**ROBO-C**

**Moulika Gutha, Manideep Tatineni, Lema Kandula and Venu Babu Kolla**

1. **Motivation**

From generations Home security systems are mainly based on CCTV cameras. However these provide limited flexibility as these systems are controlled by using keyboard and sometimes the cameras can be badly placed. This work is motivated to improve home security requirements which will increase disabled users/seniors citizens usability and reduces cost.

Our proposed model enhances the home security management which uses RoboMe as a surveillance system which is capable of observation and responding accordingly by moving around which is designed using Machine learning techniques,Spark and Objective C.

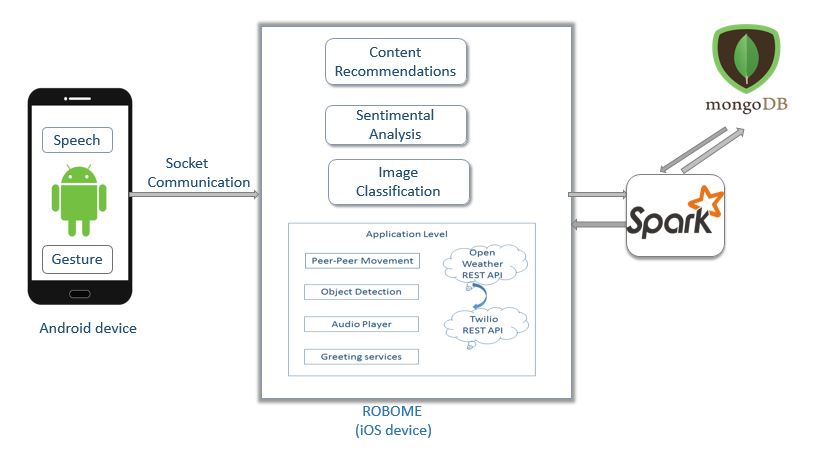
**2. Related Work**

We have many home security systems which are controlled by computer locally or over the internet, the owner can move the robot to various locations within range of the local router, take pictures and video, listen to surroundings with the on-board microphone and play sounds/music or various built-in recordings through the speaker.

Watchdog robot provides security by giving an alert to the police when there is a suspicious activity like theft. It warns the thief to move away and it also calls police if required. Inspired from this we have implemented a system in which the robot secures the home from suspicious activities like fire accident, broken windows(theft) by training the ROBO-C with the related dataset using image classification.

In our system it can respond to voice commands by using peer-to- peer connection where an Android device is communicated with iOS device which is placed inside its head. If this ROBO finds any suspicious activities it immediately sends a message/call to the user. It also recommends hospitals if finds any injury.

**3. Architecture**



ROBO-C applications has been developed in three modules

Client (Android device)

Server(iOS device)

Spark

We have developed the client module in Android devices and server module is developed using IOS application which will control ROBO-C. The client (Android devices) will communicate with the server (IPhone/IPod) through socket communication by providing the appropriate IP address and port number. User can control the robot by providing the voice -over commands through android devices which will be converted from voice to text commands and sent to the robot. If commands received by the iOS is related to Application level then iOS will utilize either built-in roboMe APIs or it will use related REST API’s to process, in order to take appropriate command.

If Commands received by the iOS is related to the Content recommendations, Sentiment Analysis and Image Classification then iOS will forward those commands to the spark system to analyze. Then spark will process that command and send the results back to iOS for taking further actions. While handling the requests related to the content recommendations the spark connects to the Monogo Labs to retrieve the user’s previous rating history which are stored already in the Mongo Labs.

**4. System Features**

ROBO-C has several features including sentiment analysis, recommending people top rated hospitals and providing home security when any suspicious activity like fire accidents or broken window is encountered in the house. The above mentioned features are done with the help of spark. Other features like gesture recognition and movements, movement based on commands like left, right, up, down.

* Peer-to-Peer Movement:

ROBO-C can be controlled by other android device that contains client application by providing IP Address and port number of the Robo-C to the client application using socket communication. Speech to text conversion is done with user giving speak commands, client application will convert speech to corresponding text and sends it to the ROBO-C.

The User can control the ROBO-C using the following commands:

* forward - to move forward
* backward - to move backwards
* left - to move left
* right - to move right
* go -to start move forward
* stop -to stop
* up - head tilts up
* down - head tilts down
* send message - to send weather report to the configured number
* sing - sings a song
* stop singing - to stop singing
* take photo - take picture
* Motion control using sensors:

Sensors in iOS device contains

1. Camera
2. Global Positioning System
3. Accelerometer
4. Proximity sensor
5. Gyroscope

Using these sensors we can control ROBO-C. There may be a path where inclinations and declinations might present. So ROBO-C will use the accelerometer data to control the movement of the robot varying the speed. Camera option is to capture pictures when the speed is varied and also to detect stop or other traffic signals. Gyroscope is for angular velocity movement where we can turn around the robot.

* Send weather details via SMS:

If user asks for weather reports then ROBO-C will send the weather reports via twilio SMS service. We have used the twilio REST API and open weather API for sending the weather details via SMS.

ROBO-C will send the following weather details :

1. Temperature in Kelvin
2. Minimum temperature in Kelvin
3. Maximum temperature in Kelvin
4. Humidity levels
5. Sky description whether its clear or cloudy.

* Singing ROBO-C:

1. Robo-C sings a song when ‘SING’ command is given.
2. Robo-C stops singing once it completes to play the song or when user says ‘STOP SINGING’.

We have used the RoboMe built-in Command Player API for making the ROBO-C sing.

* Object Recognition:

ROBO-C has the object recognition capability. When ROBO-C is moving, it will change its direction if left or right control signal is recognised and stops or starts when corresponding sign boards are recognised.

* ROBO Greetings:

ROBO-C will greet people with appropriate greetings like good morning, good afternoon or good night based on the current system timings. We have used the default RoboME API’s for developing this feature.

* Image classification:

ROBO-C is trained with many fire related, smoke related and broken window related training set. When a new image related to any suspicious activity is recognised like fire, broken window ROBO-C alerts people giving an alarm and calls the appropriate person. This way it provides security to home.

* Sentiment Analysis:

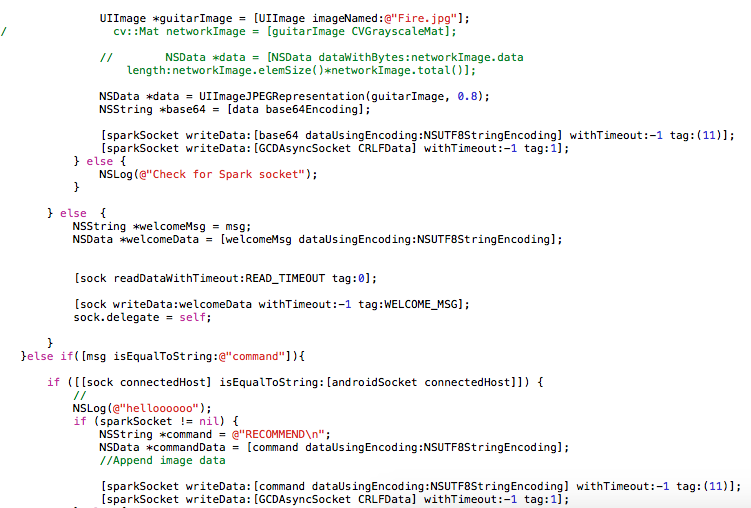
ROBO-C has a special feature to analysis sentences given by humans. For this ROBO-C analysed a wide set of data to compare with while processing the newly posed sentence. ROBO-C analyses and speaks out the state of mind of person with the help of sentence he/she has given.

* Content Recommendation:

When a people wants recommendations regarding hospitals, they can give the command recommend. ROBO-C gives the first five top rated hospitals based on the feedback. Robo-C used spark for this feature and recommendations are done based on the feedback provided by the user previously.

**5. Implementation Details**

**Peer-to-Peer based implementations**:

This feature is implemented by providing the IP address and port number to client android device from iPhone using socket communication. After establishing connection, communication is done with the help of commands so that ROBO-C takes the command and performs the corresponding action. If ROBO-C want to perform tasks related to the sentiment analysis, Hospital recommendations and Image classification then ROBO-C will connect to the spark by opening a new socket, to process the user commands as shown in the below screenshot:

**Spark based implementations**:

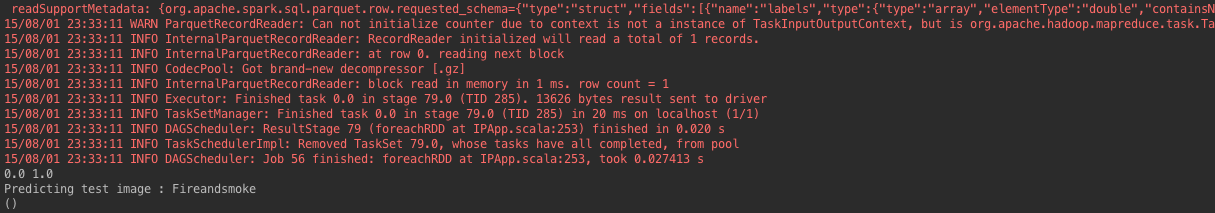
Spark played a vital role in sentiment analysis, recommending hospitals based on previous feedbacks and rating of the people and image classification. Sentiment analysis takes a dataset, processes it and develops a model so that if any new sentence is posed, it understands the sentence using NLP techniques, and speaks out the feeling or state of mind of the person.Hospital recommendations are based on the previous opinion and ratings of the user. For this, ROBO-C connects to MongoDB to get the previous experience, opinion and ratings of the user and finds out the top rated hospitals for him/her. Image classification considers

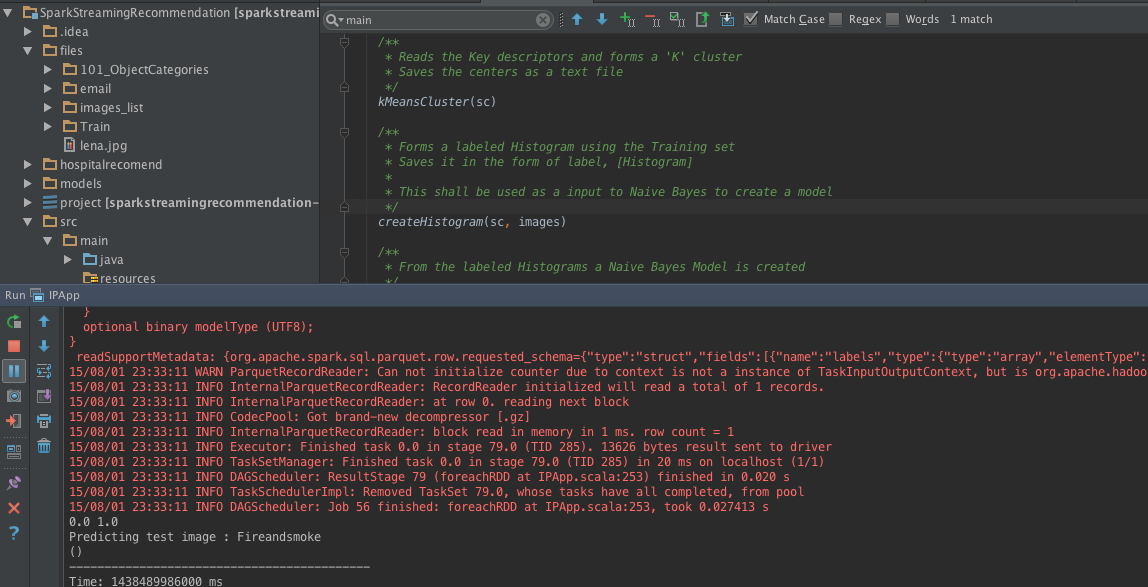
Screenshots for the image classification:

Input image from the iOS:

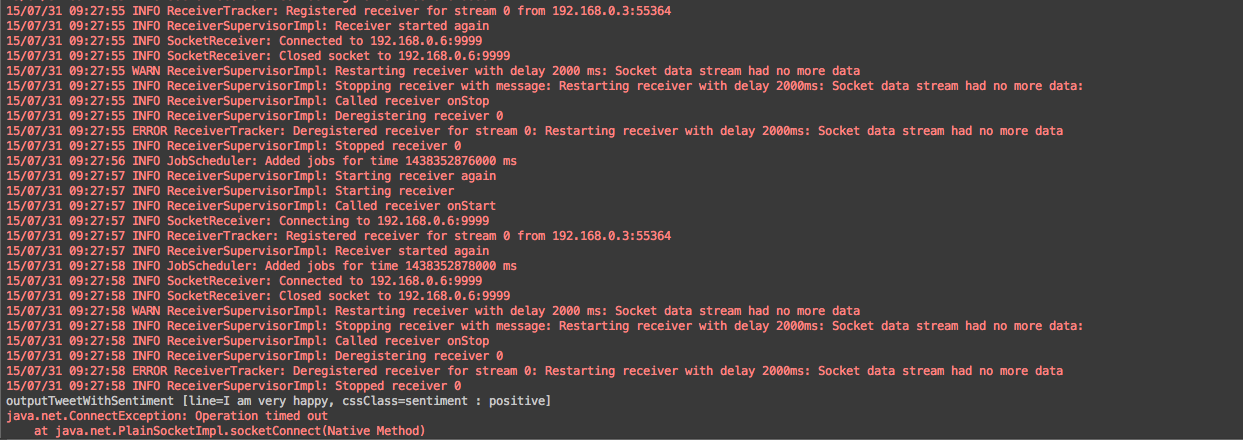


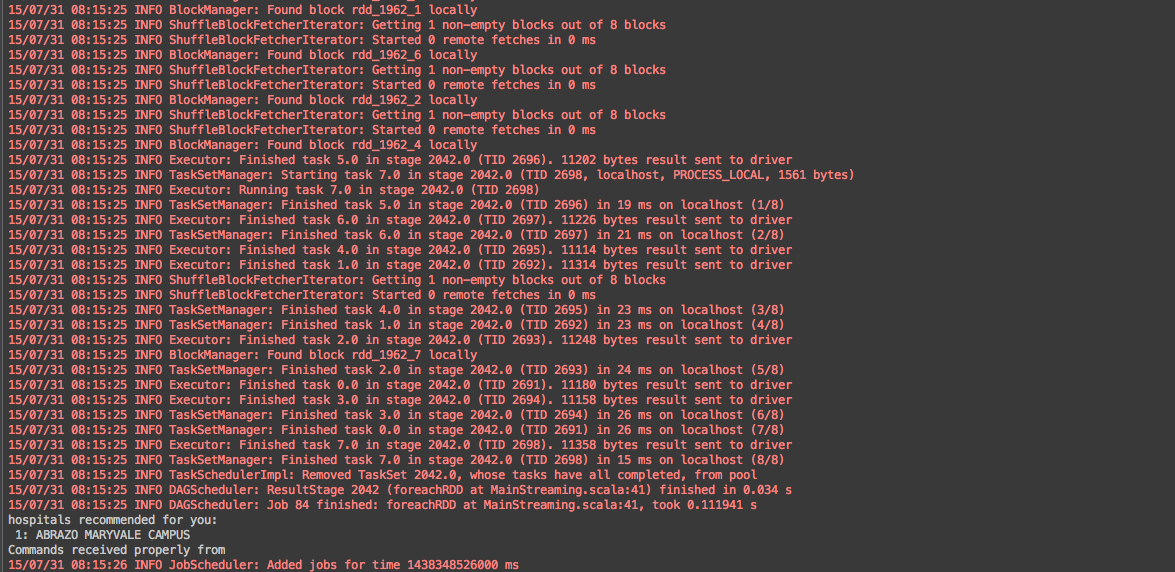
Output screenshot after image classification:





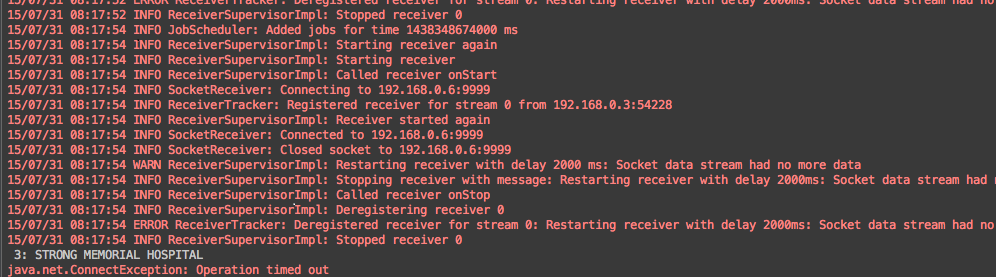
Screenshot for sentiment analysis:



Screenshot for content recommendation: 1 

Screenshot shows recommending first hospital when person wants top rated hospital recommendation.

Screenshot for content recommendation: 2



This recommendation shows second recommended hospital from the top rated hospitals list. Similarly ROBO-C recommends next three top rated hospitals. So ROBO-C recommends five top rated hospitals when requested.

**6. Results/evaluations**

This ROBO-C project was implemented with the help of NLP, machine learning techniques, google API, open weather API and REST APIs services. This project has multiple parts based on the services and methods. ROBO-C’s brain is iPhone which performs various actions according to the commands provided by humans through android device. So a socket connection is established for communication between ROBO-C and android device. Some of the commands to ROBO-C are

Left: ROBO-C moves left

Right: ROBO-C turns right

Up: ROBO-C tilts its head up

Down: ROBO-C tilts its head down

Song: ROBO-C sings a song

Recommend: ROBO-C recommends top rated hospitals

ROBO-C sends and speaks out weather details when he says that he is going to particular city. ROBO-C sends city name to open weather API and gets the weather details, speaks out and sends the same to the person

Sentiment Analysis:When any sentence is spoken out by people, ROBO-C analyses the sentence, performs sentiment analysis and speaks out the state of the person whether he or she is being positive or negative at that moment based on the sentence that is delivered by them

Image Classification:When ROBO-C encounters fire or broken windows, it does image classification and detects the image and informs the owner or person through call.

ROBO-C uses spark for sentiment analysis, recommendations and image classification.

**7. Conclusion**

In this project, we got better results when executing by developing the features using Machine learning algorithms, objective C language and Spark services while processing the commands. Even if some text is missed in the search query, NLP mechanism is able to identify the missing part of the sentence and summarize the content. Spark service can recommend some suggestions to the user effectively. Objective C language is used to develop movement in the ROBO-C efficiently.

Machine learning algorithms provided optimized solutions in sentiment analysis and image classification process. Watson speech to text service to make the application more user interactive, and also to give commands to ROBO-C for movement and other functions.

**8.Future Work**

The current image classification we used considers only few datasets belonging to few categories which are mainly related to home security. We are planning to upgrade this dataset to widely related data and on huge amount of information. We are also planning to implement sentimental analysis on some expressions like fear and tensed emotions.

The recommendations made by our machine learning algorithm gives as per the previous information provided by user and we are planning it to develop further like giving the nearest top rated hospitals.

**9. References**

[1] RoboMe - <http://www.gizmag.com/romo-smartphone-robot/20139/>

[2] Twilio API [https://www.twilio.com/user/account/developer-tools/api-explorer/message-create](https://www.twilio.com/user/account/developer-tools/api-explorer)

[3] OpenWeather API - <http://openweathermap.org/api>

[4] Text to speech API - <https://cloud.google.com/translate/v2/using_rest>

[5] NLP - <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1598694>

[6] SPARK - <https://spark.apache.org/docs/latest/programming-guide.html#external-datasets>

[7] Sentiment Analysis - <http://stackoverflow.com/questions/25908536/twitter-sentiment-analysis-using-naive-bayes-in-apache-spark>

[8] Spark Recommendation - <http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html>

[9] Project Youtube video URL - <https://www.youtube.com/watch?v=RfRR_Q3k718>