

# A Real-Time Face Motion Based Approach towards Modeling Socially Assistive Wireless Robot Control with Voice Recognition

Abhinaba Bhattacharjee<sup>1</sup>

Department of Electronics and Communication Engineering  
Adamas Institute of Technology, Barasat  
West Bengal University of Technology, Kolkata –700064  
Kolkata, West Bengal, India

Partha Das<sup>2</sup>

Department of Computer Science and Engineering  
Regent Education and Research Foundation, Barrackpore  
West Bengal University of Technology, Kolkata –700064  
Kolkata, West Bengal, India

Debasish Kundu<sup>3</sup>

Gobindapur Saphali Memorial  
Polytechnic, Gobindapur,  
Guskara, Burdwan, West Bengal,  
India

Sudipta Ghosh<sup>4</sup>

Bengal Institute of Technology and  
Management, Santiniketan,  
Gopalnagar, Birbhum, West Bengal,  
India

Sauvik Das Gupta<sup>5</sup>

School of Electrical and Computer  
Engineering,  
Oklahoma State University  
Stillwater, OK, USA

**Abstract**—The robotics domain has a couple of specific general design requirements which requires the close integration of planning, sensing, control and modeling and for sure the robot must take into account the interactions between itself, its task and its environment surrounding it. Thus considering the fundamental configurations, the main motive is to design a system with user-friendly interfaces that possess the ability to control embedded robotic systems by natural means. While earlier works have focused primarily on issues such as manipulation and navigation only, this proposal presents a conceptual and intuitive approach towards man-machine interaction in order to provide a secured live biometric logical authorization to the user access, while making an intelligent interaction with the control station to navigate advanced gesture controlled wireless Robotic prototypes or mobile surveillance systems along desired directions through required displacements. The intuitions are based on tracking real-time 3-Dimensional Face Motions using skin tone segmentation and maximum area considerations of segmented face-like blobs, Or directing the system with voice commands using real-time speech recognition. The system implementation requires designing a user interface to communicate between the Control station and prototypes wirelessly, either by accessing the internet over an encrypted Wi-Fi Protected Access (WPA) via a HTML web page for communicating with face motions or with the help of natural voice commands like “Trace 5 squares”, “Trace 10 triangles”, “Move 10 meters”, etc. evaluated on an iRobot Create over Bluetooth connectivity using a Bluetooth Access Module (BAM). Such an implementation can prove to be highly effective for designing systems of elderly aid and maneuvering the physically challenged.

**Keywords**—Face detection; Skin tone segmentation; Voice Commands; Speech Recognition; Wi-Fi Protected Access (WPA); Arduino Wi-Fi Shield; iRobot Create; Surveillance systems

## I. INTRODUCTION

In today's age, the robotic industry has been developing many new trends to increase the efficiency, accessibility and accuracy of the systems in order to automate the processes involved in task completion. With the improvement of the advancing technology, humans have inherited a tendency of reducing physical and mechanical efforts to avoid repetitive jobs that are boring as well as stressful. Since long Robots have been built to reduce tedious human efforts and human errors. Though robots can be a substitute to manpower, they still need to be controlled by the humans, whether the robots be wired or wireless and thus needs to be handled by a controller device, both having pros and cons associated with them. The complexity of calculations and computations has made a breakthrough in modern technology, and human-computer interactions using natural resources has been made possible with user-friendly interfaces and machine learning algorithms. With these algorithms, system training is made effective as the system responds to simple human gestures or colors available in the environment with corresponding outputs and hence can prove efficient in controlling man-machine interactions. So beyond controlling the robotic system through physical or electronic devices, if the recent gesture control method is applied to embedded robotic systems then it provides a rich and intuitive form of interaction with the system which mainly involves Image Processing and Machine Learning algorithms for application development. Beyond this, it also requires some hardware and software interfacing with the system for gesture acquisition and corresponding control signal generation. Many attempts have been made in this field using motion sensing apparatus like that of an accelerometer with a combination of the gyroscope. These have been conventionally used in various

systems of Human-Computer Interaction for sensing and tracking hand gestures externally to direct mechanical Robots to desired directions. However, acquisition of speech signal and emulation of various system navigation events using Digital Speech Processing has also been a method of Human System Interaction. But this method did not prove to be effective as it is very much prone to the environmental noise and may result in inefficient outcomes until and unless Speech recognition is taken into account.

In reference Arce [1] et al, made an intuitive effort by carrying extra circuitry attaching a number of accelerometers on the hand to develop a hand gesture recognition technique using ANN. Hand gesture recognition using image processing algorithms many a times involve the use of colored gloves. By tracking this color glove, different hand gestures can be interpreted as described by Luigi Lamberti and Francesco Camastra in their paper [2] where they have modeled a color classifier performed by Learning Vector Quantization. Kim [3] et al. developed a pattern recognizing algorithm that has been used to study the features of the hand. Some gesture recognition systems involve adaptive color segmentation [4], hand finding and labeling with blocking, morphological filtering, and then gestures are found by template matching. These processes do not provide dynamicity for the gesture inputs. While in another approach, gestures are recognized using Microsoft Xbox 360 Kinect(C)[5]. Kinect gathers the color and depth information using an RGB and Infra-Red lens respectively. There are many papers where training of hands using a large database of near about 5000-10000 positive and negative images is considered. But this hand gesture recognition technique doesn't provide an identical real-time biometric logical authorization to access control of surveillance systems, which made us shift the Gesture acquisition domain from Hand tracking to real-time face tracking as well as the acquisition of definite voice commands using digital speech recognition algorithm.

In our approach we have developed an intelligent user interface which involves two distinct and intuitive forms of interaction – either by acquiring real-time three-dimensional face motions by real-time face tracking algorithm, or by definite voice commands using digital speech recognition algorithm. Both the processes provides a unique logical authorization of the user to access a control station.

The system implementation involves the design of an advanced User Interface to access a control station to control an embedded wireless robotic prototype wirelessly, either using a Bluetooth access module or an encrypted Wi-Fi Network using a Wi-Fi shield with the help of either face gesture acquisition or specific voice commands. We have tested our algorithms on two different embedded platforms. The Speech recognition algorithm is implemented on a ready-made platform of *iRobot Create* using a wide range Bluetooth Access Module, while the real-time face detection and tracking algorithm has been evaluated on a self-developed embedded prototype built on an open source AVR microcontroller-based platform (*ARDUINO*) with an Arduino Wi-Fi shield to connect to a protected Wi-Fi network and control its navigation over the web via an HTML Web page. The next Section describes the Interfaces involved in the

system design with Section (III) elucidating the Hardware Platform of the system followed by Section (IV) describing the methodology to design the overall system implementation. After that comes the Experimental Evaluations and Results in Section (V) with finally the Conclusion and the Future Works of the paper.

## II. INTERFACES

Interfacing is an integral part of every embedded electro-mechanical systems controlled by either automation or Human Computer Interaction. Our work is mainly composed of two fundamental interfaces namely *User Interface* and *Wireless Interface*.

### A. User Interface

Non-invasive techniques for controlling are in high pace with the advancement of technology. Many works have already been done by Computer Vision experts that include Augmented Reality, Controlling PC-Mouse events through color gestures [6] or hand gestures that include selecting, opening, closing files. Both mouse and keyboard can be replaced by virtual keyboard and mouse described by Tsang, W.-W.M [7] which reduces hardware components of a PC.

In our approach the user interface involves the interaction between the User and the Control station through face gestures or voice commands. Here we have used a Personal Computer (PC) to serve the purpose of control station for accessing control over the robotic prototype.

In case of control through real-time face detection and tracking, the PC webcam is used to collect data in the form of images in a live video stream. This data is further processed and computed by a real-time face detection and tracking algorithm using image processing [8] and computer vision algorithms to provide the prototype the exact control signal of what the user wants to direct through various face gestures.

While in the case of control through definite voice commands, the inbuilt PC microphone is accessed to acquire the speech signals in the form of sinusoidal waveforms, process them using speech recognition algorithm and generate control signals to direct the prototype in a specific direction according to the delivered voice command.

### B. Wireless Interface

This includes the wireless connectivity of the embedded robotic prototype with the control station either by Bluetooth connectivity or through internet over a protected Wi-Fi network. In our work we have used a Bluetooth Access Module to connect the control station i.e. the PC to the readymade robot of *iRobot Create* and control it through Voice commands, while we have used a wireless Wi-Fi network for wireless hardware to hardware interfacing between the control station and the prototype developed on Arduino based platform.

For the second case both the prototype and the control stations are connected to the same Wi-Fi network and control signals generated through tracked real-time face gestures are transferred in TCP/IP in the form of data packets via a HTML

web page over a secured Wi-Fi Protected Access encryption mode.

### III. HARDWARE PLATFORM & SYSTEM REQUIREMENTS

The system design of our project work involves a synchronization of hardware and compatible software integration for its effective working with satisfying results. Tested on both a readymade platform as well as an open source platform, the system includes the following specifications.

#### A. Control Station

In our work we have used a Personal Computer (PC) for voice as well as real-time face gesture acquisition to work as

our control station with the following minimum hardware requirements:-

- i. Processor Clock speed should be at least 2 GHz (Intel Core2Duo or Higher versions )
- ii. RAM (Primary Memory) should be not less than 2GB.
- iii. The OS (Operating System) should be compatible to support Arduino IDE 1.0.2v and higher versions of MATLAB (Windows7 onwards, Linux, MAC).
- iv. A Digital Megapixel Front Web camera is needed to detect and track Faces.
- v. An inbuilt microphone to acquire voice commands for processing speech recognition.



Fig. 1. iRobot Create

Fig. 2. The Hardware setup for the overall implementation of Arduino based Self-developed Robot

#### B. Prototype Platforms

The design implementation has been evaluated and simulated in two different microcontroller based platforms in order to check Mechanical Robustness, performance levels, flexibility and feasibility of the system.

##### ❖ iROBOT – CREATE® (Readymade Hobbyist Robot )

This is a total set of robot development kit with a complete mechanical assembly developed by iRobot® based on the iRobot's Roomba Vacuum cleaning platform. It is rather known as a hobbyist robot with which one can develop new Robotic behaviours without worrying about the low level control. The hardware specifications include:-

- i. *Cargo Bay* – which houses external electronic hardware like robotic arm or, sensors and actuators for external attachment.
- ii. *DB-25 Port* – for providing serial communication to attached peripheral electronics.
- iii. *7 Pin Mini DIN Connector* – A serial port through which sensor data can be read and motor commands can be issued using iRobot Roomba's Open Interface (ROI) protocol.
- iv. An Omnidirectional IR Receiver – for obstacle detection or wall sensing.
- v. Bluetooth Access Module (BAM) – This is a high range Bluetooth connectivity device manufactured by Element

Direct. This hosts an individual client within an optimum range.

In addition to these features the platform accepts virtually all accessories designed for iRobot's second generation Roomba 400 Series domestic robots and can also be programmed with iRobot's own Command Module (a microcontroller with a USB connector and four DE-9 expansion ports).

The present day models replaced iRobot Roomba's Open Interface (ROI) and has provided Create's own Software interface that allows Create's behaviour manipulation through a series of commands including mode commands, actuator commands, demo commands, and sensor commands that is sent to Create's serial port by way of a PC connected to the DB-25 port using a BAM.

##### ❖ ARDUINO based (Self-Developed Robot )

This is a Prototype built on an Open Source Microcontroller based platform (Arduino UNO) and especially designed to provide a facility of Navigational control through the Wi-Fi based web access using internet protocol.

The Hardware requirements involve:-

- i. *ARDUINO Uno R3* – which primarily consists of ATMEL AtMEGA 328p which is a 28 pin 8-bit AVR-RISC based microcontroller operating at a clock speed of 20 MHz.
- ii. *Arduino Wi-Fi Shield (Compatible to Arduino Boards)* – Allows Arduino boards to connect to the internet

- wirelessly through IEEE 802.11b/g/n system in package using both the TCP and UDP protocols.
- iii. *L293D* – Motor Driver circuit, a 16 pin H-bridge IC to provide sufficient power output to the DC motors.
- iv. *DC Motors (300RPM, 12V)* – a pair of geared DC motors providing a maximum of 300RPM at 12 Volts.
- v. *Servo Motor (Hitec HS-485HB)* – a 3pole top ball bearing heavy duty high performance 180 degree servo motor providing a maximum torque of 6.4 Kg/cm at 6 Volts.
- vi. *External Power Bank (Ambrane 13000mAh)* – To provide required power to all the circuitry operating at 5 Volts.
- vii. *9V Batteries* – a set of 3 batteries in series is need to provide an uninterrupted 12 Volts potential difference to the pair of geared DC motors.

- viii. *Network Camera (DLink DCS-933L)* – This is a Wi-Fi supported Day/Night surveillance camera with IR night vision which acquires live video footage of the prototype end and communicates it to the control station end remotely.

The primary motive for such a design is to develop an interactive embedded robot operated over the web through Wi-Fi, with a surveillance camera for live prototype feed, which can serve the purpose of various industrial applications as well as be a promising support for the physically challenged or elderly society.

#### IV. METHODOLOGY

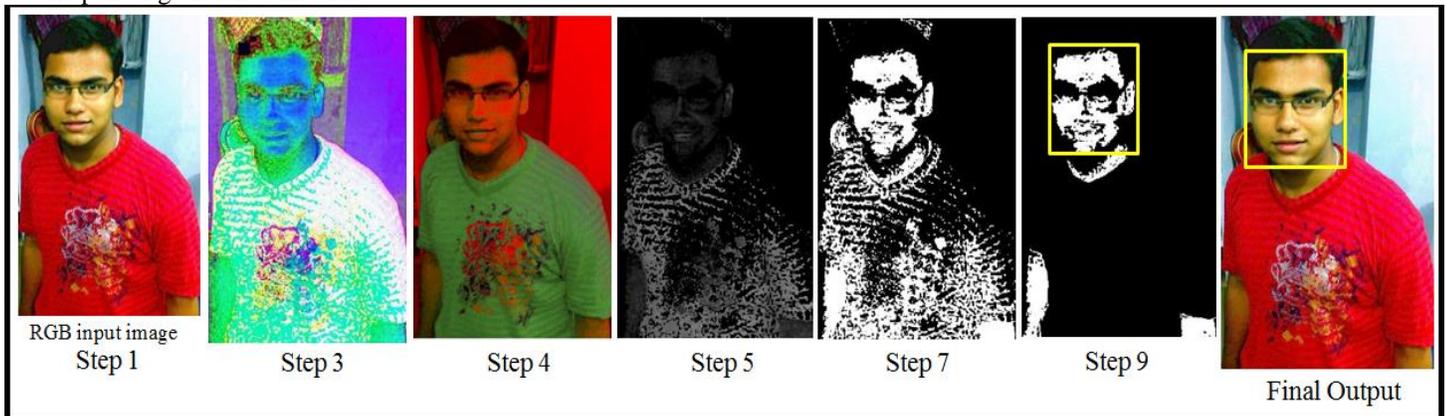


Fig. 3. Pictorial representation of the Steps of Real-time Face Detection and Tracking Algorithm

This is a Software and Hardware based Human System Interaction approach to develop an interactive wireless Robotic interface over Bluetooth connectivity or over Wi-Fi to generate specific industrial applications based on surveillance systems with live biometric authorisation, advanced vehicular facilities, support for the physically challenged and elderly aid.

The User interface involves the Human-Computer Interaction between the User and the Control station which has been accomplished in two different simulation modes – First is with the help of Voice command control, by speech recognition, while the Second is with the help of real-time face detection and tracking by skin tone extraction and analysis. Both the modes have been evaluated in two different wireless embedded platforms individually.

##### A. System Description

In every Human Computer interaction system Gesture acquisition, analysis and corresponding control signal generation for proper functional execution of the system, forms the fundamental steps of such an implementation. Likewise in our proposed system we have divided the implementation into two different parts. Based on their simulations and evaluated platforms, they are:-

- a) *Real-time Face gesture controlled Arduino based self-developed Wi-Fi Surveillance Robot.*
- b) *Voice command controlled iRobot Create®.*

This section will elucidate the details of both the simulations with their respective platforms.

##### B. Simulation of Arduino based Self Developed Robot

The design basically focuses on real time face detection and tracking to provide a live biometric authorization to the system. The movements of the detected face is acquired as face gestures corresponding to which different control commands are generated at the control station, which in turn are simultaneously communicated to the embedded prototype through Wi-Fi enabled web access, provided the prototype is within the respective Wi-Fi zone and as a result facilitates the different navigational actions of the embedded prototype using the acquired face gestures. The steps carried out in the design are sequentially illustrated as follows:-

###### 1) Real-time Face Detection

Human face is an identity of an individual which acts as a distinguished and deterministic information for security passwords, database search engines and biometric pattern recognitions. Many approaches have already been made in the field of human face detection for both standalone image frames as well as live video streams since the last decade and presently it has become a major field of interest in current research and technology.

Face detection has achieved moderate detection rates for standalone image frames using complex image processing algorithms which involves color pixel thresholding and normalisation in different color spaces [9], edge based human

face detection [10], Skin pixel clustering and quantized skin color regions merging using wavelet packet analysis[11][12], face detection algorithm using eigen image and template matching. These algorithms require huge computational complexity and hence cannot be used for real time processing unless a GPU is involved within the hardware.

In recent trends computer vision based machine learning algorithms proved quite efficient for real-time face detection and tracking among which Viola-Jones algorithm [13] is one of the most popular vision based algorithm for real-time face feature detection as it uses integral imaging and cascade classifiers based on rectangular HAAR-like features to detect the frontal framework of a possible human face like region. However another possible way of detecting and recognising faces would be by mimicking the working principle of human brain with the help of pattern recognition by machine learning algorithms based on training neural networks with supervised learning processes which requires a huge number of training data sets.

Inspired by these works we attempt to come up with a technique which can detect faces with slight tilts and rotations in real-time. Performing experimental trials with several color spaces to remove the maximum number of the non-face pixels, in order to narrow the focus to the remaining predominant skin colored regions [14], we found that the HSV color space fits the best. Since a given picture can have variations of the light incident on it, we had to first make sure that we could cancel the distortions caused by these variations. After going through various color spaces and vision based techniques to reject false positives in real time processing, we decided to go for Skin tone segmentation with a hybrid color space formed by subtracting the Hue channel (H) from the Chrominance channel (I) obtained from the HSV & NTSC color spaces, respectively [15], which results into a skin tone segmented image represented by a 2 Dimensional Matrix. This is then converted to a binary image by grayscale thresholding followed by noise removal and various other morphological processes like dilation and erosion. In order to further classify faces, we implemented a geometric analysis step which would discard any skin tone segmented region that doesn't fall under the shape of a face and thus reduce the number of false positives. Faces are generally elliptical or rather oval in shape. With this assumption we have filtered the skin segmented regions with an eccentricity filter along with a thresholding in the elliptical aspect ratio i.e. length of major axis by minor axis ratio of each skin segmented blob. The resultant skin segmented blob, with eccentricity ranging from 0.5 to 0.7 and elliptical aspect ratio ranging from 1 to 2, is demarcated as a face like blob which is marked with a rectangular bounding box. This whole process continues iteratively in a loop for successive frames of the webcam acquired live video stream to produce effective real-time face detection.

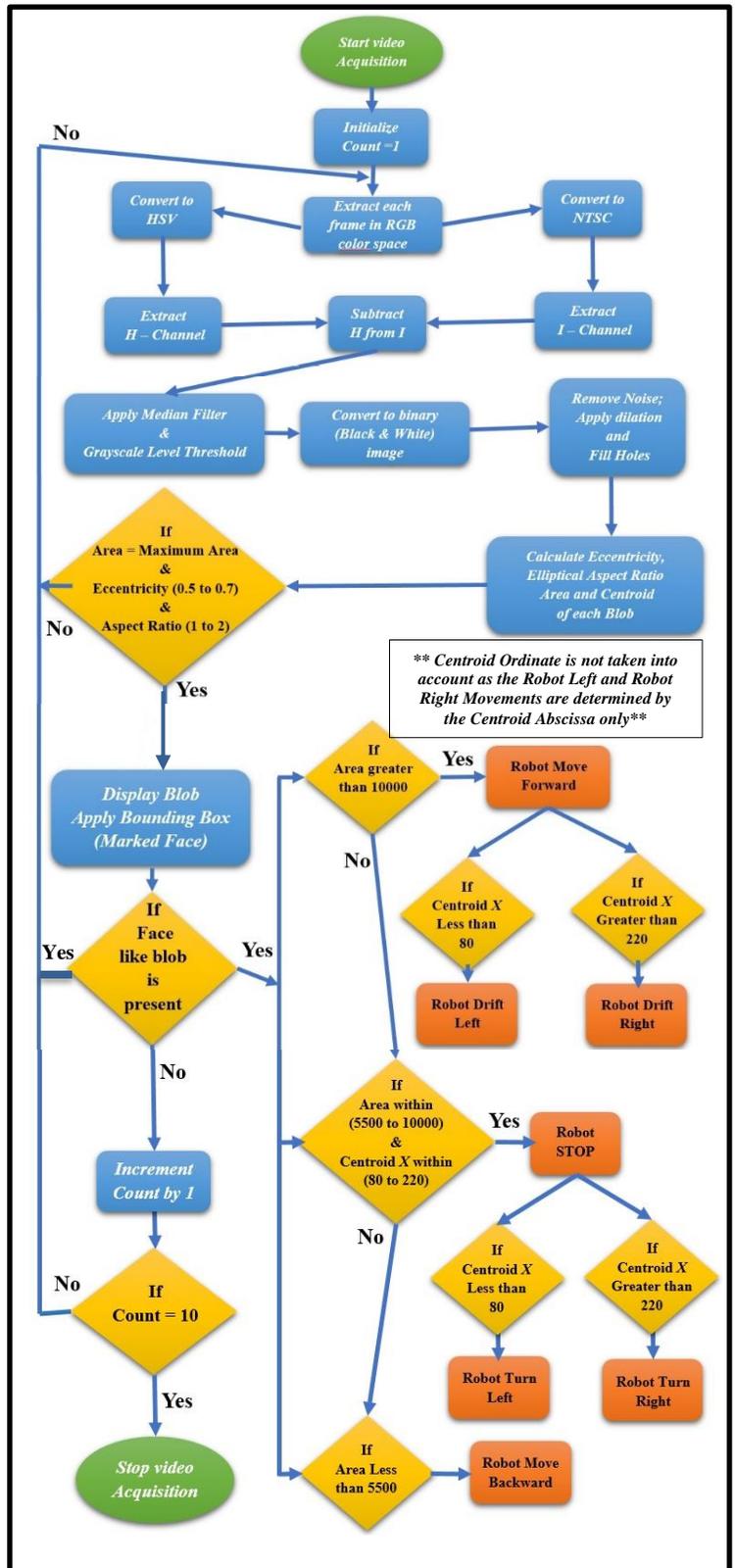


Fig. 4. Complete Flowchart of the developed Algorithm

## 2) Real-time Face tracking

This is a computer vision based technology that determines locating faces and their sizes in an arbitrary image and tracks for the face coordinate vectors of successive image frames in a live video stream to achieve a real time live face detection and tracking system. Similarly in our approach the algorithm takes in an RGB image and after performing various steps of skin tone segmentation [16] as discussed above followed by several morphological and filtering processes, it outputs the same RGB image with the largest possible face-like Blob surrounded by a rectangular bounding box[17] as the detected face. When these processes continue iteratively in a loop, the system tracks the user's face by locating the coordinates of the face-like blob within the preview. The rectangular bounding box surrounding the detected face-like blob demarcates the area, instantaneous position vectors and centroid of the blob. The rectangular bounding box moves accordingly along with the movement of the faces in the live preview. The algorithm is developed such that it tracks for the face like blob having the largest area in the preview while discarding other face like blobs in the background. This feature tracks only the user's face regardless of other faces present at the back and hence tracks for the instantaneous face movements of the user within the preview for corresponding Navigation and Control of the Prototype.

3) The complete face tracking algorithm is given as follows:-

**Step 1:** Start the video acquisition in default RGB color frame.

**Step 2:** While frames are being captured initialize Count to unity and do Steps 3 to 10.

**Step 3:** Convert captured RGB frame to HSV color plane and Extract the H channel.

**Step 4:** Convert captured RGB frame to NTSC color plane and Extract the I Channel.

**Step 5:** Subtract HUE (H) channel of HSV from Chroma Intensity (I) channel of NTSC.

**Step 6:** Apply median filter & grayscale level threshold to the resultant image.

**Step 7:** Convert the resultant image to binary black and white image.

**Step 8:** Remove noise, apply dilation and calculate the eccentricity and elliptical aspect ratio of major blob.

**Step 9:** If eccentricity ranges from 0.5 to 0.7 and elliptical aspect ratio ranges from 1 to 2 then mark the Blob as Face.

**Step 10:** Calculate the Area, Location & Centroid of Face-Like Blob within the image.

**Step 11:** If the Parameters of Step 10 are nonzero then repeat Steps 3 to 10, else increment Count by 1.

\*\*A cache of 10 iterations is created which checks for the presence of the face like blob within the preview\*\*

**Step 12:** If Count is not equal to 10 then do Step 11, else jump out of loop.

**Step 13:** Stop video acquisition.

## 4) Face Gesture Interpretation and Control

Since the last decade, there have been remarkable advancements in the acquisition of human gestures and their interpretation with the help of computational and mathematical algorithms where any bodily expression or motion is detected by the system and execution takes place accordingly.

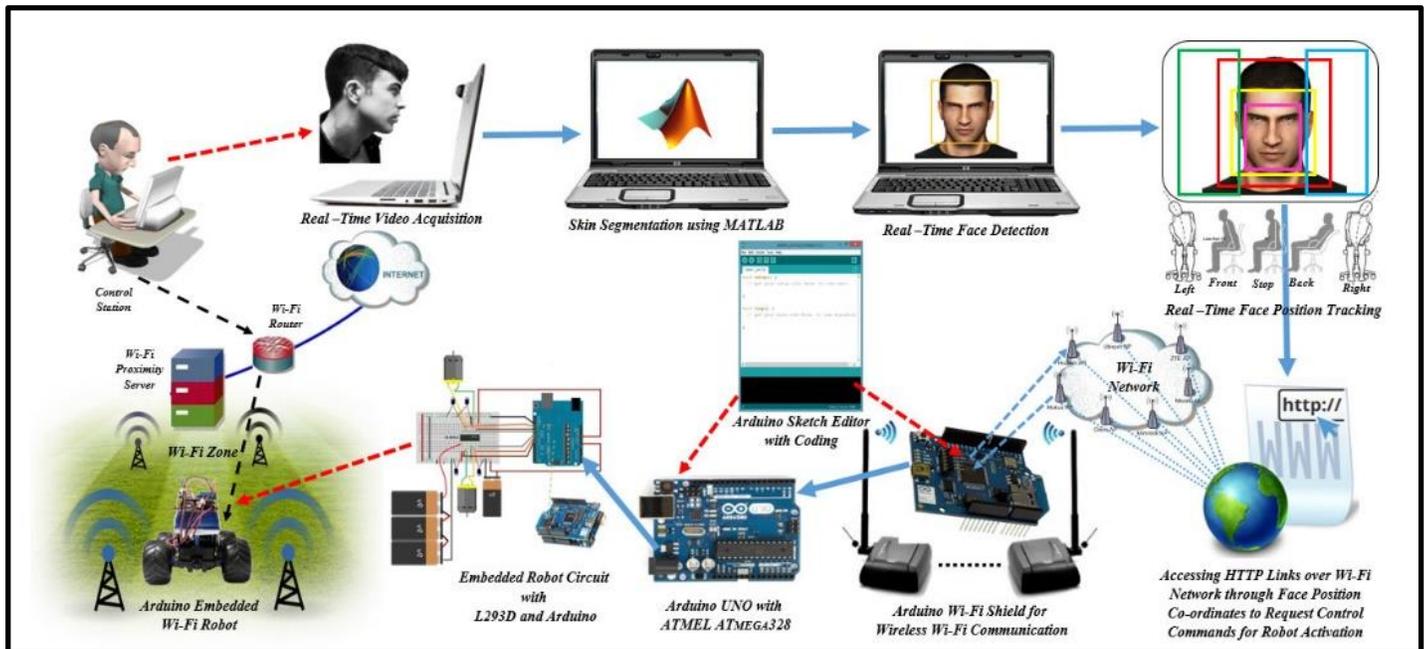


Fig. 5. The overall Schematic diagram of the total System Design and Control

Our approach is to develop a hardware and software based integrated system where there is a scope of man-machine interaction remotely without any direct mechanical or hardware interference with the user. The Real-time face tracking algorithm developed is used to detect and locate the user's face in the preview and according to the instantaneous face movements, the algorithm navigates our self-developed embedded wireless prototype in desired directions. The instantaneous face positions are located by recording the coordinates of the rectangular bounding box surrounding the detected face. These coordinates provide a centroid of the rectangular bounding box. The difference in area and centroid of the segmented major face-like blob is used to track the instantaneous facial movements or face gestures for navigation and corresponding control signal generation.

The Navigational Controls to be covered by the Prototype are movements in Forward and Backward directions along with turning Left and Right, both while in motion as well as when at rest. The halt feature, to bring the prototype from motion to rest, is a fundamental control that must be introduced to stop the movements of the prototype. The gestures corresponding to these navigational controls are tracking face movements along the 3-Dimensional Space. The Back and Forth motion of the Face along the Z-axis corresponds to the backward and forward movements of the embedded wireless prototype respectively, considering the PC screen to be the X-Y plane. This back and forth motion of the 2-Dimensional face-like blob is tracked by the difference in area calculations. As the face moves back the area decreases while it increases when the face moves front towards the Webcam. Hence the difference in area (square pixel units) is recorded to simulate the backward and forward motion of the prototype. Tilting left along the negative X-axis and tilting right along the positive X-axis executes the Left turn and Right turn actions of the prototype respectively, while at rest. Similarly we have simulated the turning effects of left and right turns during forward motion by two conditions. First the user must make a forward motion gesture for area increment and then simultaneously tilt left or right of the Y-axis to evaluate the turning effect of left and right turns respectively, while in forward motion. Here the difference in centroid co-ordinates are counted to track the left turn and right turn actions of the Prototype, while the Robot Halt action is marked by the rest position of the User without any movements, when the centroid coordinates of the face like blob merges the line  $X=0$ . In this way the navigational controls of the prototype are interpreted at the control station with the help of various face movements.

### 5) Arduino Based Hardware Platform

The hardware used for prototyping is an open source electronics platform based on a microcontroller board *Arduino UNO R3*, which is designed around a 8-bit Atmel AVR microcontroller ATmega328p (a 28 pin microcontroller) with six analog input pins and 14 digital I/O pins (of which 6 pins can support PWM outputs) along with 6 power pins in addition to it, SDA, SCL, IOREF and AREF pins are also present. The Board features a USB interface together with a DC power jack, an ICSP header and a reset button. Operating at 16 MHz clock speed and 5V input voltage, it consists of

32KB flash memory, 2KB SRAM, 1KB EEPROM and can accommodate various extension boards. It also provides a programming interface and an Arduino Integrated Development Environment (IDE) where programs are written in embedded C or C++ and facilitates a software library which is capable of compiling and uploading programs to the board with a single click. This board uses UART TTL (5V) serial communication to interact with other external peripherals either in the form of USB through Virtual COM Port or through I<sup>2</sup>C using the Wire Library supported by the Arduino IDE.

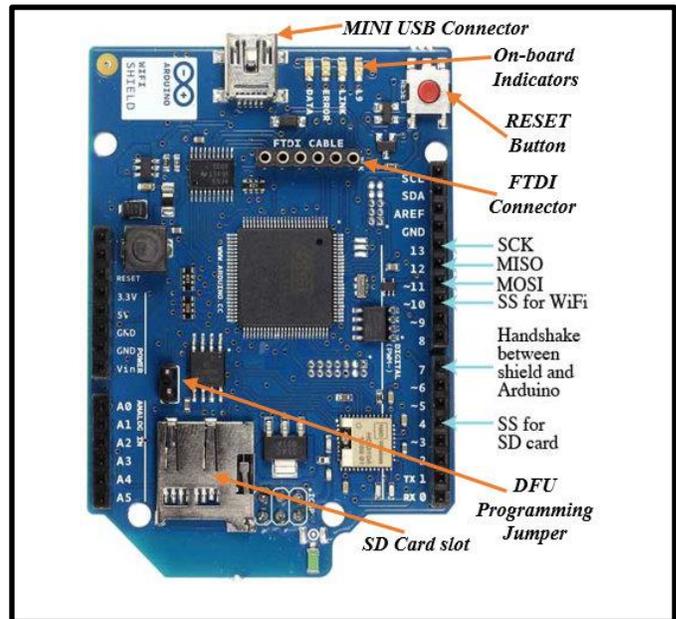


Fig. 6. Hardware Configuration of Arduino Wi-Fi shield

The Arduino Uno R3 board doesn't have any provision to support wireless communication itself or directly get connected to the internet, unless any wireless communication supporting peripheral or any internet accessing device is externally attached and programmed to it.

So in order to connect to the internet wirelessly over Wi-Fi connectivity we have used the *Arduino Wi-Fi Shield* which is compatible to most of the Arduino Boards. Based on the HDG204 Wireless LAN 802.11b/g/n System in-Package, it features a FTDI connector for serial debugging, an SD card slot for downloading/uploading and storing data, a mini USB connector for updating firmware and an AT32UC3 on board chip which provides a network (IP) stack capable of both TCP and UDP data communications through Wi-Fi connectivity. Physically it consists of 30 female pins and six ICSP header pins with long wire wrap headers extending through each of the pins to fit exactly on top of the Arduino Uno board. Communication between Arduino Uno and Arduino Wi-Fi Shield is over SPI bus using Arduino's digital pins 10-13 (CS, MOSI, MISO, and SCLK). While Pin 7 is used for handshaking between the board and the Shield, pin 4 is reserved for SD card storage and hence these pins (4, 7, 10, 11, 12, 13) of the shield cannot be used as Digital I/O pins. The shield can be programmed using the Wi-Fi Library *Wifi.h()* supported by the Arduino IDE and requires the

broadcasted SSID name to be connected to Open (unencrypted) networks, together with the Network Password to be connected to WPA personal encrypted networks, while it needs the key index along with the SSID to connect to WEP encrypted networks.

#### 6) Communication between Control Station and Prototype

The Control station communicates with the Prototype wirelessly by data communication accessing the internet over a WPA2 (Wi-Fi Protected Access 2) personal encrypted Wi-Fi network. To access internet connectivity both the prototype and the control station must be connected to the same network using its broadcasted SSID and secret Password. The prototype containing the Wi-Fi Shield can serve as either a server accepting incoming connections or a client making outgoing ones. In our approach we have programmed the shield in such a way so that it acts as a virtual Server accepting request from hosts connected to the same network through the default HTTP server Port 80.

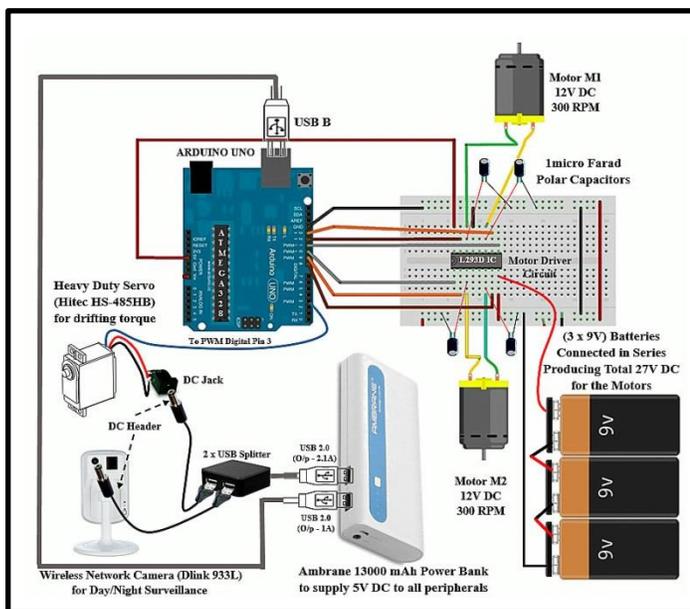


Fig. 7. Total Embedded Circuitry of the Arduino Based Self-developed Robot

In order to connect to a particular encrypted network, the shield first searches for the network in range, when found within range, it sends request to the network for establishing a connection, if all the required network parameters written in the program matches then the network acknowledges the Shield and establishes a connection by providing a local IP address to the shield within the network gateway with which exchange of data takes place between hosts and the shield through the default port 80. An HTML web page is developed at the Shield's URL i.e. the Local IP address assigned to the Shield by the Network. Once gestures are acquired at the control station for a particular navigational control, the control station sends packet data through TCP or UDP in the form of files containing HTTP headers that hits the corresponding link at the HTML webpage with a tagged word written in the code. The HTML web page consists of Hyper Text referenced web links each with a tagged word for all prototype navigational

controls that sends control signal files having HTTP headers which request the Wi-Fi shield and the Arduino Uno to activate the prototype via a dedicated motor driving circuit.

#### 7) Prototype Circuit design and Working Principle

The hardware design implementation of the prototype includes a motor driver circuit, connected to the microcontroller unit (MCU), that controls the pair of 12V geared DC motors attached to the rear wheels where the Microcontroller Unit sends control signals for forward and backward motions respectively, while the turning effects in left and right directions for both static turnings as well as forward motion drifting effects are generated by a heavy duty servo motor attached to the front wheel.

Our self-developed Prototype is a three wheeled Wi-Fi Robot built on an open source Arduino based embedded platform. Since the MCU outputs 5V DC at an average of 40-60mA as the digital output, it becomes insufficient to drive a 12V geared DC motor at such a power rating. Thus in order to overcome this we need to design a Motor Driver circuit using a 16 pin DIP (Dual in line package) H-bridge IC L293D.

The L293D is a motor driver IC which can control two low power DC motors simultaneously, to drive on either direction by just using 4 digital I/O pins. This is a symmetrical IC operating at an average of 5V DC with 4 output pins (3, 7, 11, 14) two on either side, directly attached to each DC motors with an amplified output current rating of 600mA each along with 4 input pins (2, 7, 10, 15) two on either side which are directly connected to the MCU to control the clockwise and counter-clockwise motions of the DC motors by reversing the current directions. The Chip also has a provision of providing an external supply voltage up to 36V DC from which the motors can draw the required amount of voltage to achieve their full RPM ratings. There are two enable pins (1 and 9) on either half of the chip which are responsible for regulating the voltage levels on respective halves and hence can be accessed for speed control of the DC motors attached on either sides.

The Speed Control Mechanism is very essential for special cases where the Motor needs to rotate at minimized or rather variable speeds with respect to its maximum RPM rating at necessary instances. Therefore we have included this feature of speed control mechanism in order to get the effect of geared accelerations of forward motion in our prototype. The implementation of the speed control mechanism is performed by calibrating the maximum RPM ratings of a DC motor into 256 discrete levels from 0 -255. This can be achieved by connecting the enable pins of each half of L293D to two PWM digital output pins of the MCU. Each PWM pin of the MCU by default divides its output voltage range into 8-bits i.e. 256 discrete levels and this calibration can be accessed only with the help of the function *analogWrite()* provided by the Arduino IDE. In our design we have experimentally selected 3 discrete levels out of 256 levels in order to visualize the acceleration effect of the prototype. Hence the speed control mechanism of accelerated forward motion is achieved by calling three user defined functions written in the code of the MCU namely:-

- ✓ *MoveForward()* – the prototype moves at a minimized speed calibrated at the level 150.

- ✓ *MoveForwardFaster()* – the prototype moves at a moderate speed calibrated at the level 200.
- ✓ *MoveForwardSuperFaster()* – the prototype moves at the full RPM rating and highest speed calibrated at highest level 255.

The Gestures corresponding to these speed control levels are similar to the forward motion gestures with a specific increment in area consideration for each level. The prototype moves forward if the blob area is 30% more than default, moves forward faster for 60% more than default, while it moves at its highest rating if the area is 90% more than default. The default area is the area of the face blob of the user at rest position of the robot.

However the physical design of our prototype mainly consists of two rear wheels, each driven by a 12V geared DC motor of 300 RPM rating, with relatively larger radius as compared to the front wheel which is attached to a heavy duty 180 degree servo motor. The rear wheels drives the prototype forward and backward with the servo motor being fixed at 90 degrees which is the default position of the front wheel. The rear wheels are also responsible for clockwise (Right) and anti-clockwise (left) motions i.e. the static right and left movements of the prototype with the servo motor being fixed at 45 degrees and 135 degrees respectively. Connected to a digital PWM pin of the MCU, the main objective to include a servo motor within the prototype is to generate the forward motion turning effects and facilitate the prototype to move along a zigzag path by the user tilting left and right alternatively while performing the forward motion gesture at the control station end. These gestures are tracked by the developed algorithm and the corresponding control signals containing HTTP headers are generated and sent to the Wi-Fi shield in the form of data packets transferred through TCP or UDP over the web via encrypted WPA2 Wi-Fi protocol. The Wi-Fi shield then instructs the Arduino Uno to activate the digital output pins corresponding to the control signal as written in the code and set them HIGH. The digital signals are

passed over to the corresponding input pins of the Motor driver IC L293D and the corresponding output pins attached to the DC motors get activated respectively to perform the requested navigational function. The DC motors are externally supplied with a series of 3×9V DC batteries which provide a maximum of 27V DC which is sufficient to drive a pair of 12V geared DC motors at their highest RPM ratings. A 5V DC power bank of 13000mAh is added to the physical setup of the Prototype in order to provide uninterrupted power supply to the MCU, the servo motor and a Surveillance network camera which has been added to the periphery of the prototype in the end. The Surveillance camera is a Wi-Fi based wireless network IP camera, having a facility of infrared Night vision, which accesses the Wi-Fi network of the control station and streams a real-time live feedback of the prototype at the camera URL i.e. the local IP assigned to the network camera by the Wi-Fi network. The user can view the live feed of the camera by logging in to the camera URL with default username and password provided by the manufacturer.

### C. Evaluation of Voice Command Controlled iRobot Create®

Speech is one of the oldest and most efficient form of communication that has been used by humans to communicate their feelings. It has been constantly evolving over the time and has risen to become the primary method of communication for us. Hence, a robust speech recognition or automatic speech recognition system is an essential part of any Human-Computer Interactive systems. An embedded speech recognition system, ideally, should be able to isolate background noise and be able to segment and recognize any words spoken to it, be it a single word or a complete sentence. Once recognized, subroutines, based on the recognized words are to be executed that can be used to accomplish various tasks. As can be seen, the system needs to be asynchronous and independent of the other systems. Since the actions are conditional based on only specific voice commands, a speech recognition system, is often without a User Interface.

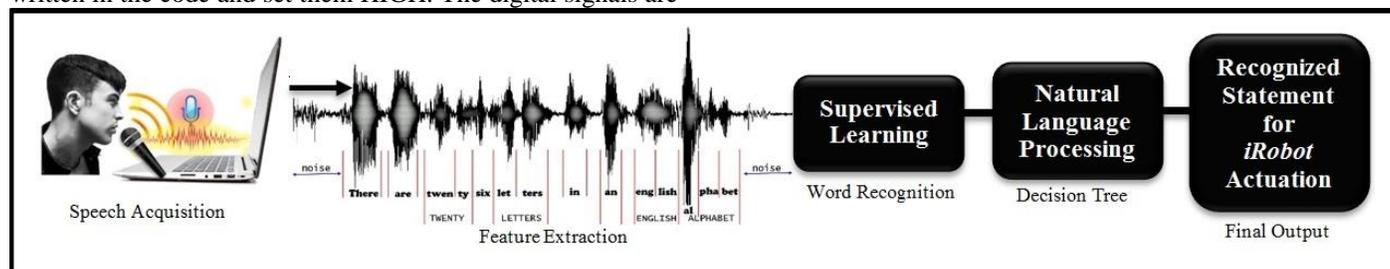


Fig. 8. Representative Flow Diagram for the Proposed Speech Recognition Algorithm

In case of humans, speech is basically a combination of specific sounds that, in a specific combination, makes up different words, which we have come to associate with different actions and meanings. Moreover, specific combinations of these words in itself, makes up sentences and grammatical structures that fine tune the communication procedure. Humans produce various sounds by varying the shape of the vocal tract, with varying frequencies. Herein comes the problem of the various users and gender differences. Due to different characteristics of properties like pitch, frequencies, etc. sound quality varies widely over

genders and even among various persons. This property is used to implement speech recognition as a biometric security measure. Since speech characteristics vary from person to person, the speech can be used for user identification. This property, in turn, gives rise to two types of speech recognition, a user dependent speech recognition and user independent speech recognition. The former, as stated above, is trained to and recognizes speech from only a single user, which is the same as that used in biometric security applications. The latter, however, is a wider, and more complicated implementation of speech recognition, wherein the system

recognizes speech irrespective of who the user is. This kind of speech recognition is finding a wider variety of application as it has a tremendous applicability in the field of assistive systems, which is one of the primary fields of Human-Computer Interaction.

In any speech recognition system, there are basically three steps involved in creating a successful system that can give a proper threshold of recognition. The steps involved are:

- i. Speech Acquisition.
- ii. Speech Analysis, and
- iii. Subroutine linking or User feedback.

For the first part, a simple microphone is used to acquire the speech signal. Care should be taken that microphone is unobstructed. Once the signal is acquired, the signal is then analysed. One of the most primitive filtering is the filtering of the background noises and breaking it up into blocks. The blocks are then passed through various pre-processing steps and made ready for matching with pre-trained sets, which are similarly analysed and readied in large datasets. Some of the popular algorithms used in speech recognition are Dynamic Time Warping[18], Hidden Markov Models[19], Neural Networks or Deep Neural Networks[20] and Deep Learning algorithms. Of these, algorithms based on Hidden Markov Models have been proven to be historically robust and to give a very good threshold of speech recognition. Recently however, Deep Learning has been shown to offer good applicability in the field too and some extensive works are being done in the said field.

For our system, we chose to go with the Speech Application Programming Interface or SAPI, developed by Microsoft for the purpose of Speech Recognition and Speech Synthesis. It is a very robust system with a high rate of correct detection and highly accessible programming interface that allowed us to easily code it to work with our system. Since the system is aimed at assistive systems, the ability of speech synthesis is an added advantage. Moreover, being freely distributable, it is available on all windows platforms supporting Windows 98 onwards. The version that we used was SAPI 5 with Visual Basic 2008. The SAPI allows one to recognise predetermined words that are linked to a user defined dictionary. These dictionaries can be dynamically loaded during runtime. But since we aim to make the system a complete and seamless assistive system, we implemented the speech recognition in such a way that the user will speak a natural sentence and the system will be able to recognise what the user wants to command. The intuition is similar to how the human's hearing and comprehension works. Often it has been seen that in order to understand the communication intended behind a spoken sentence, it is not always necessary for the receiver to understand each and every word, but rather an acceptable number of words. For the words that were lost in the communication medium, they can be filled in by pre-emptive filling based on words preceding it. Hence using this intuition, we model a search tree for the Natural Language Processing in our speech recognition system. The SAPI gives us a robust word recognition system which we use to create the probable grammar dictionary. When the user speaks a sentence, our program, using the isolated words, traverses

through a pre-modelled search tree. The path of traversal of the tree is then used to determine what the actual intended command by the user is. Once that is determined, a contracted command string is generated which is used by the program to execute the associated sub-actions. For testing purposes we used Visual Basic to send commands to an iRobot Create which is programmed in MATLAB through iRobot Roomba's Open Interface (ROI) protocol. The result was the user just needed to say simple sentences like, "Trace 5 squares" or "Move 10 meters" or "Follow an Object" etc. so as to make the iRobot traverse a square 5 times or to travel 10 meters forward or exactly traverse a path in real time corresponding to the pattern traced by a colored object at the control station end using color based segmentation. However the shapes to be traced has been tested on Squares, Rectangle, Triangle, Circle and the Shape of digit 8 respectively. The User Interface was done in Visual Basic, which can listen for speech in the background asynchronously, thus leaving the system free to execute other functionalities when valid speech is not spoken. The subsequent interfacing of iRobot Create with the control station was done by constant power cycling of the Hardware with MATLAB through a high range Bluetooth Access Module (BAM).



Fig. 9. Setup of Voice controlled iRobot Create over Bluetooth Connectivity

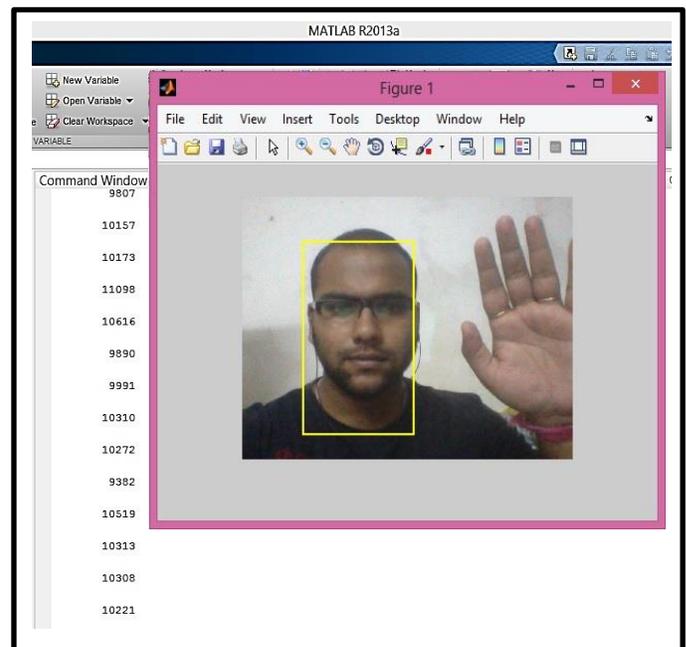


Fig. 10. Screenshot of MATLAB Command window and webcam preview showing Area Results of Face Tracking Algorithm which establishes a sharp demarcation between Face like regions and Non-face like regions in Real-time

V. EXPERIMENTAL EVALUATIONS AND RESULTS

In this paper we put forward a technology which states the concept of controlling embedded wireless systems using two distinct Human intuitions i.e. voice commands and Face movements. Face being the primary element of human identity is a distinct feature (except for the Identical twins) to distinguish between two persons. So detecting and tracking face in real-time with substantial detection results to control various system functions is of primary importance in our approach. Color is the property by virtue of which objects are generally classified. However the face detection by extraction of skin color using color based segmentation is highly dependent on the optimal lighting conditions. Moreover the quality of the acquired image highly depends on the amount of light incident to the aperture of the webcam. The performance of the face tracking algorithm proved to be significantly effective in moderate and ambient light as well as high luminous intensities. It is evident from Fig.10 that the geometrical analysis distinguishes face pixels from non-face pixels quite efficiently. The palm, being classified among the non-face pixels although larger in area than the tested face gets rejected as it doesn't fulfill the criteria of the eccentricity or the elliptical aspect ratio conditions. If there is no face available, even after the cache Count equals to 10, then an alert message appears on the command window "No Face Present" and the video acquisition stops after a while. The screenshot displays the live preview (320x240) video resolution of the real-time face tracking algorithm with corresponding areas of the detected face like blob in square pixels units in MATLAB's command window. TABLE I is generated on the basis of the screenshot result.

TABLE I. EXPERIMENTAL RESULT ANALYSIS FROM FIGURE 10

Iterations	Area of Face-Like Blob	Mean Area	Detection Rate
1	9807	10218.35	95.97%
2	10157	10218.35	99.39%
3	10173	10218.35	99.55%
4	11098	10218.35	91.39%
5	10616	10218.35	96.10%
6	9890	10218.35	96.78%
7	9991	10218.35	97.77%
8	10310	10218.35	99.10%
9	10272	10218.35	99.47%
10	9382	10218.35	91.81%
11	10519	10218.35	97.05%
12	10313	10218.35	99.07%

13	10308	10218.35	99.12%
14	10221	10218.35	99.97%
Mean Area	10218.35	Average Detection	97.32%

From the above table it can be concluded that there have been 14 iterations out of which the Minimum area is 9382 sq. pixels at the 10th iteration while the Maximum area is 11098 sq. pixels at the 4th iteration.

$$Mean Area = \frac{\sum_{i=1}^{14} Area(i)}{14} \dots\dots (1)$$

The mean area of the 14 iterations is 10218.35 sq. pixels. So the Detection rate is calculated for each iteration as a percentage deviation from the Mean area.

$$Detection Rate(i) = [1 - \frac{Abs\{Area(i) - Mean Area\}}{Mean Area}] \times 100\dots\dots (2)$$

Here the absolute value of  $[Area(i) - Mean Area]$  determines the amount of variation of the data from its tabulated arithmetic mean. According to the equation, when expanded, the magnitude of this variation is subtracted from the *Mean Area* itself, the resultant of which is calculated as a percentage of deviation from the arithmetic (*Mean Area*) for each iteration and hence determine the working consistency of the algorithm for real-time detection of a static face.

But for low lighting conditions the grayscale level threshold drastically changes leading to the rectangular bounding box alternately switching between a non-face skin-like blob and the detected face-like blob for successive frames, which results into increased number of false positives and a higher magnitude of environmental noise. So environmental noise is a multiplying factor which effects detection to a higher extent. After testing in various conditions, as shown in Fig.11, it is highly recommended that bright intensities of hue values of these respective colors (Red, Orange and Yellow) of the visible spectrum should not be present in the background as a major color. It might interfere the performance of the algorithm with poor detection rates.

However the detected and tracked face returns the position vectors of the rectangular bounding box which helps calculate its centroid. The abscissa of the centroid distinguishes between gestures for Left and Right Motion evaluations while the difference in area gives the information for Forward and Backward motions respectively which is evident from TABLE II given below.

To establish Wi-Fi connectivity in between the control station and Self-developed prototype we used a 3G Wi-Fi Dongle (MTS MBlaze). The Wi-Fi shield scans for all the available networks as shown in Fig.12(b) with their respective Signal Strengths and matches for the SSID name written in the code. If match is found then it checks for the encryption type and authenticates the passkeys. Once the connection is established the Wi-Fi shield is assigned with a local IP and the Arduino Serial Monitor asks to browse the IP address in order to access the HTML page of navigational controls for the Prototype activation as in Fig.12(c)

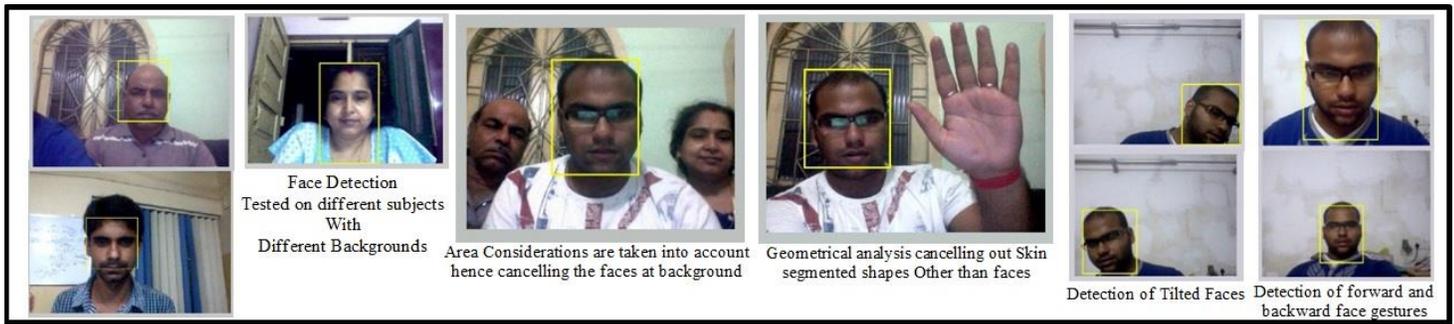


Fig. 11. Experimental Results of the Face tracking Algorithm

TABLE II. PROTOTYPE ACTUATION WITH CORRESPONDING GESTURE PERFORMANCE

Body Posture	Face-Gesture Performance	Gesture Acquisition	Servo Motor Movements	DC Motor Movements	Prototype Actualization
		If Centroid abscissa is within (120 to 180) pixels and Area is within (5000 to 11000) Sq.Pixels	Default Position 90 degrees	Both the Motors (M1 and M2) Do not Move	 Robot Halt
		If Centroid abscissa is within (120 to 180) pixels and Area is (more than 11000) Sq.Pixels	Default Position 90 degrees	Both the Motors (M1 and M2) Move in the clockwise Direction	 Move Forward
		If Centroid abscissa is within (120 to 180) pixels and Area is (less than 5000) Sq.Pixels	Default Position 90 degrees	Both the Motors (M1 and M2) Move in the anti-clockwise Direction	 Move Backward
		If Centroid abscissa is less than (120 pixels) and Area is within (5000 to 11000) Sq.Pixels	Servo Sweeps to 45 degrees and remains fixed	M1 moves anti-clockwise while M2 moves clockwise respectively	 Turn Left from Rest
		If Centroid abscissa is more than (180 pixels) and Area is within (5000 to 11000) Sq.Pixels	Servo Sweeps to 135 degrees and remains fixed	M1 moves clockwise while M2 moves anti-clockwise respectively	 Turn Right from Rest
		If Centroid abscissa is less than (120 pixels) and Area is (more than 11000) Sq.Pixels	Servo Sweeps to 45 degrees and returns to default 90 degrees	Both the Motors (M1 and M2) Move in the clockwise Direction	 Drift Left
		If Centroid abscissa is more than (180 pixels) and Area is (more than 11000) Sq.Pixels	Servo Sweeps to 135 degrees and returns to default 90 degrees	Both the Motors (M1 and M2) Move in the clockwise Direction	 Drift Right

While testing and performing several experimental trials as in TABLE III, it has been found that there is a sufficient amount of lag for a few milliseconds from Gesture performance to command signal generation to the Prototype actuation.

This lag due to network congestion might highly affect real-time applications and may lead to Gesture misinterpretation as well as erroneous results. This lack of synchronization of the system implementation can be overcome by proper systematic network programming, as over here we have just provided a proof of concept for the system implementation.

TABLE III. EXPERIMENTAL OBSERVATIONS FOR PROTOTYPE ACTUATION WITH CORRESPONDING GESTURE PERFORMANCE

Observations	No. Of Trials	No. Of Successful Outcomes	Success Rate
Move Forward	20	19	95%
Move Forward Faster	20	19	95%
Move Forward Super Faster	20	18	90%
Move Backward	20	18	90%
Move Backward Faster	20	17	85%
Drift Left (Low Speed)	20	15	75%
Drift Left (Medium Speed)	20	17	85%
Drift Left (High Speed)	20	16	80%
Drift Right (Low Speed)	20	15	75%
Drift Right (Medium Speed)	20	15	75%
Drift Right (High Speed)	20	17	85%
Turn Left	20	16	80%
Turn Right	20	17	85%
Robot Halt (Stop)	20	16	80%

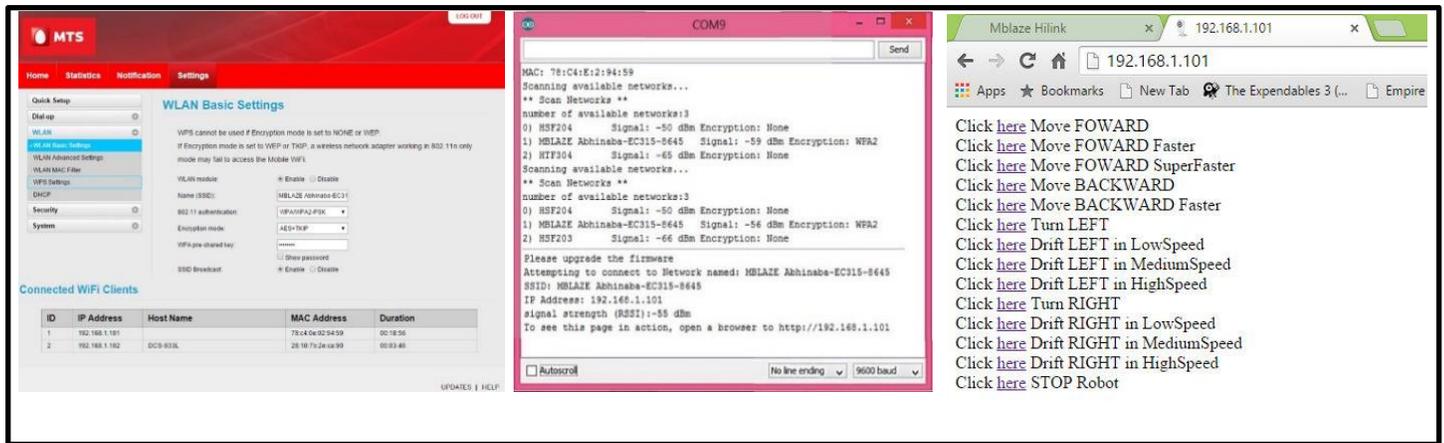


Fig. 12(a). Network Home Page Showing SSID broadcast and number of hosts connected

Fig. 12(b). Available networks and the connected Network displayed by the Arduino Serial Monitor

Fig. 12(c). URL of the Wi-Fi device representing the HTML webpage for prototype actuation

However the system implementation aims at several industrial applications of which surveillance security is one major interest. In order to fulfill the purpose we have successfully integrated a network camera into the system which sends the live prototype surveillance feed to the control station wirelessly over the web through a Wi-Fi connectivity. From Fig.12(a) we can see that the Network assigns two different IP addresses to the two Wi-Fi devices connected in which the later one is the network camera with IP address 192.168.1.102 and on browsing this IP address we can access the live surveillance preview of the prototype both in light and dark conditions which is evident from Fig.13.

However to increase the flexibility and security of our implementation we introduced a second type of human intuition to control wireless systems i.e. speech, which solely serves the purpose to navigate and direct systems in a more definite way. TABLE IV. shows the tested results of the speech recognition algorithm. From the above table it is confirmed that the implementation employed here is user dependent i.e. it gives a higher detection rate for the person with whom it has been calibrated. As such a short calibration of the system is essential, albeit a single time calibration. But once the calibration is done, the system achieves a high rate of recognition for the given user even on a wide and varied dictionary of words as well as on complete sentences.

The user dependence, in lieu of our system is also an added bonus. Since, after the initial setup of the system, which includes, among other things, the calibration, the same system can be used to identify the user and serve as a voice print identification. As the implementation is geared towards biometric security based assistive systems or surveillance systems, the system responding to only particular user is an advantage. Hence we decided to go with particular user recognition rather than multi user recognition. Hence based on the recognized statements we have implicitly evaluated certain voice commands which are directly implemented on the iRobot Create for various action execution and can be changed as per necessity.

TABLE IV. EXPERIMENTAL OBSERVATIONS FOR THE SPEECH RECOGNITION ALGORITHM

User Dependency	No. of Trials	No. of Successful Recognition	No. of False Positives	Success Rate
Particular User	30	26	4	86.67%
Multiple Users	40	13	27	32.5%

## VI. DISCUSSIONS AND CONCLUSIONS

From the favorable statistics of results it can be concluded that the implementation of Real-time Face Tracking algorithm is quite efficient with high accuracy rates at moderate and ambient lighting conditions, which overrules many other techniques as it can detect both tilted and rotated faces to some extent with a few limitations. These limitations that affect the detection rates are:-

- ✓ Luminance factor of the Environment with ambient lighting conditions. However, multicolored lights might detect faces but affect the tracking efficiency.
- ✓ Monochromatic Color predominance is another major factor that might affect subsequent detection rates. However, bright intensities of the color spectrum with Hue values ranging from 0 to 45 and 210 to 239 present as a dominant color in the background may produce invariant segmentation where skin tone gets diminished and hence result in false positives.
- ✓ The last but not the least is a hardware limitation where detection depends on the quality of image acquisition that is directly dependent on the aperture width of the Webcam. For real-time processing the more is the aperture width, higher is the quality of the image acquired.

The Arduino based wireless system is considered to be effectively optimized and secured as it provides two modes of control, the first – is a Machine level method by directly

accessing the local IP address of the hardware device, while the second one – is through the User-friendly face motion based implementation, both the modes using a protected Wi-Fi connectivity. In contrast to it the user dependent speech recognition based implementation on iRobot Create over Bluetooth connectivity is a more directed and definite form of Human-Computer Interfacing. As for the speech recognition, the user dependency at initial calibration can prove to be a security authorization code for accessing and controlling technologically advanced military armors or weapons. Moreover, the face gestures can be used to control their directions and motions. Similarly, we can integrate wireless systems with these methods for surveillance and monitoring activities, for keeping a check on intruder activities, spying on enemies at Defense Academies or Military Base Camps.



Fig. 13. Screenshot of Network Camera preview in normal and in dark conditions

The live face detection can also act as a live password for accessing surveillance systems in satellite substations. Touchless interfaces are especially useful in healthcare

environments where information are accessed while maintaining total sterility. The face tracking algorithm can be used to design assistive monitoring systems at Intensive Care Units in synchronization with other medical instruments where continuous tracking of patient's face, among other things, can provide necessary information about the patient's instantaneous condition. However, if any anomalous face movement is found then, immediate medical assistance would be fetched by the system.

Moreover, the unit configured onto a wheelchair with a camera fitted on the hand-rests focusing the face of the person on it will enable the physically challenged to control motion and navigation of his or her artificial legs comfortably within their domestic limits. Additionally the face gestures performed with different body postures may restrict further physical immobility and might act as a key for the recovery of patients with impaired legs or spinal deformity.

However, these gesture interfaces are also gaining importance in the entertainment fields where touch-free motion based games are being commercialized with various external hardware setups. However, our method can be adopted to implement external hardware free gaming environments with biometric user security where gestures based on face or body movements can be tracked to evaluate various gaming controls with the PC webcam as the only sensor involved.

The proposed work can also be an authoring method capable of operation controlling motions in the fields of industrial automation and control. Vacuum cleaners can be automated by hardware interfacing robots (as in iRobot Roomba) to reduce mechanical effort as well as manpower in various Industrial sites, Resort, Shopping malls, Housing complexes, etc.

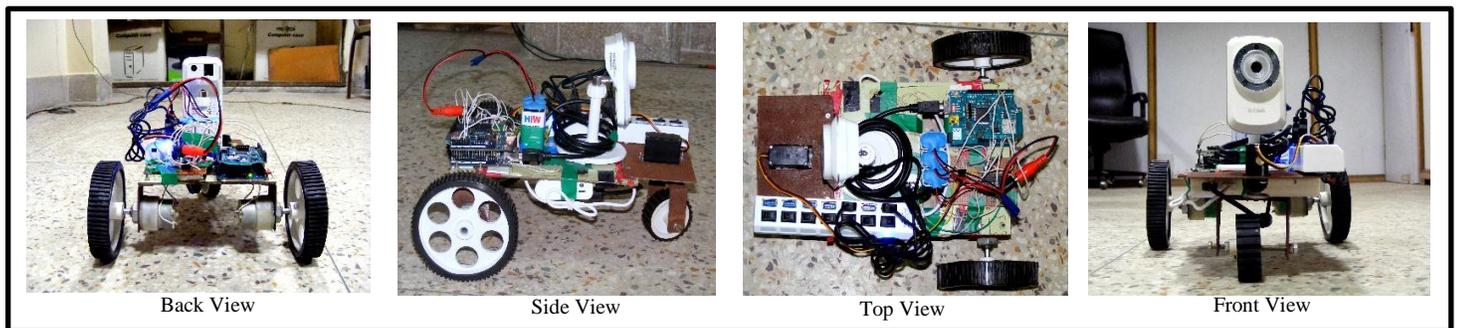


Fig. 14. Arduino based Self-Developed Wi-Fi Robot

As the paper presents an embedded wireless prototype with almost all the features of a geared automobile, so the system design if adopted by the automobile designers can prove to be an effective methodology to integrate real-time gesture controlled automobiles.

The gesture-based access based on face motions (with the user's face being a live biometric authorization) to evaluate various automotive actions in real-time can act as a second mode of control with the geared mechanical access being the primary mode of control.

So as a whole we can conclude that the system implementation is quite a success with the overall experimental results showing high accuracy and detection rates with a wide range of possible advanced commercial implementations as well as industrial applications. Although the Real-time system lag in milliseconds is an interfering factor, it can be eliminated with further research. However, the major advantage of our system over other systems is that it provides real-time face gesture recognition, leading to an effective and natural way for controlling embedded wireless systems.

The proposed speech recognition method is expected to provide effective and implementable solutions for not only just industrial robots but also for intelligent embedded robots like humanoids.

## VII. FUTURE WORKS

The Present System is a real-time gesture tracking system used for motion control of Wi-Fi robots. The face tracking algorithm can be enhanced with a Graphical User Interface (GUI) which presently runs on Standalone scripting codes. The GUI will provide a more user-friendly as well as robust interface for interaction. The face detection algorithm doesn't include a face recognition or Gender detection part, which might make the system much more secured with biometric face recognition for restricted access only. Moreover, the video footage of the Network camera so received can be further processed with various computer vision algorithms to make the robots more intelligent and adaptive to the surrounding environment. Advancements in the speech recognition along with interactive AI as that implemented on CORTANA or SIRI can be used to increase adaptations of the system. Introducing a Real-Time Object Tracking algorithm on the received preview of the network camera would be a breakthrough for such systems and can be extensively used to design assistive robots for elderly aid. Thus, Simultaneous Localization and Mapping (SLAM) can be introduced using the post-processing feature enacted upon the video feed of the network camera. Also, the method can be implemented on advanced automobiles to assist driving and control intellectually.

The Night Vision mode can be used to track objects even at dark provided the Algorithm must be flexible enough and should be tested at various luminous intensities. It can assist elderly drivers with impaired vision or color blindness or night blindness by tracking on road vehicles, traffic signals in real time and thus provide a safe and sound drive by collision detection or avoiding Potholes (especially at Indian roads) even in the dark.

However, these computer vision algorithms provide a more friendly and intellectual interaction of the user with the system. Special robots can be created to assist Room services and housekeeping purposes while accessing them remotely from distant places over Wi-Fi connectivity. Such robots can be widely used in industrial automation and can be implemented for automating cleaning of toilets at public places like shopping malls, railway stations, etc.

The system has a high scope of importance in biomedical applications to assist the physically challenged people. Such systems can be used for maneuvering physical activities of people with paralyzed limbs. If the system is integrated on mobile wheelchairs, the body movements for acquiring face gestures might prove to be a great assistance to partially overcome their physical immobility.

As a whole it can be concluded that the system has a huge scope of further research and application that can prove to be effective in various fields.

## ACKNOWLEDGEMENT

We would like to thank Arnab Pal, Techno-Marketing Consultant at ESL (e-School Learning) for helping us with the technical details of the project and ESL, [www.eschoollearning.net](http://www.eschoollearning.net), Kolkata for giving us a platform to work on this project.

## REFERENCES

- [1] F. Arce, J. M. G. Valdez, "Accelerometer-Based Hand Gesture Recognition Using Artificial Neural Networks" in *Soft Computing for Intelligent Control and Mobile Robotics Studies in Computational Intelligence*, vol. 318, pp 67-77, 2011.
- [2] Luigi Lamberti and Francesco Camastra, "Real-Time Hand Gesture Recognition using a Color Glove," *Department of Applied Science, University of Naples Parthenope*, 2010, pp.321-344.
- [3] J.S. Kim, C.S. Lee, K.J. Song, B. Min, Z. Bien, "Real-time hand gesture recognition for avatar motion control," *Proceedings of HCI'97*, pp. 96-101, February 1997.
- [4] Chao Hy Xiang Wang, Mrinal K. Mandal, Max Meng, and Donglin Li, "Efficient Face and Gesture Recognition Techniques for Robot Control", CCECE, 1757-1762, 2003.
- [5] Gesture Controlled Robot using Kinect <http://www.e-yantara.org/home/projects-wiki/item/180-gesture-controlled-robot-using-firebirdv-and-kinect>
- [6] Abhinaba Bhattacharjee, Indrani Jana, Ankana Das, Debasish Kundu, Sudipta Ghosh, Sauvik Das Gupta, "A Novel probabilistic approach of colored object detection and design of a gesture based real-time mouse tracking along with virtual teaching intended for color blind people," *2nd International IEEE Conference on Signal Processing and Integrated Networks (SPIN)*, 19-20 February 2015, Noida, India., pp.512-519, ISBN: 978-1-4799-5990-7.
- [7] Tsang, W.-W.M, "A finger-tracking virtual mouse realized in an embedded system", *IEEE Intelligent Signal Processing and Communication systems (ISPACS)*, 2005, pp. 781-784.
- [8] Rafael C. Gonzalez, *Digital Image Processing*, Pearson Education, 3rd ed., 2008.
- [9] S Mahanta, S Ghosal, P Das, A Datta, S Debnath, S Das Gupta, "A Novel approach for Graphical User Interface development and real time Object and Face Tracking using Image Processing and Computer Vision Techniques implemented in MATLAB", *IOSR Journal of Computer Engineering (IOSR-JCE)* e-ISSN: 2278-0661, p- ISSN: 2278-8727 Volume 15, Issue 5 (Nov. - Dec. 2013), PP 61-68 [www.iosrjournals.org](http://www.iosrjournals.org).
- [10] Poonam Dhankar and Neha Sahu, "Edge based Human Face Detection using Matlab", *Proceedings of IRF International Conference*, 16th February 2014, Goa, India. ISBN: 978-93-82702-58-0.
- [11] P. Peer, J. Kovac, F. Solina, "Human Skin Colour Clustering for Face Detection," *EUROCON1993*, Ljubljana, Slovenia, pp. 144-148, September 2003.
- [12] C. Garcia and G. Tziritas, "Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis," *IEEE Trans. Multimedia*, 1(3), pp. 264-277, 1999.
- [13] Viola, Paul and Michael J. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features", *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001. Volume: 1, pp.511-518.
- [14] Foley, Van Dam, Foriner, and Hughes, "Computer Graphics: Principles and Practices", *Addison- Wesley, second edition in C*.
- [15] Partha Das, Anirban Sarkar, Soumyadeep Halder, Debasish Kundu, Sudipta Ghosh, Sauvik Das Gupta, "A Novel approach towards detecting faces and gender using skin segmentation and template matching," *2nd International IEEE Conference on Signal Processing and Integrated Networks (SPIN)*, 19-20 February 2015, Noida, India., pp.431-436, ISBN: 978-1-4799-5990-7.
- [16] Michael Padilla, Zihong Fan – EE368 Digital image processing Project "Automatic Face Detection Using Color Based Segmentation and

- Template/Energy Thresholding”, Department of Electrical Engineering, Stanford University, Spring 2002-2003.
- [17] Mingli Song, Tao Dacheng, Sun Shengpeng, Chun Chen, S.J. Maybank, “ Robust 3D face landmark localization based on Coordinate coding”, *IEEE transaction on Image Processing Volume23*, 21 October 2014, ISSN: 1057-7149.
- [18] G Poli, J.F. Marie, J.H Saito, “ Voice command Recognition with Dynamic Time Warping (DTW) using Graphics Processing Unit (GPA) with Compute Unified Device Architecture (CUDA),” *International Symposium on Computer Architecture and High Performance Computing* ISSN 1550-6533 Pg. 19-25.
- [19] A.D Deelip, C.C. Shekhar, “HMM based Intermediate matching Kernel for classification of Sequential patterns of speech using Support Vector Machines”, *IEEE transactions on Audio Speech and Language Processing* ISSN 1558-7916 Pg. 2570-2582.
- [20] G.E Dahl, Deng Li, Yu Dong, “Context-Dependent Pre-trained Deep Neural Networks for Large-Vocabulary Speech Recognition”, *IEEE transactions on Audio Speech and Language Processing* ISSN 1558-7916 Pg. 30 -42.