**CS5590BD Final Report**

**By**

**TRM-PG05**

**BHARGAVA GELLABOINA (SG22)**

**RAMESH KUTHALA (SG22)**

**PAVAN KUMAR BOLLARAM (SG23)**

**RAJASEKHAR INTURI (SG23)**

**Motivation:**

The main motivation behind this project is to involve the robot in unmanned missions where a human presence is impossible. A robot can be controlled remotely by another device using networking socket concepts so that it can be monitored easily and it can be programmed to think by itself. A self-thinking robot can avoid the obstacles and move forward by its own and at the same time it can send required information to controlling unit like capturing visuals and sending out videos of the location where robot moves around.

**Significance:**

Although there are many robots present in the market very few of them works in real time manner. We can train the robot to make it think by itself using machine learning concepts. This is where the Big Data comes into picture. Capturing the sensor values in real time manner and make an analysis to extract some features. The features can be further used to guide the bot in decision making.

**Technologies:**

* Objective-C programming
* Socket programming
* REST services
* ROMOTIVE SDK
* Android
* OpenCV Image Processing

**Frameworks:**

1. CoreVideo
2. OpenCV2
3. CoreMedia
4. CoreImage
5. CoreFoundation
6. ImageIO
7. AssestsLibrary
8. AVKit
9. CFNetwork
10. Security
11. CoreMotion
12. AVFoundation
13. ExternalAccessory
14. Quartzore
15. UIKit
16. Foundation
17. CoreGraphics
18. System configuration framework
19. LIBZ
20. LIBSQLITE 3.0

**Related work**

Spirit Rover is one of the first robot deployed over mars by NASA to study the planet conditions and its atmospheric changes. These studies are very crucial in defining how far the climatic conditions satisfies living conditions for human. We motivated from this and we thought of implementing bot which does surveillance job and measures the required values and post to the controlling station.

**Architecture**

**Controller (Android Device):**

Android platform is most widely used open source platform in the world and most effective user interfaces can be created by android platform. In this project we have used android device to communicate with ROMO using networking socket protocol. A series of commands are sent to ROMO which constantly listening to commands over socket. ROMO listens to these commands and makes movements accordingly.

ANDROID DEVICE

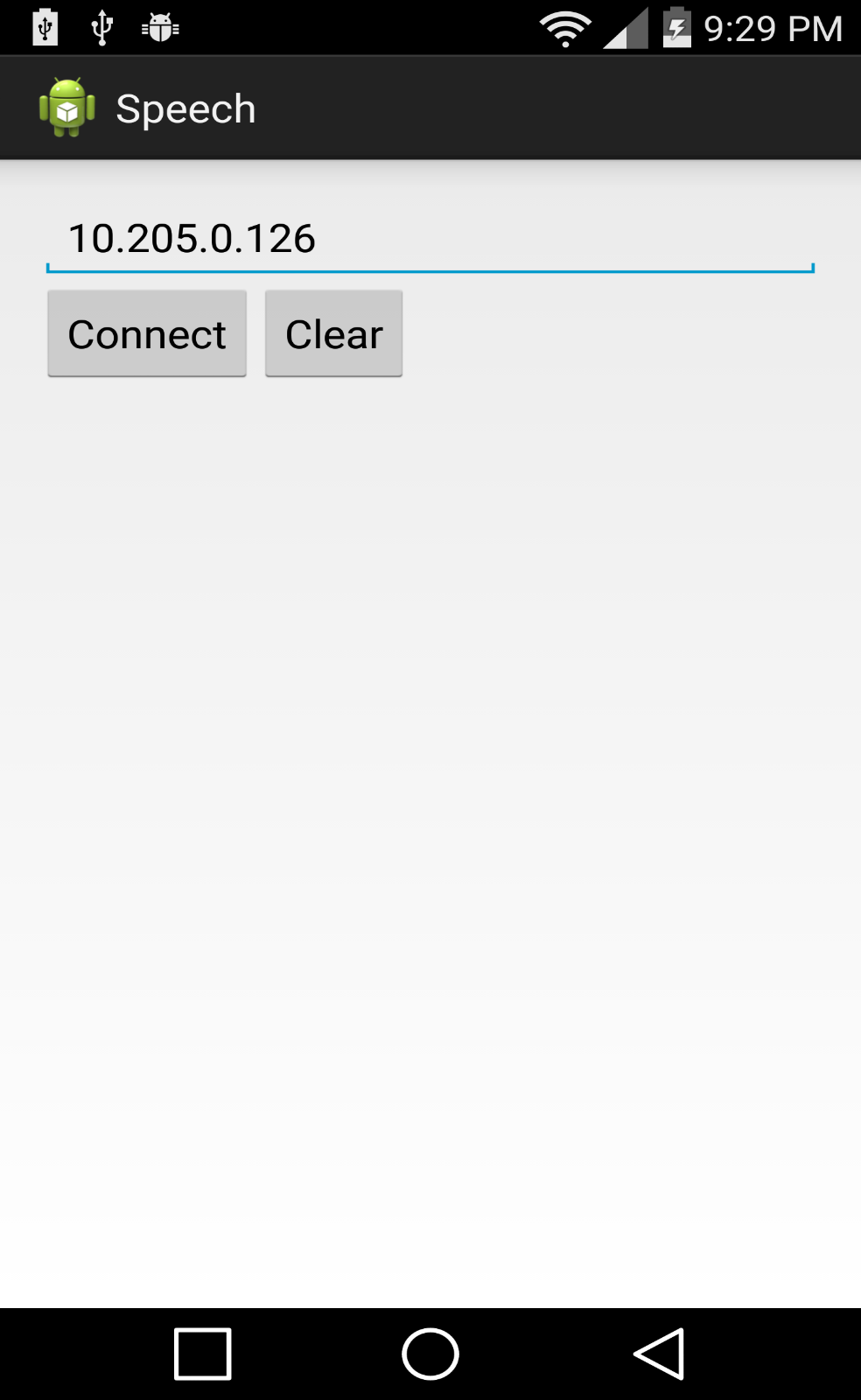
CLIENT SOCKET

IPhone5/ IPhone5S

SERVER SOCKET

**video**

**Architecture**

****

**Android View to connect to IPhone Server**

****

**Android View to send Voice Commands**

**Implementation – functions:**

The Romotive SDK allow us to write our own programs to Romo robot. The SDK is broken into multiple frameworks and we can choose any framework depending on the requirement.

The Romo is generally operated with the help of IOS devices starting from IPhone 5 and other latest versions. So once if can program IOS device we can control bot movements. We have used Objective-C programming language for IOS programming.

OpenCV is another framework used in this programming which is mainly used for computer vision and image processing. A set of programming functions are imported using this framework.it is written in C++ and very easy to use.

The android device and ios device are connected using socket programming where sockets are created to send and receive commands over TCP/UDP networking protocols so that both devices can communicate wirelessly. Only limitation of socket programming is both devices needs to be connected to same network. Once the commands are received from android the IOS device internal framework interprets and does customized functions defined by us.

**System features**:

* Navigating ROMO movements using voice commands
* ROMO movement automated using the accelerometer sensor
* Image capturing
* Face detection and color detection
* Gesture recognition using Android touch and Motion Sensors
* Activity Recognition and Tracking
* Live Streaming and monitoring
* Image classification using spark system

All the above features are implemented and tested using the RomoMe. We have used IPhone Accelerometer sensor to collect the x, y and z movements and control the robot movement accordingly. If there is an elevated path then robot will increase its acceleration to climb it and if the path is declined then the speed is controlled.

**Gesture Recognition**

1. **Touch Events:**

In this project we implemented Gesture recognition with ROMO, as part of this touch movements on the android device are recognized and categorized for different movements of ROMO. Touch swipe on android device will result in changes of velocity values in X direction and Y direction. These values are measured using VelocityTracker class supported by android library. getXVelocity() and getYVelocity() methods are used to compute the velocity in each direction.

We observed that swipe towards right side result in positive values of X and Y values whereas swipe towards left side result in negative values velocity in X and Y direction. This kind of pattern can be used to drive the ROBO in right and left directions.

Once swipe movements are done these values can be sent to ios device mounted on ROMO using socket connection

Android

ROMO

Move RIGHT

Move LEFT

RIGHT swipe

LEFT swipe

**GESTURE PATTERNS**

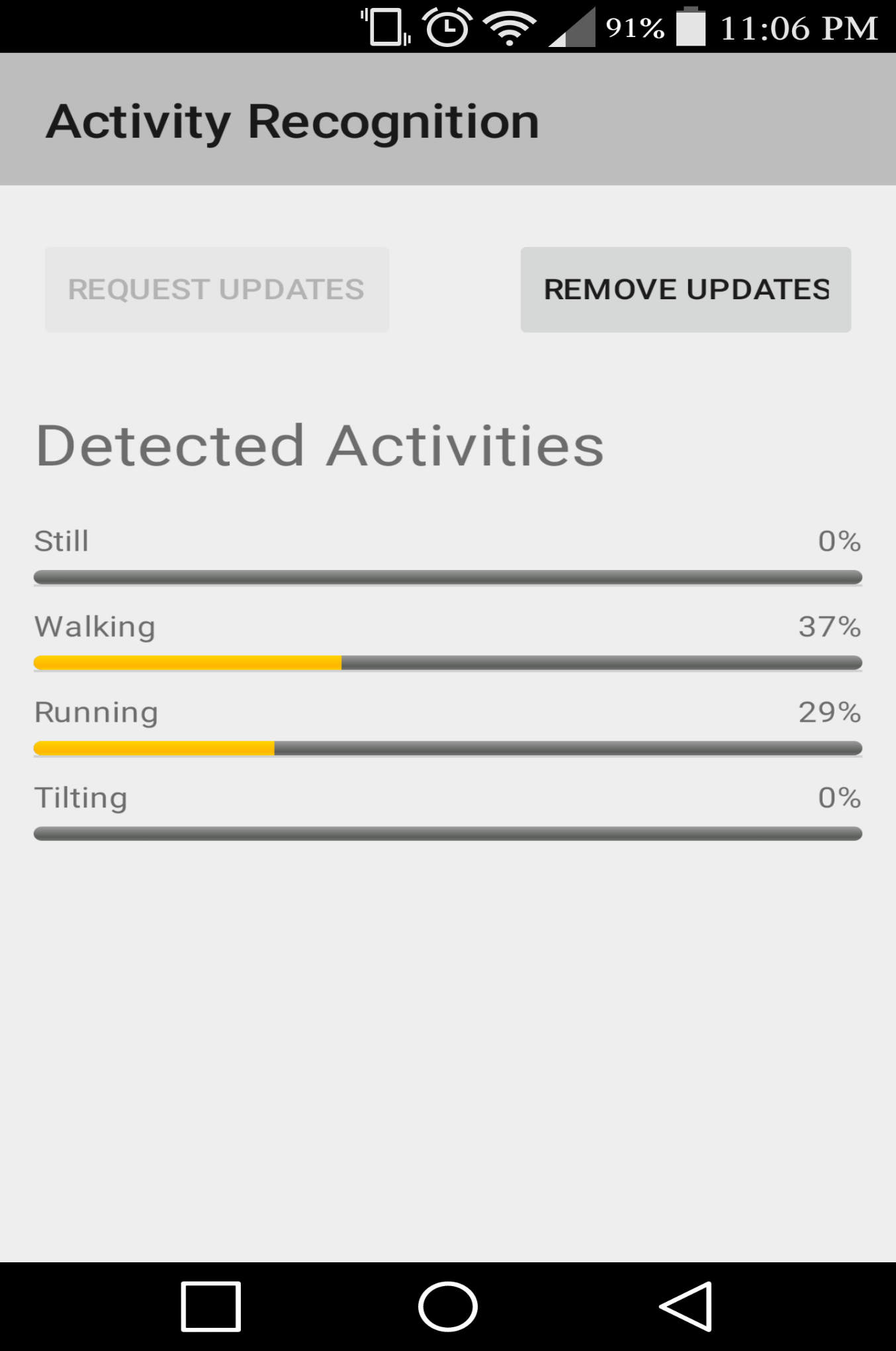
1. **Motion Event:**

We use Accelerometer data to detect the gestures. Accelerometer data is collected on the android client and the gesture is detected using our own gesture recognition algorithm. Steps followed to detect the gesture are below.

1. Collect the accelerometer data from the android client.
2. Filter the data collected.
3. Follow the algorithm for detecting the gesture.
4. Send the gesture to the IPhone and command Romo to perform to appropriate function.

**Activity Recognition**

We developed Android Activity Recognition and give the detected activity to ROMO and guide it to perform the same. For example if the activity detected is walking then ROMO moves with a speed of 1.0, if it is running then the speed is 4.0. An interface for the activity recognition is also developed and below is the screenshot of it.



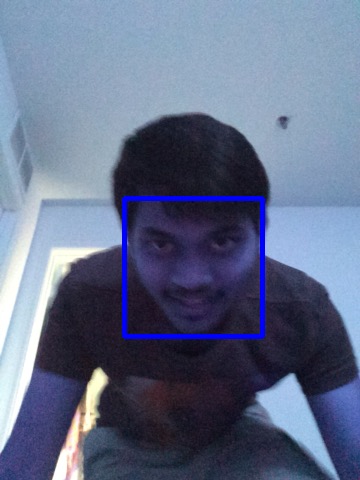
 WALK ACTIVITY ---------------------- > Less speed

 RUNNING ACTIVITY ----------------------- > More speed

We can track the movements if we can persist these reading to a real time data base so that any significant changes in person activities can be analyzed.

**OpenCV Face Detection and Image Capture:**

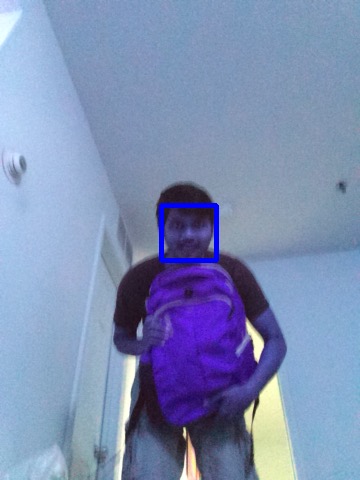
We have used Objective-C OpenCV framework to detect the faces using the camera view. Whenever a face is detected in the camera picture is taken and stored into the gallery.

****

**This image is captured by the robot when the face was detected.**

**OpenCV Color Detection:**

We used Objective-C OpenCV framework to detect the colors and guide the robot to perform some actions according to the specified work.



**Above image was captured when the face was detected by robot and red color is detected and its movement is stopped.**

**LIVE STREAMING AND MONITORING**

As part of monitoring ROMO about its movement we implemented video streaming with the ios device mounted on ROMO. When ROMO starts its movements it feed the video using its front camera to the server using RT protocols and this video is broadcasted to controlling device using secured URL. At the same time using this video ROMO movements can be tracked and altered using voice commands.

Capturing video from using front camera

server

ROMO (ios) RT Protocal

Voice commands Broadcast URL

Controlling Device/ Android

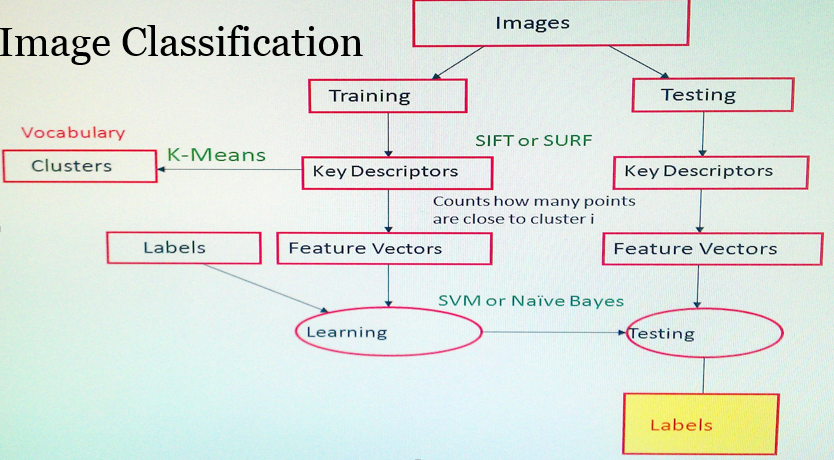
Video Feed

[**http://www.wowza.com/forums/content.php?20-apple-ios-devices**](http://www.wowza.com/forums/content.php?20-apple-ios-devices)

We followed the above link to perform the video live streaming from IOS to WOWZA sever and the broadcast link is provided to the client. It uses Remote Transfer Protocol to stream the video data to the server.

**REAL TIME SPARK IMAGE PROCESSING**

Images captured by ios device are sent to SPARK for imaging analysis and in return best matched label from spark training set can be sent back by SPARK. As part of this implementation SPARK model was created which accepts input images from ios device for analysis. Feature extraction is done at spark end using bag of words and image histograms are calculated. Once all of this was done features are extracted and they are compared with already existing test data and a best matched image i.e very closest image from test data is returned. This same image can be sent to the ios as a recommendation from spark.



We followed same approach as mentioned above except SVM model. We have used Naïve Bayes model for classification of images. At the end of flow labels are returned.

In this project we have tested few images by giving input to ios device. When our ROMO captures image it send them to spark system where we have taken few words like FIRE, High Temperature, water. When ever ROMO captures any one of these images it classifies that image and return to user. This is how our ROMO captures and informs the user with its analysis.

**Deployment:**

GitHub: <https://github.com/pavankumar-b/Bigdata/tree/master/Final%20Project>

**Responsibility:**

**Tasks:**

1. Android Client Development
2. IPhone Server Development
3. Android activity recognition
4. Android gesture recognition with touch event
5. Android gesture recognition with motion event
6. OpenCV Face Detection
7. OpenCV Color Detection
8. Capture Picture
9. IPhone Sensor based robot movement
10. SPARK setup
11. SPARK Real time streaming
12. SPARK Machine Learning Model creation using MLLIB
13. IOS Video Streaming
14. IOS Socket Client

All the above tasks are divided among the team and carried out the testing after developing the tasks.

**Risk Management**

1. Very new to IOS development, which is taking time to understand and implement features.
2. Not Enough IPhone devices for the team to develop and test the application.
3. Limited access to the Lab.
4. SPARK setup took long enough to finish tasks assigned.
5. Faced difficulties while working with SPARK and with video streaming

**References:**

<http://www.romotive.com/>

<https://developer.apple.com/library/ios/navigation/>

<http://developer.android.com/index.html>

<http://docs.opencv.org/doc/tutorials/ios/video_processing/video_processing.html>

<https://developers.google.com/android/reference/com/google/android/gms/location/DetectedActivity>

<https://github.com/googlesamples/android-play-location>