton data.

3.1.2.2 Obtain Skeleton Coordinate Value

Kinect's skeletal tracking features combined with the NUI library allow users and their actions to be recognized. A human body is represented by a number of joints representing body parts such as the head, neck, shoulders, and arms. Each joint is represented by its 3D coordinates.

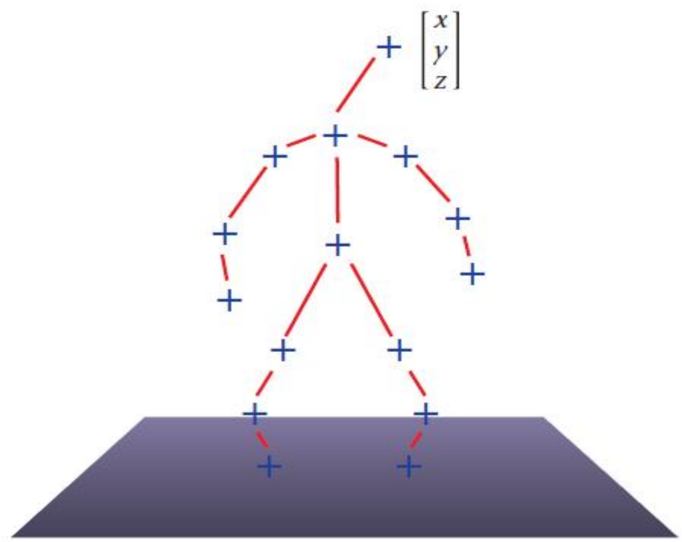

Figure 3: Skeletal structure representation

Software running on a PC uses the depth image obtained from the Kinect sensor, decodes it, and recognizes the image's elements as human body shapes. The software has been "trained" with a wide variety of body shapes. It uses the alignment of the various body parts and how they move to identify and track them. After skeleton tracking, each joint's position in 3D space is returned by the NUI library in the format of X, Y, and Z coordinates expressed in meters according to the skeleton space coordinate system.

3.1.2.3 Formulate Distance Features

The skeleton model of features computation is shown in the figure, where the skeleton coordinates will be captured from the video surveillance system using a Kinect sensor. These coordinates are used to calculate the features like a, b, c, and d where "a" is the distance between the right and left hand, "b" is the distance between the right hand and floor, "c" is the distance between the left hand and floor while "d" is the distance between the pelvis and floor. Then these features will be trained in the Fuzzy Inference System.

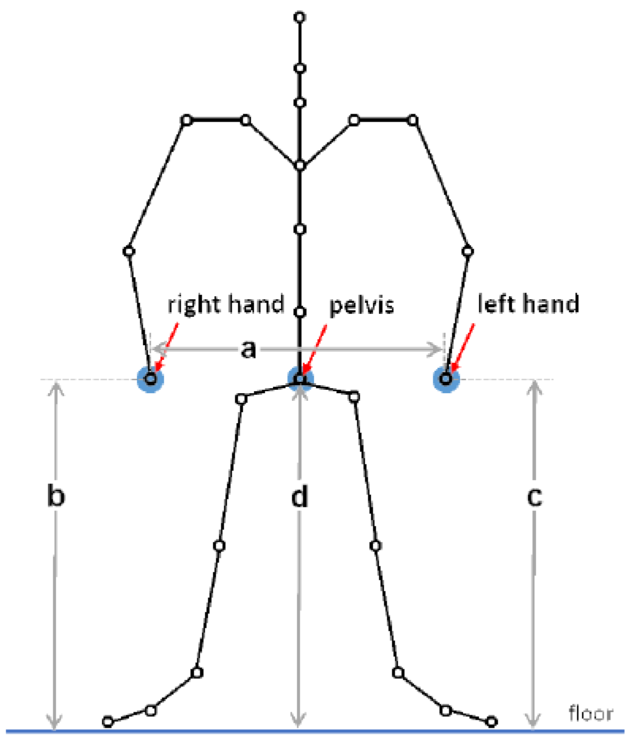

Figure 4: Skeleton model of features (a, b, c, and d) computation

The distance between two specific points or joints in a skeletal image are used as features here to identify a particular gesture. These distances may be obtained by taking the differences between a reference point on the skeletal structure and a point of interest. The point of interest is the right hand in the proposed system, and the shoulder point is taken as a reference. Thus distance features of the right hand from shoulder point are obtained by taking the difference of (x,y,z) coordinates of the right hand with (x,y,z) coordinates of shoulder point.

$$dxR = Rx-Sx;$$
$$dyR = Ry-Sy;$$
$$dzR = abs\,(Sz-Rz);$$

where
Rx → Right-hand x coordinate
Ry → Right-hand y coordinates
Rz → Right-hand z coordinates
Sx → Shoulder x coordinate
Sy → Shoulder y coordinate
Sz → Shoulder z coordinate

3.1.2.4 Fuzzy Inference System

Fuzzy Inference System (FIS) is the key unit of a fuzzy logic system with decision-making as its primary work. It uses the "IF…THEN" rules along with connectors "OR" or "AND" for drawing essential decision rules. FIS's output is always a fuzzy set irrespective of its input, which can be fuzzy or crisp. A de-fuzzy unit would be there with FIS to convert fuzzy variables into crisp variables.

I. Functional blocks of the FIS

- Rule Base − It contains fuzzy IF-THEN rules.
- Database − It defines the membership functions of fuzzy sets used in fuzzy rules.
- Decision-making Unit − It performs an operation based on rules.
- Fuzzy Interface Unit − It converts crisp quantities into fuzzy quantities.
- De-fuzzy Interface Unit − It converts the fuzzy quantities into crisp quantities.

Following is a block diagram of the Fuzzy Inference System.

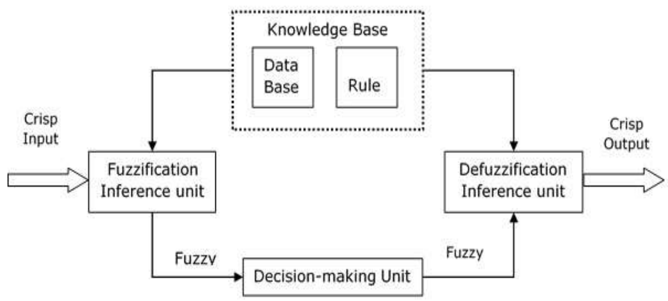

Figure 5: Block diagram of the Fuzzy Inference System

II. Steps for Computing the Output

Following steps need to be followed to compute the output from this FIS:

- Step 1 − Set of fuzzy rules need to be determined in this step.
- Step 2 − In this step, the input would be made fuzzy using the input membership function.

- Step 3 – Now establish the rule strength by combining the fuzzy inputs according to fuzzy rules.
- Step 4 – In this step, determine the consequent rule by combining the rule strength and the output membership function.
- Step 5 – For obtaining output distribution, combine all the consequents rules.
- Step 6 – Finally, a de-fuzzy output distribution is obtained.

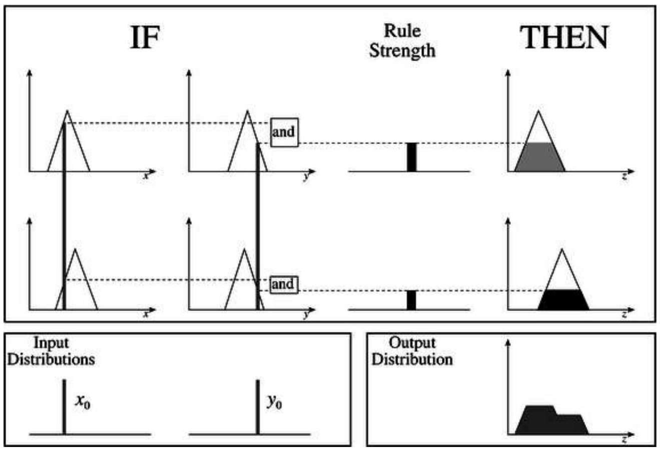

Figure 6: Figure 3.6 Block diagram of Mamdani Fuzzy Interface System

III. FIS Design

i. The FIS system is designed with four input variables dxR, dyR, dzR, and dxL. The four variables represent the (x,y,z) coordinates of the right hand and left hand.

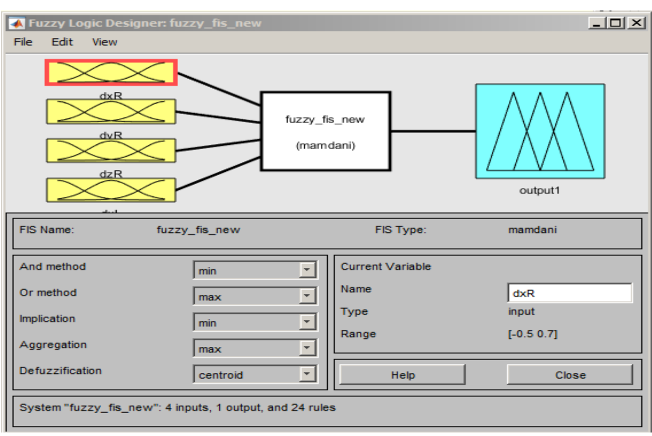

Figure 7: FIS design window

ii. Defining Membership function

The member functions for each variable defined are described here.

1. Membership function for dxR

The function for dxR is defined such that it is low up to 0.2 and high above 0.337, and any value in between 0.2 to 0.337 is fuzzy.

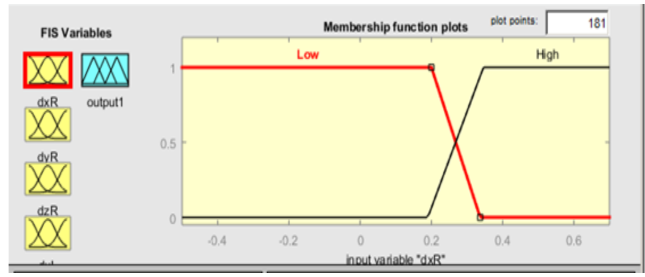

Figure 8: Membership function for dxR variable

2. Membership function for dyR

The function for dyR is defined such that it is low up to -0.4, medium between -0.3 to 0.3 and high above 0.4, and any value in between -0.37 to -0.45 and 0.24 to 0.42 is fuzzy.

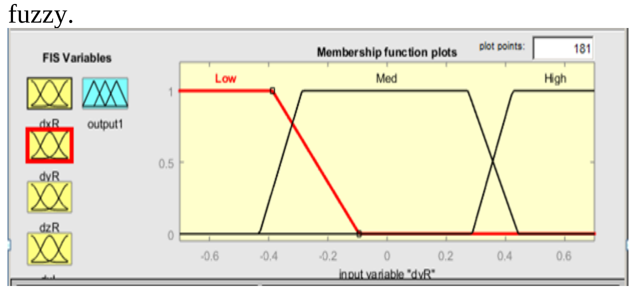

Figure 9: Membership function for dyR variable

3. Membership function for dzR

The function for dzR is defined such that it is low up to 0.22 and high above 0.39, and any value in between 0.22 to 0.39 is fuzzy.

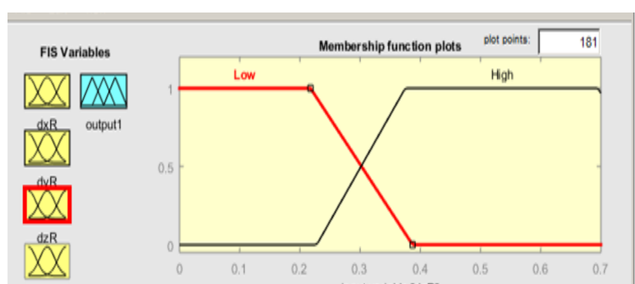

Figure 10: Membership function for dzR variable

4. Membership function for dxL

The function for dxL is defined such that it is low up to 0.4 and high above 0.5, and any value in between 0.4 to 0.5 is fuzzy.

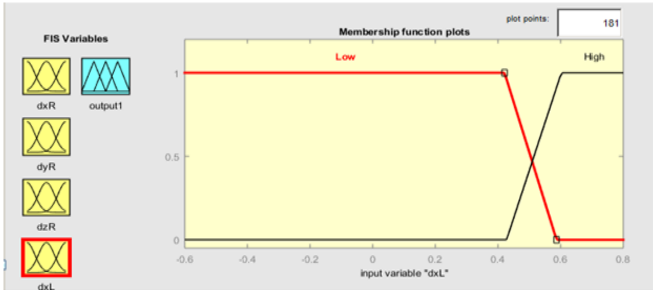

Figure 11: Membership function for dxL variable

iii. Defining Rules

The rules are defined, as shown in table 1.

Table 1: Rule table for the FIS system

dxR	dyR	dzR	dxL	Output
L	L	L	L	No signal
L	L	L	H	Bulb-2
L	L	H	L	No sign
L	L	H	H	Bulb2
L	M	L	L	No signal
L	M	L	H	Bulb-2
L	M	H	L	TV
L	M	H	H	Bulb-2
L	H	L	L	Fan
L	H	L	H	Bulb-2
L	H	H	L	Fan
L	H	H	H	Bulb-2
H	L	L	L	No signal
H	L	L	H	Bulb-2
H	L	H	L	No signal
H	L	H	H	Bulb-2
H	M	L	L	Bulb-1
H	M	L	H	Bulb-2
H	M	H	L	Bulb-1
H	M	H	H	Bulb-2
H	H	L	L	Bulb-1
H	H	L	H	Bulb-2
H	H	H	L	Bulb-1
H	H	H	H	Bulb-2

The fuzzy system rules are shown in figure 12, and the graphical view of the rules is shown in figure 13.

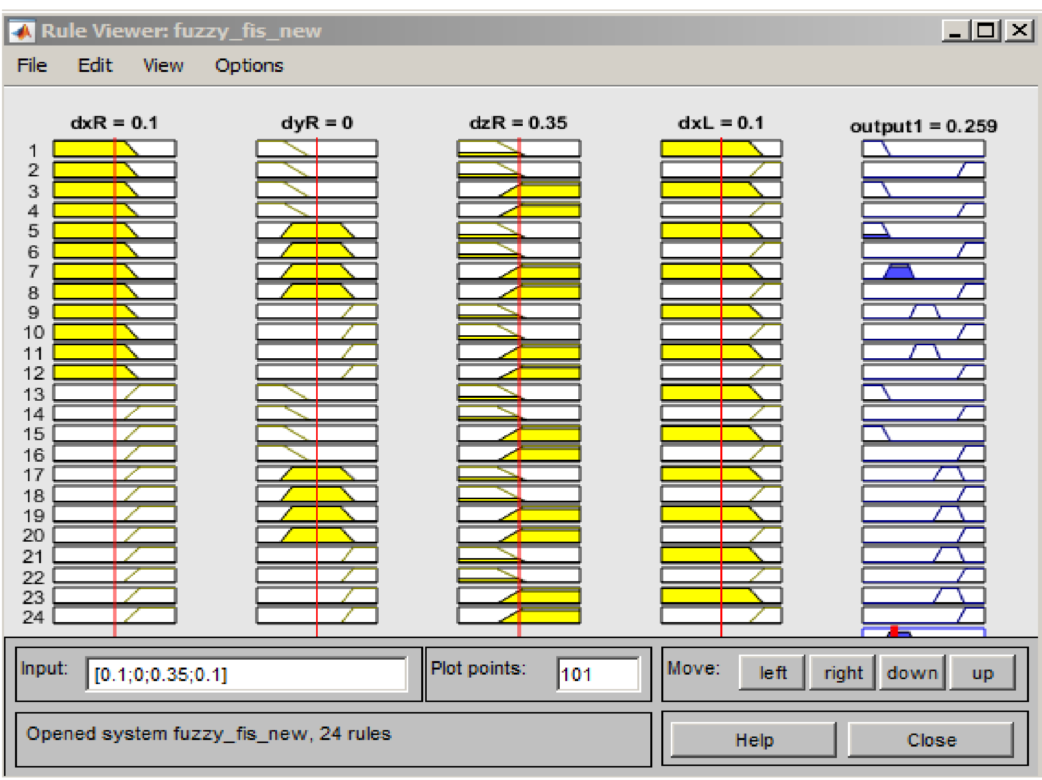

Figure 12: Rule display window of the FIS system

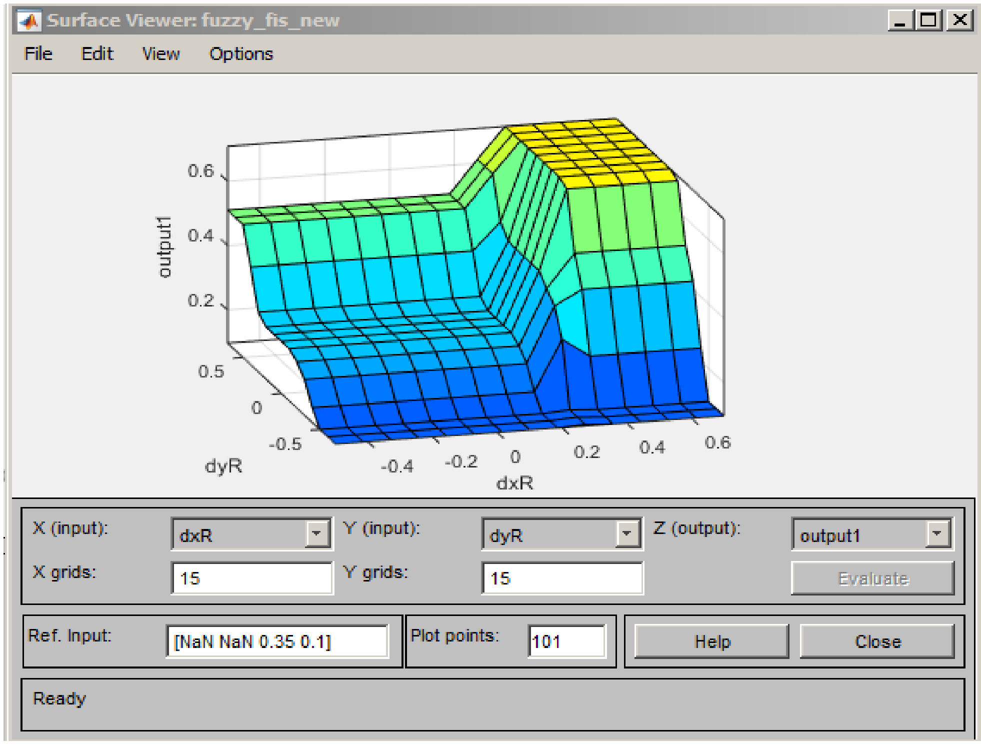

Figure 13: Surface viewer of the FIS system

3.1.3 Automation System

The system consists of a microcontroller unit and the devices to be controlled. The microcontroller unite receives the control information from the action recognition system through its serial port and turns on or off a device. Each device will turn or off as a result of a corresponding gesture assigned to it. The devices are connected to the microcontroller unit through hardware relay switches.

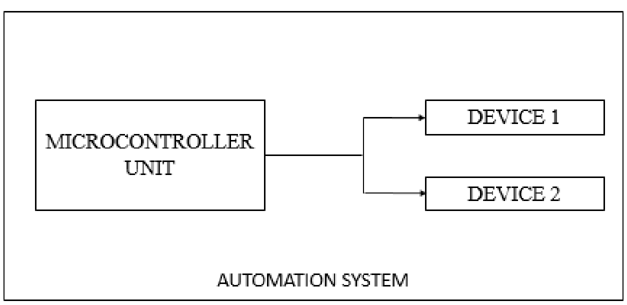

Figure 14: Block diagram of the Automation System

4. Results and Discussions

This section discusses the results obtained after the successful implementation of the proposed system. The system turns on or off a selected device upon identifying a suitable gesture. Although the Fuzzy Inference System's mathematical model has been designed to recognize four gestures, the automation system has been implemented to control only two devices. The system is programmed such that the same action is used for turning the device "on" or "off."

Case 1: No activity from the user

The figure below displays the Kinect sensor output when the system is enabled. The system is now ready to accept any gesture made by the user and recognize them.

IJCSN - International Journal of Computer Science and Network, Volume 7, Issue 3, June 2018
ISSN (Online): 2277-5420
www.IJCSN.org
Impact Factor: 1.5

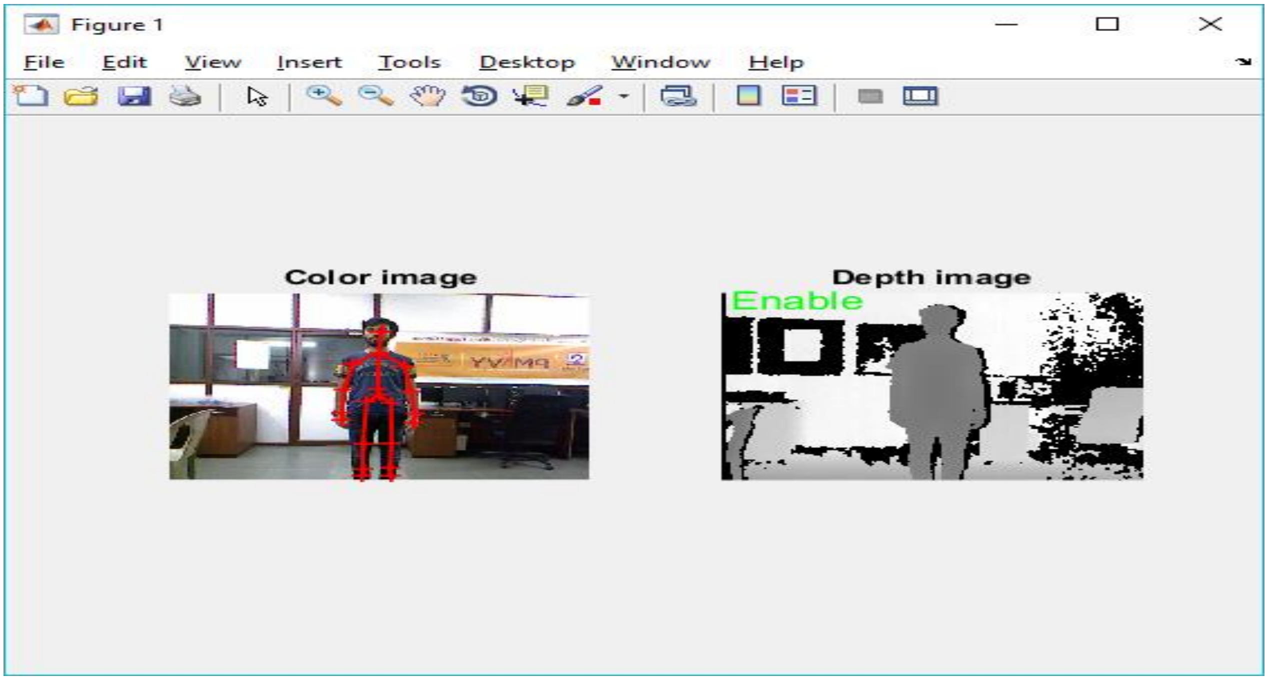

Figure 15: Kinect sensor output indicating that the system is enabled

Case 2: Left-arm signal

The user making a gesture of raising his left arm to the side is indicated in the color and depth images, as seen in figure 16.

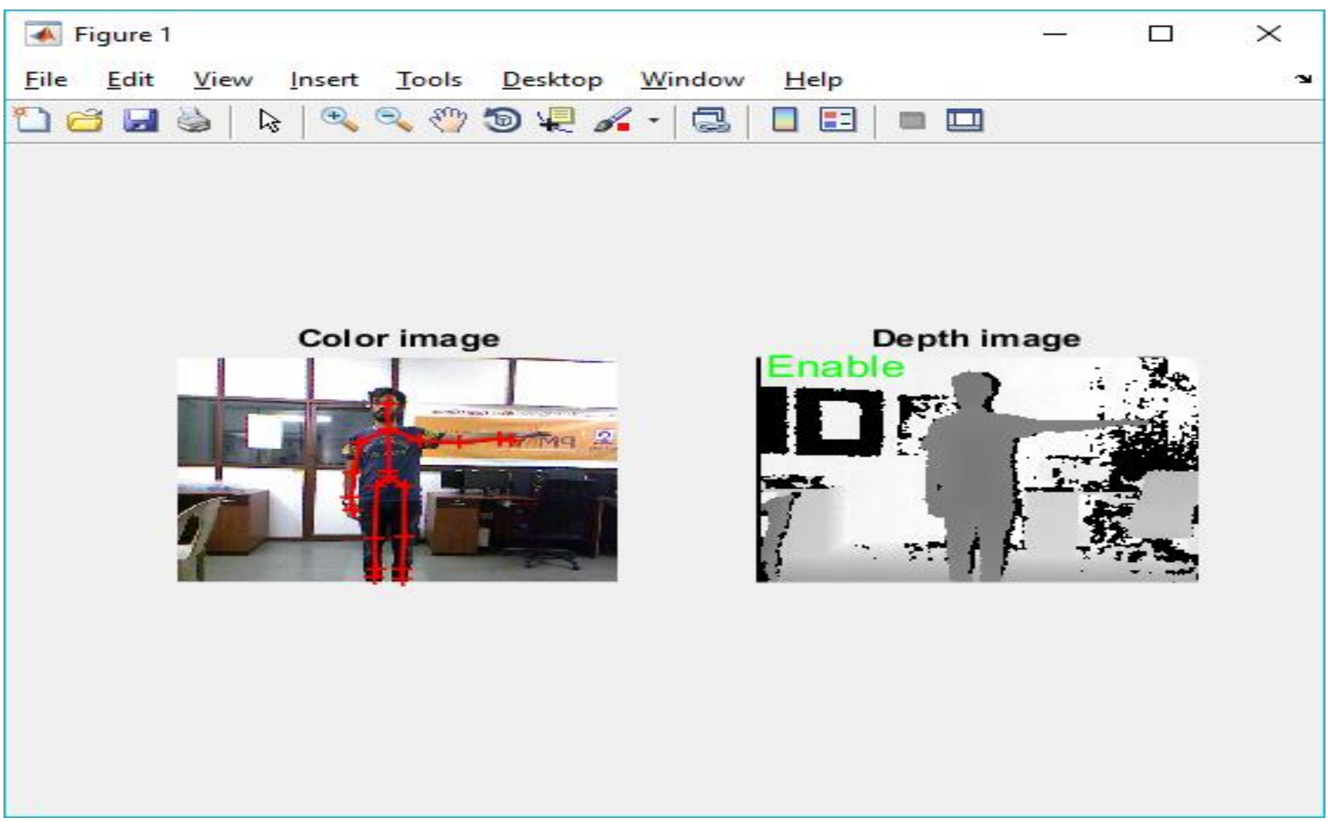

Figure 16: Kinect sensor output when the left arm is raised to the side

The system recognizes this gesture and indicates it by turning on device one. The turning "on" of device one is marked by bulb one glowing, as seen in figure 17.

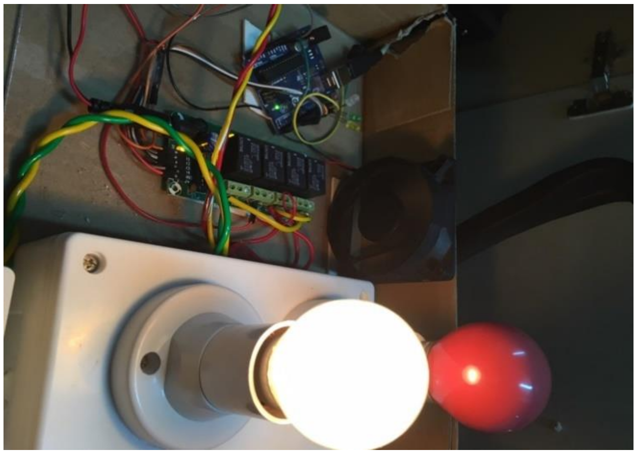

Figure 17: Snapshot of Bulb-1 turned on

Case 3: Right-arm signal

The user making a gesture of raising his right arm to the side is indicated in the color and depth images, as seen in figure 18.

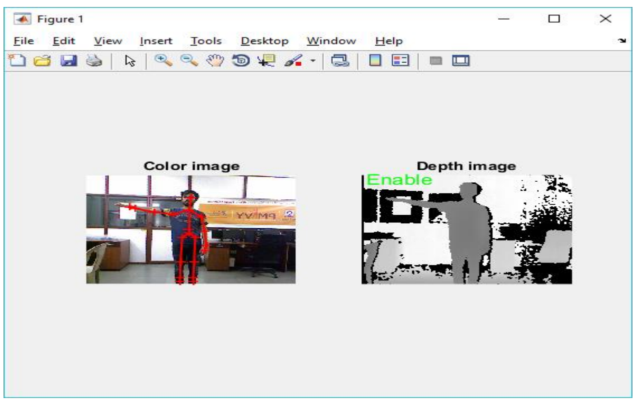

Figure 18: Kinect sensor output when the right arm is raised to the side

The system recognizes this gesture and indicates it by turning on device two. The turning "on" of device two is marked by bulb two glowing, as seen in figure 19.

Figure 19: Snapshot of Bulb-2 turned on

The Matlab window displaying the status of devices in the command window can be seen in figure 20.

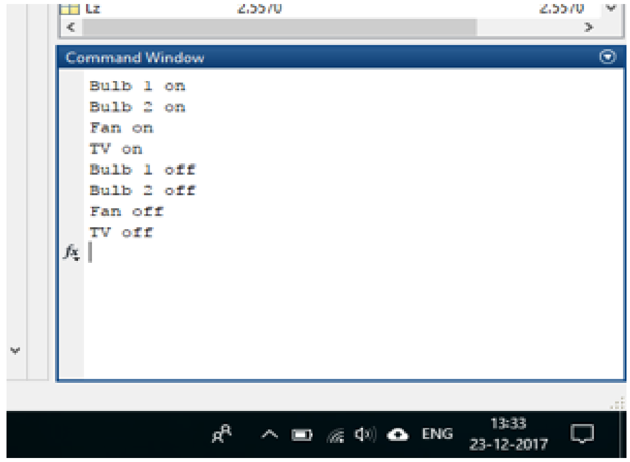

Figure 20: Matlab window displaying the status of devices

5. Conclusion and Future Work

A human action recognition system was successfully designed and implemented. The recognition system was designed based on the human skeletal features obtained using a Kinect sensor. The recognition system was developed as a software system that takes input from the Kinect sensor and delivers the detection result to an automation system. The automation system was implemented using a microcontroller unit and two devices. The devices were controlled with two different user gestures. The recognition system was successfully designed and implemented on Matlab, along with the necessary hardware and software resources. The results depicted that the devices were controlled by gestures made by the user.

The system can be updated to recognize multiple gestures in the future. The system can also be updated to work with 3-D images.

References

[1] Akila. K and Chitrakala. S, "A Comparative Analysis of Various Representations of Human Action Recognition in a Video", IJIRCCE, Vol. 2, Issue 1, January 2014.

[2] Taha, Ahmed, et al. "Human activity recognition for surveillance applications." Proceedings of the 7th International Conference on Information Technology. 2015.

[3] Ahmed Taha, Hala H. Zayed, M. E. Khalifa, and El-Sayed M. ElHorbaty, "Exploring Behavior Analysis in Video Surveillance Applications," In The International Journal of Computer

Applications (IJCA), Foundation of Computer Science, New York, USA, Volume 93, Number 14, pp. 22-32. May 2014.

[4] Jyotsna E, Akhil P V, Arun Kumar," Silhouette based human action recognition using PCA and ISOMAP", International Journal of Advanced Research in Computer and Communication Engineering Vol. 2, Issue 11, November 2013.

[5] Ahmed Taha, Hala H. Zayed, M. E. Khalifa, and El-Sayed M. ElHorbaty, "Exploring Behavior Analysis in Video Surveillance Applications," In The International Journal of Computer Applications (IJCA), Foundation of Computer Science, New York, USA, Volume 93, Number 14, pp. 22-32. May 2014.

[6] Lulu Chen, Hong Wei, James Ferryman, "A survey of human motion analysis using depth imagery," In Pattern Recognition Letters, Elsevier Science Inc., Volume 34, Issue 15, pp. 1995-2006, November 2013.

[7] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa, "Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group," In Proceedings of the International IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, Ohio, USA, pp.588-595, June 2014.

[8] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim, "A Depth Video Sensor-Based Life-Logging Human Activity Recognition System for Elderly Care in Smart Indoor Environments," In the International Journal of Sensors, Volume 14, Number 7, pp. 11735-11759, July 2014.

[9] Samuele Gasparrini, Enea Cippitelli, Susanna Spinsante, and Ennio Gambi, "A Depth-Based Fall Detection System Using a Kinect Sensor," the International Journal of Sensors, Volume 14, Issue 2, pp. 2756- 2775, February 2014.

[10] Salah Althloothia, Mohammad H. Mahoora, Xiao Zhanga, Richard M. Voylesb, "Human Activity Recognition Using Multi-Features and Multiple Kernel Learning," In Pattern Recognition Journal, Volume 47, Issue 5, pp. 1800–1812, May 2014.

[11] Jiang Wang, Zicheng Liu, Ying Wu, Junsong Yuan, "Mining Actionlet Ensemble for Action Recognition with Depth Cameras," In Proceedings of the International IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, Rhode Island, USA, pp. 1290-1297, June 2012.

[12] Ferda Ofli, Rizwan Chaudhry, Gregorij Kurillo, René Vidal, and Ruzena Bajcsy, "Sequence of the Most Informative Joints (SMIJ): A New Representation for Human Skeletal Action Recognition," In proceedings of the IEEE Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, Rhode Island, USA, PP. 8-13, June 2012.

[13] Kavita V. Bhaltilak, Harleen Kaur, Cherry Khosla, "Human Motion Analysis with the Help of Video Surveillance: A Review," In the International Journal of Computer Science Engineering and Technology (IJCSET), Volume 4, Issue 9, pp. 245-249, September 2014.

[14] Chen Change Loy, "Activity Understanding and Unusual Event Detection in Surveillance Videos," Ph.D. dissertation, Queen Mary University of London, 2010.

[15] Choudhury, Ananya, Anjan Kumar Talukdar, and Kandarpa Kumar Sarma. "A Review on Vision-Based Hand Gesture Recognition and Applications." Intelligent Applications for Heterogeneous System Modeling and Design. IGI Global, 2015. 256-281.

[16] Osunkoya, Toyin, and Johng-Chern Chern. "Gesture-based human-computer-interaction using Kinect for Windows mouse control and PowerPoint presentation." Department of Mathematics and Computer Science, Chicago State University, Chicago, IL 60628 (2013).

[17] Khan, Rafiqul Zaman, and Noor Adnan Ibraheem. "Hand gesture recognition: a literature review." International Journal of Artificial Intelligence & Applications 3.4 (2012): 161.

[18] Cai, Xingquan, et al. "Infrared Human Posture Recognition Method for Monitoring in Smart Homes Based on Hidden Markov Model." *Sustainability* 8.9 (2016): 892.

[19] David, Nathan, et al. "Design of a home automation system using Arduino." *International Journal of Scientific & Engineering Research* 6.6 (2015): 795-801.

[20] S. Imran J. Vignesh, V. Singh, Dr.T. Prasath "Smart Home Automation Based On IoT Using Arduino Mega", International Conference on Current Research in Engineering Science and Technology (ICCREST-2016).

[21] Zhang, Zhengyou. "Microsoft Kinect sensor and its effect." *IEEE Multimedia* 19.2 (2012): 4-10.

[22] AIAA OC Rocketry (Revision April 3 27, 2014 - http://aiaaocrocketry.org)

[23] Smith, Alan G. "Introduction to Arduino." *Reference book, September* 30 (2011): 1-172.

[24] Arduino Uno "A Product Review".

[25] https://www.tutorialspoint.com/fuzzy_logic/fuzzy_logic_inference_system.htm

Authors:

Sai Kailash is a final year B.E student in the Electronics and Communication Engineering (ECE) department of B.M.S College of Engineering, Bangalore.

Sai Karthik is a final year B.E student in the Electronics and Communication Engineering (ECE) department of B.M.S College of Engineering, Bangalore.

Purav M Shah is a final year B.E student in the Electronics and Communication Engineering (ECE) department of B.M.S College of Engineering, Bangalore.

B Madhukar is a final year B.E student in the Electronics and Communication Engineering (ECE) department of B.M.S College of Engineering, Bangalore.

K. Vijaya is working as an Assistant Professor in the Electronics and Communication Engineering (ECE) department of B.M.S College of Engineering, Bangalore.