

Robot-Assist

Big Data Analytics and Applications Project Final Document

Project Group #10

Vaishnavi Aienampudi #01

Sarika Bommavaram #10

Phanideep Salapaka #40

Gouri Priya Vangavargu #45

**Motivation**

We can do many things with the help of robots. They can help people in many fields like doing some household work, helping in medical field, technological field etc.. The idea of this project comes from the question “Can a robot be a personal assistant to a disabled person?” We believe that humans can do this work but at some point of time or in some cases they become lazy or may not be able to do some work. So if we think of a robot, it can help humans to complete their work. This motivated us to make our robot as a personal assistant for a disabled person. Not only as a personal assistant doing things but if we can be able to integrate a big data platform like Spark, it can add more value. This integration of big data platform not only adds value but it can also be used in many real world applications. For example consider, the user can give his personal ratings to restaurants, books, movies etc.. Our robot when connected to Spark, it stores the users personal ratings, and it will recommend the movies, books, restaurants based on his ratings. Our robot mainly helps to automate things for the disabled persons. Consider an example, for disabled persons, it would be difficult to make a call by searching contacts in their phone. Our robot-assist will help them by making their work easy by taking speech command dynamically. The user can direct give the command as “call John”. This will automatically search our contacts and connects a call to John. This is how, it will help physically challenged users.

**Goals and Objectives**

Humans can do their work by themselves. But if we consider disabled persons, it would be difficult for them to do their work without depending on anybody. If we can make a robot that helps disabled people they can do their work with the help of these robots. Our main objective is to build application software that will help the robot to learn about personal assistant and how it works. It should also be able to connect to Spark platform and use the big data to help them. It should be able to help disabled people to do their work in every aspect.

**Related Work**

Kompai Robot:

This robot is developed by Robosoft that is built to help physically challenged and aged people. We can communicate with the robot through our regular language. It also helps individuals to remember gatherings, plays music, and fills in as a videoconference framework for clients to chat with their specialists, for instance.

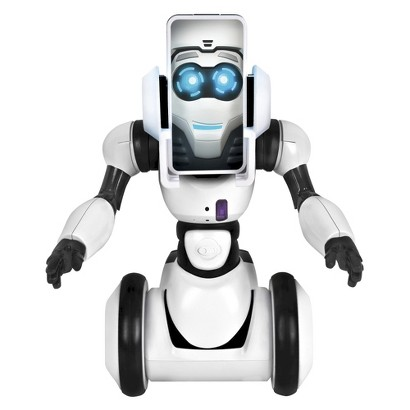
NAO:

With this application, NAO is fit for reminding you, or others in your family, to do undertakings, stay informed concerning a certain measure of time, and alarm you to general updates through your own customized voice recording.

We want to include some more features and to develop using robome such that it is used for assistance of a disabled person. i.e, the above features will be developed in perspective of a disabled person.

There are many robots that will help the physically challenged people but there is no robot that will connect to big data platform and make use of the big data to help in recommending them. So, our project aims to connect to big data platform Spark and also helps the disabled people in doing their things.

**System Architecture**



Command

User

Sends command

Acts as brain

Sends command

Acts as a server

Stores data

Fetches data

Does analysis

Here, the user can give commands directly to the android device which acts as a client and a controller. The iPhone being the server lets the client connect to it and takes the command from the android controller and passes it to the robot. The robot will perform the action based on the command. The android also passes the command to the Spark which uses the data from the mongo lab to do analysis part. The results will be passed to the iPhone again. This is the architecture of our system.

**System Features**

These are the following featureswe developed for our robot-assist.

1. **Face Recognition:**This feature is implemented using Kairos SDK. First the application is trained with sample data. And whenever it captures the image, it will compare with the trained data. If it matches it will give a greeting message with their name. If it does not matches, it will treat them as a new user.
2. **Simple Questions:** This feature is implemented using NLP and speech API. It takes the questions in the form of speech and give out answers in the same form. Our robot can answer nearly 20-30 questions.
3. **Music:** If the user gives the command as music, it will play the music. If the user wants, he can pause, replay it again. If he wants to completely stop he can stop it using Stop command.
4. **Dynamic Call:** This will take the name of the user directly and searches the address book for that name and connects a call to them. This feature uses address book framework.
5. **Simple Movements:** The user can give somple commands to move left, right, forward,backward etc..
6. **Gesture Recogntion:** The user can make gesture movement, so that the robot can move according to the gesture. It will also speak out the direction in which it is moving.
7. **Weather:** The user can directly give any city name as the command and the robot will speak out the weather conditions of that particular place. It uses text-to-speech API and Open Weather API for this feature.
8. **Color Detection:** Our robot will also deetct colors and perform action on recognizing that particular color. This is intended to detect traffic lights signals.
9. **Reminder:** We can also set a reminder using our robot. It will access reminders from our phone and will set a reminder.
10. **Screen Sharing:** Using our robot, the user can see the screen shared by the robot. For disabled people this is one of the interesting feature.
11. **Image recognition and Classifcation using Spark:** With this feature, the images can be detected and classified based on the training data set. This uses Apache spark which acts as a client that connects to the iOS device which is a server.
12. **Books Recommendation:** With this feature, the user can get recommended ratings based on the user’s personal ratings.
13. **Twitter Streaming and Sentimental Analysis:** This feature is developed by collecting tweets on the particular keyword and storing them in Mongo Lab. Spark fetches the data from the MongoLab for doing sentimental analysis. It does this sentimental analysis for every tweet and will display the single result whether the overall tweets are positve, negative or neutral. It does the aggregation of all the tweets.

|  |  |
| --- | --- |
| Platforms Used | XCode, IntelliJ |
| Languages Used | Objective C, OpenCV, Scala |
| Data Base Used | MongoLab |
| API’s Used | Text-to-speech, Speech-to-text, Kairos SDK, Open Weather API |

**Application Specifications:**

**Existing Services:**

**1)** **Weather API:**

This API provides the service for weather. It takes input as name of the city and gives weather details like temperature, minimum temperature, maximum temperature etc...

<http://api.openweathermap.org/data/2.5/weather?q=Kansas>

**2) Text to Speech API:**

High-quality text-to-speech API. Adding parameter & return\_url=1 to the URL will give you the

URL to generated MP3 instead.

http://tts-api.com/tts.mp3?q=hello

**3) Google Maps SDK:**

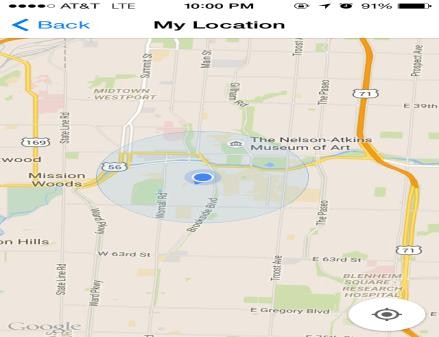
The use of google maps API is to know the current location. This is mainly used when user visits a new place, then user uses the API and determines his current location.

4)  **Speech** **to** **Text**

We used Google’s speech-to-text API. It takes input as speech message and gives output as text message.

**Implementation Details And Results with Screen Shots**

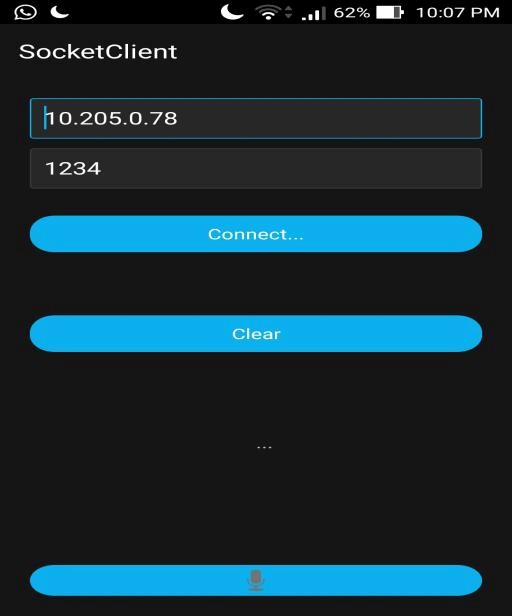
1. **Using** **Google** **Maps** **SDK**:

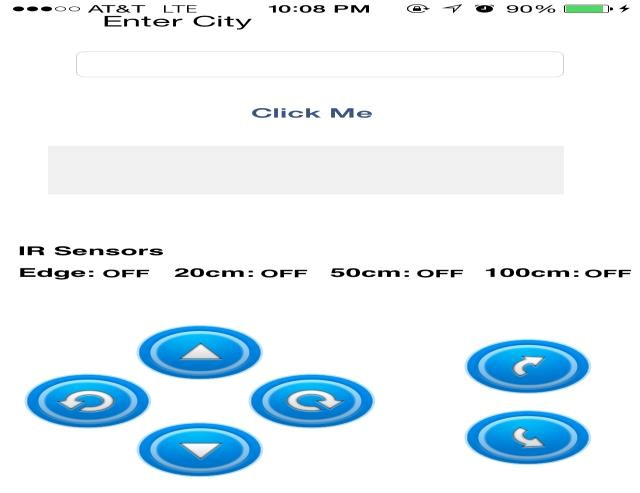


* Using google maps SDK we are determining the current location of the robot.

**2. Speak** **Weather:**

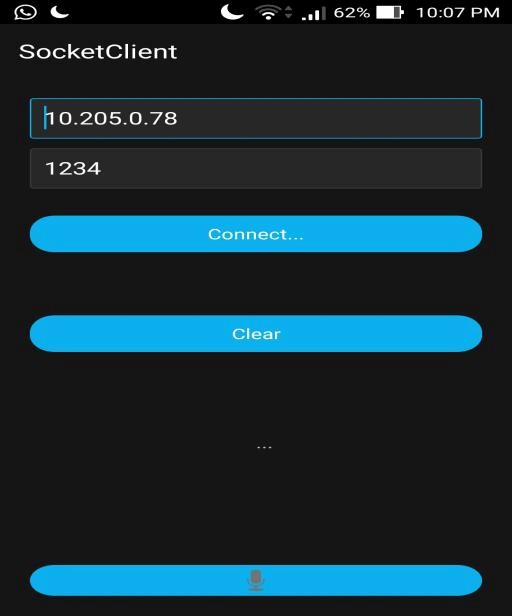
* This is the android controller.
* It uses socket programming to connect to the iOS device.
* This acts as a controller to the robot.
* The user can give commands directly to the robot using this device.





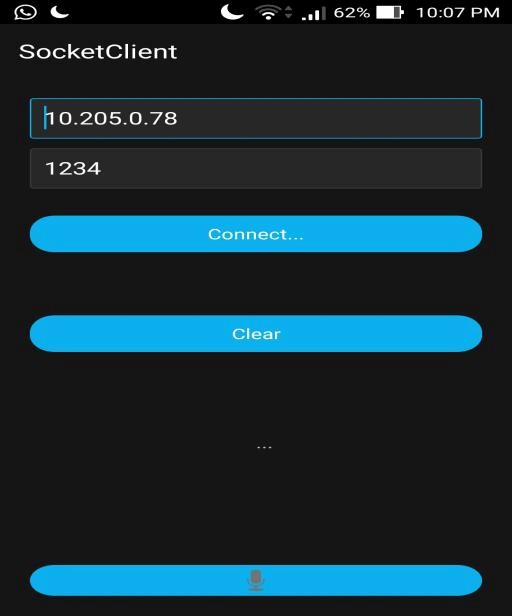
* The user enters the name of the city he wants to find the weather.
* This uses open weather API, speech-to-text API and text-to-speech API.
* It will give the results in the form of speech messages.

**3. Robo** **-** **Me** **Commands:**



* The user can give simple commands using the mic button.
* It will send these commands to the server which is iOS device.
* The robot will perform actions based on the commands given.

**4. Making** **a** **Call:**



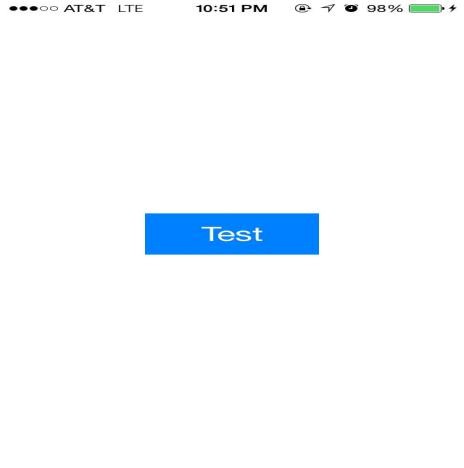
* This is the android controller where the user can click on the mic button. If he gives command as call which is a voice text, the command will be received by the robo me application.
* On receiving the command “call”, a call will be automatically made to the emergency number.
* The user can also give command as “call name”.
* This feature is implemented with the help of Address book framework. It takes input as voice command from the android controller and makes a call to the desired person.

5. **Clicking** **a** **photo**

* The user can open the camera app to click photos.
* He can change camera from front or back.
* He can also record the videos.
* The photos and videos will be saved to our photo library.

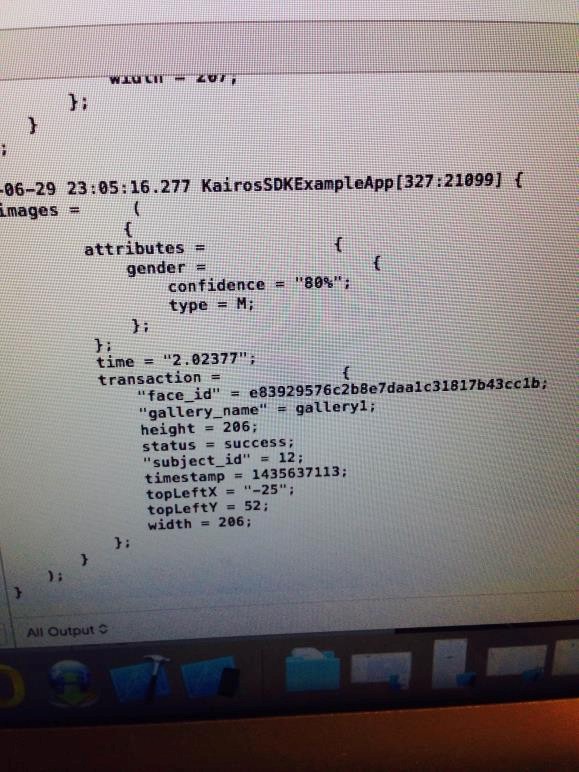
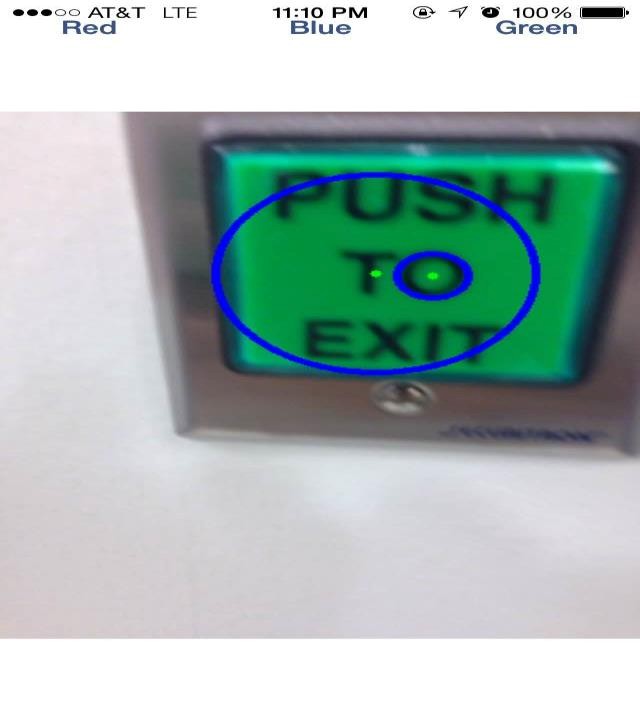
**6. Face** **Recognition**

* This feature is implemented with the help of Kairos SDK.
* First it is trained with the sample data.
* Whenever it captures the face, it matches with the trained data and gives the greeting message.
* It it matches with the trained data, it will give the name of the person.
* Otherwise, it will welcome them as a new user.



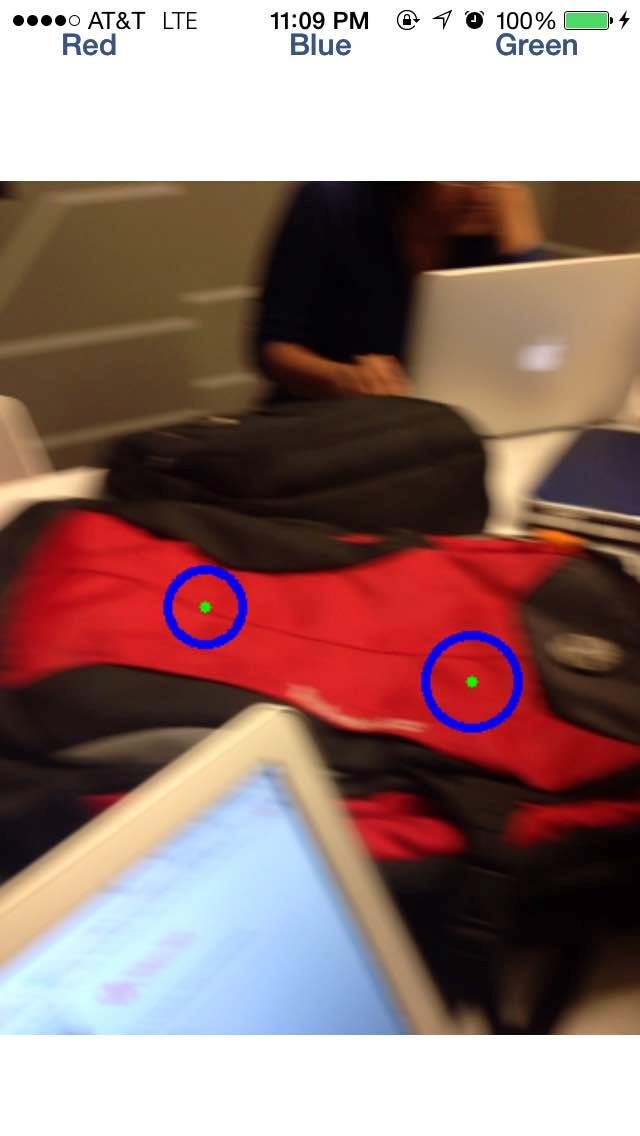


The figure shows the similarity score as 80% which shows the matching of images.

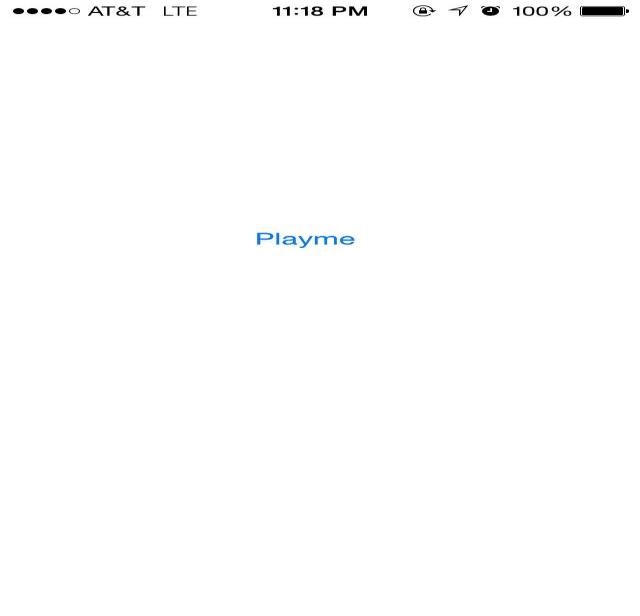


**7. Color Detection and following Traffic Signals:**

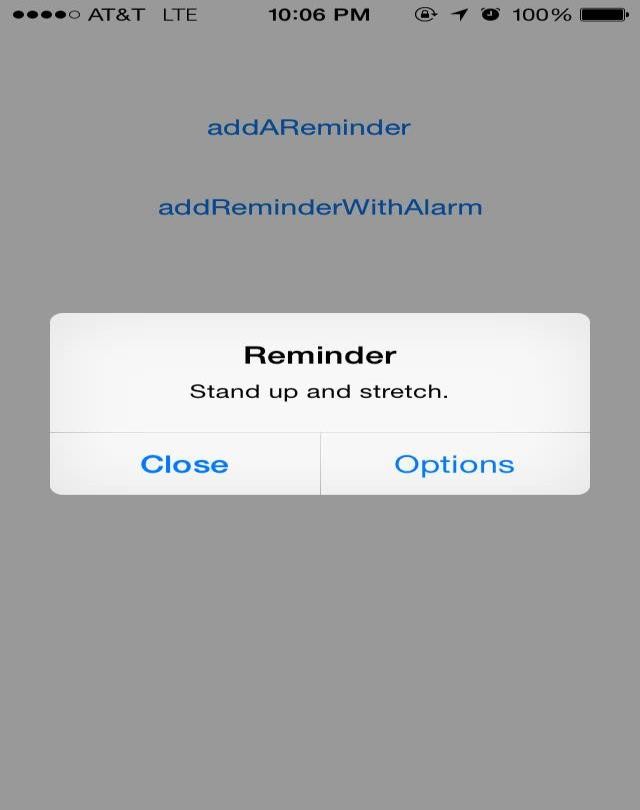
* From the figure, we can see the circle formed on detecting green color.
* We did this with the help of image processing using OpenCV.
* This concept is mainly used to follow traffic signals.
* We can see the below screen shot that forms circles on detecting red color.
  + This feature helps the robot to follow the traffic signals. The robot will function in the following way:



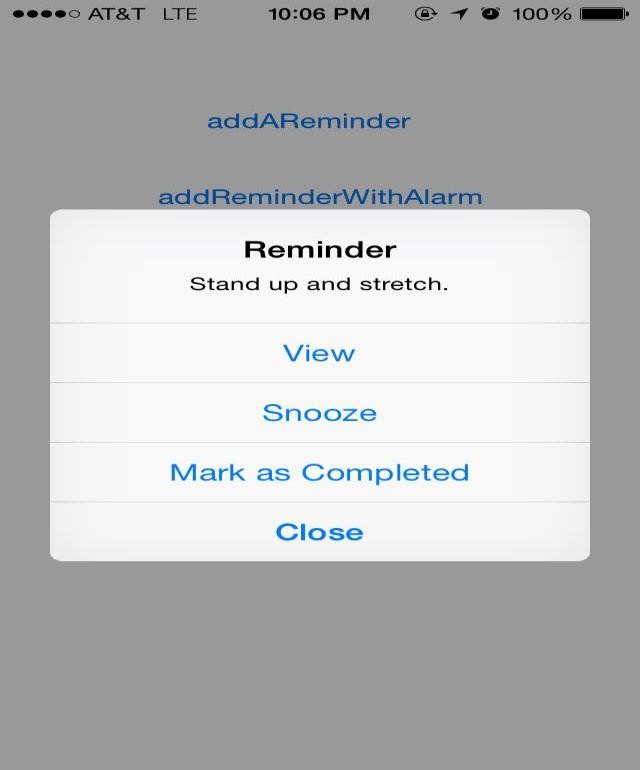
* If the robot detects a red color, it stops its movement.
* If the robot detects a green color, it moves with its maximum speed.
* If the robot detects an orange color, it moves with its minimum speed.

**8. Playing** **Music:**

* We implemented this feature of playing music on giving command “sing a song”.
* We used “AudioToolBox” framework for this feature.
* We can add music files and specify the path for the music file.

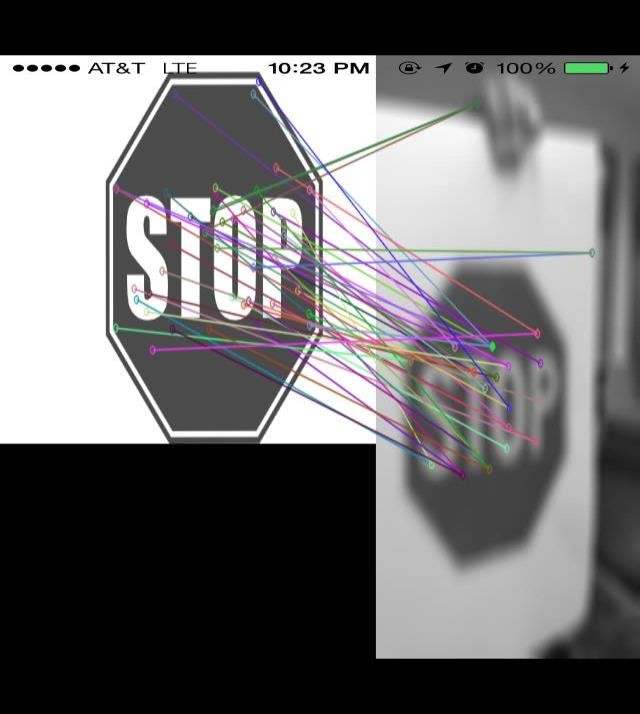
**9. Reminder:**

* When the user clicks on options, he can access his reminder list.
* It will give the following options
  + View
  + Snooze
  + Mark as completed
  + Close.

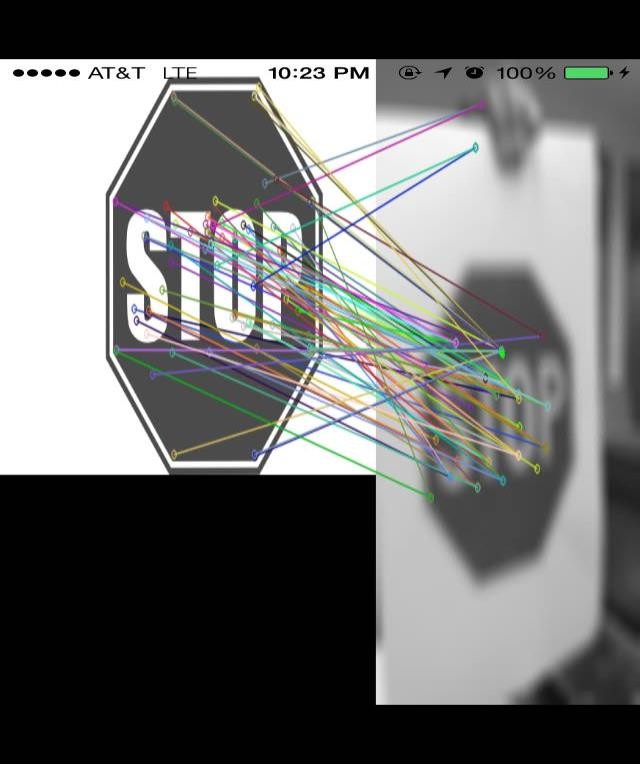


* These are the options displayed when the user clicks on Options button.
* View –He can view his reminders list.
* Snooze- He can postpone it some time.
* Mark as Completed – It will remove the alarm
* Close- Closes the application.

**12. Object Recognition**

****

* For the object to be recognized, we have to first store the image.
* Whenever it captures the image, it matches with the already stored image.
* From the figure we can see that the features being extracted.

****

* The extracted features will be matched to the stored image.
* Here we can see that the image that has been already stored is very clear and the image that is being captured is somewhat blurred.

**10. Question and Answers**

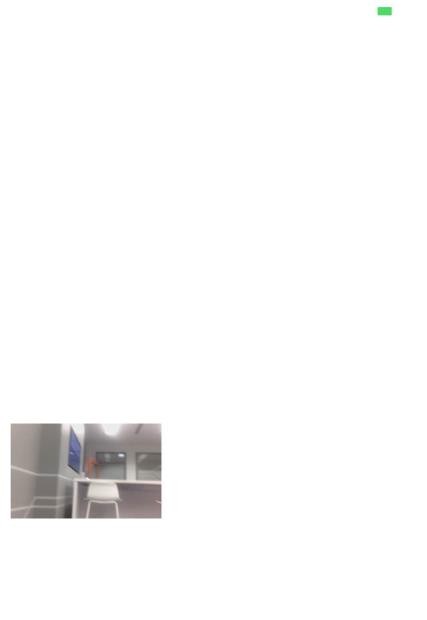
Our robot will respond to simple questions like what is your name?, How are you ?, etc.. We implemented this feature with the help of NLP, and text to speech API. For this, we are using NLP to get the tags and tokens and we integrated a text-to-speech API for answering the questions.

For some questions like where are you? We are determining the current location of the robot by using google maps SDK.

**11. Gesture Recognition**

Whenever it recognizes a gesture movement, it will change its direction based on the gesture. Additionally it will speak out its direction. For example, it recognizes a gesture movement of left side, it will change its direction to the left and will say I’m turning left.

**12. Screen Sharing:**



* This is one of the interesting feature for the physically challenged people because, they can’t go everywhere. So, with the help of this feature, the robots can go and share the screen with them so that they can know and see what’s happening there.
* Here we can see a small picture down which shows the live capture being captured.
* During Live capture user can click the photos if he wants.

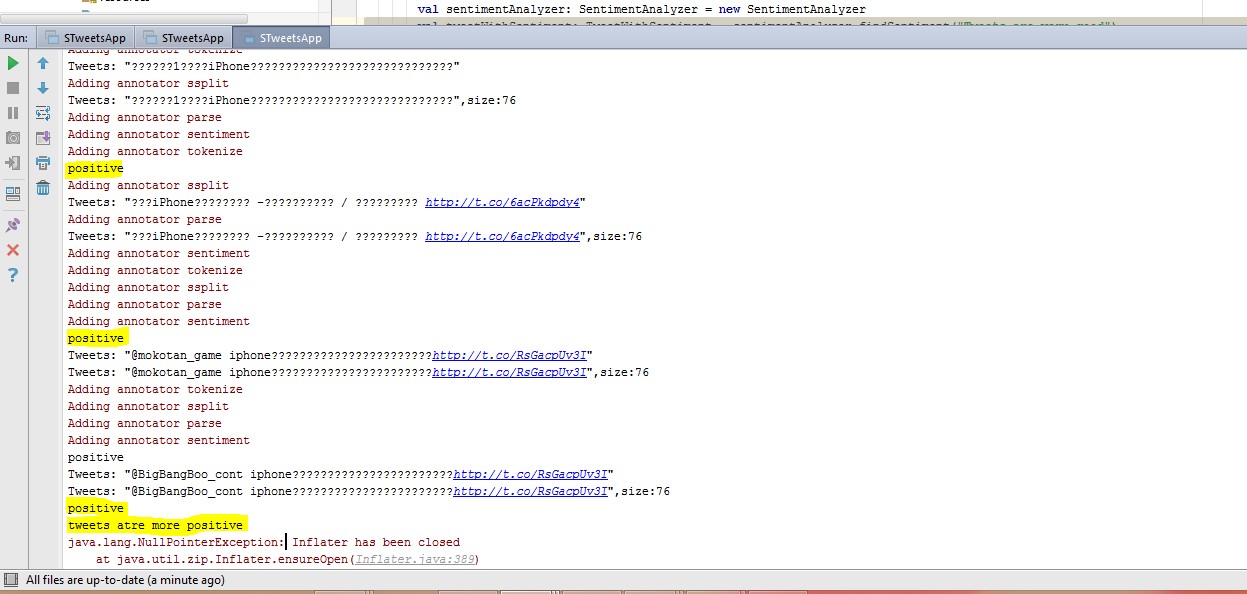
**13. Book Recommendations**

Here we can see that result is the recommendations of movies. It takes the books data set which has three data sets books, ratings, and users. We also give a personal ratings for a particular user. So that we can recommend books for that user. This feature is implemented with the help of Machine Learning library with the help of Spark. We can see the result in the screen shot below.

Here the Spark machine works as a client and iOS device acts as a Server. The command is sent from the android device as recommendations. The output will be displayed by the robot as a speech command.



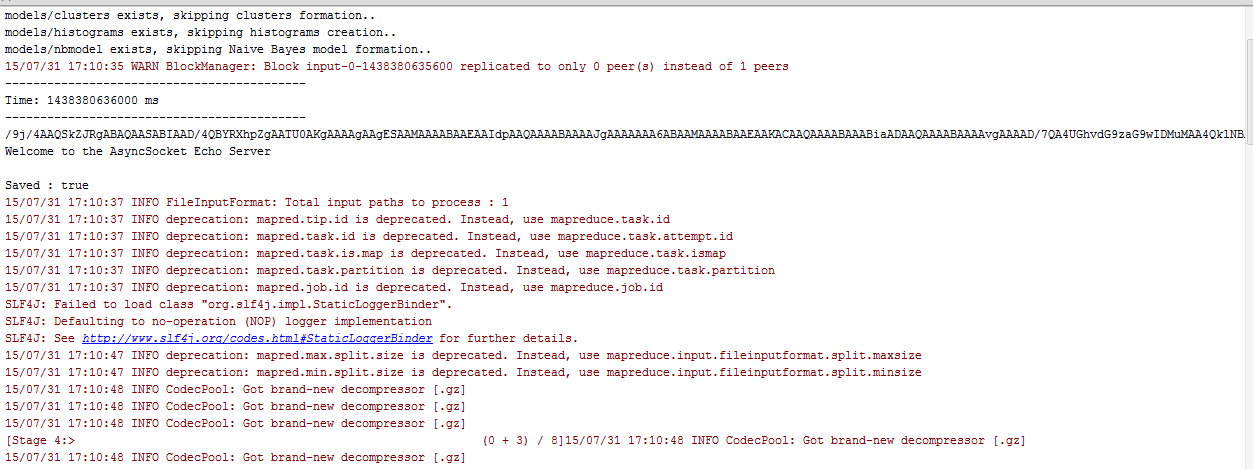
**14. Sentimental Analysis of Streaming Data with the help of NLP**



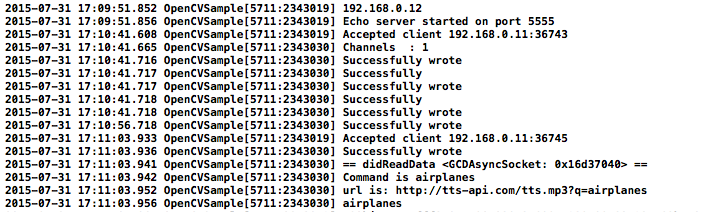
Here we collected the tweets for a particular keyword “iPhone” using twitter streaming API. We stored all the streamed tweets in the Mongo Lab. For doing Sentimental Analysis, we are again retrieving the tweets from the Mongo Lab and performing sentimental analysis on them. It analyzes each and every tweet and give the result. Then the result for all the tweets are aggregated and the final result is given as whether the context of the tweets are positive or negative or neutral. This result is passed to the iOS program using iOS connector.Scala class and there we are passing it to the text-to-Speech API.

**15. Image Classification with iOS as a server:**

Here the iOS device acts as a server and the spark machine acts as a client. First the socket program is executed so that the server starts. Then if we start the spark machine, it connects to the server and receives the image from the server. The client does the image classification and sends the result back to the server. For sending this result back to the server we are using iOS connector. Scala class.



Here you can see the image being captured from the iOS device is being save here. Now the classification is done on spark machine and the result is displayed.



These are the logs generated on server side. You can clearly see the airplanes command that has been sent from the client. This is how we are connecting iOS, Spark and and sending the result back to the iOS device.

**Individual Performance:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature NO | Feature Name | Done by | Hours Worked | Status |
| 1. | Face Recognition using Kairos SDK | Sarika, Vaishnavi, Gouri, Phanideep | 30 | Done |
| 2. | Appending greeting message to face recognition | Vaishnavi | 2 | Done |
| 3. | Deriving name out of face recognition and appending it to subject | Sarika | 2 | Done |
| 4. | Adding Music feature | Phanideep, Gouri | 10 | Done |
| 5 | Making a simple call | Phanideep, Gouri | 14 | Done |
| 6. | Using NLP to get tags and tokens | Vaishnavi | 4 | Done |
| 7. | Adding Dynamic feature to the simple call by taking names dynamicaly | Vaishnavi, Sarika | 20 | Done |
| 8. | Using google maps SDK to determine the current location | Vaishnavi | 4 | Done |
| 9. | Designing android controller so that, it can take commands directly as a speech message from the android device | Vaishnavi, Sarika (Done as a part of Lab assignment) | 5 | Done |
| 10. | Making simple movement | Phanideep, Gouri | 4 | Done |
| 11. | Movement based on acceleration | Phanideep, Gouri | 12 | Done |
| 12. | Movement according to color detection. | Vaishnavi, Sarika | 12 | Done |
| 13. | Objection Recognition | Sarika | 20 | Done |
| 14. | Movement based on gesture | Phanideep, Gouri | 4 | Done |
| 15. | Appending Speech messages to the gesture movement | Vaishanvi | 4 |  |
| 16. | Using a rest service to determine weather of a particular city | Vaishnavi, Phanideep | 6 | Done |
| 17. | Using text-to-speech and speech-to-text API | Sarika, Gouri | 8 | Done |
| 18. | Integrating it with the main project | Gouri | 6 | Done |
| 19. | Using NLP and text-to-speech API, to answer simple questions | Vaishnavi, Sarika | 10 | Done |
| 20. | Adding a camera feature to our project | Phanideep | 4 | Done |
| 21. | Reminder feature | Phanideep, Gouri | 12 | Done |
| 22. | Screen Sharing | Sarika, Phanideep | 10 | Done |
| 23 | Books recommendation using Spark | Vaishnavi, Sarika | 20 | Done |
| 24. | Streaming tweets using a search keyword | Gouri, Phanideep | 20 | Done |
| 25. | Doing Sentimental Analysis on the streaming tweets | Sarika | 10 | Done |
| 26. | Appending the result to the iOS device | Sarika | 12 | Done |
| 27. | Getting results from books recommendation and passing the result to the iOS device | Vaishnavi, Sarika | 12 | Done |
| 28. | Image Classification using spark | Vaishnavi | 20 | Done |
| 29. | Passing the result to the iOS device | Vaishnavi | 5 | Done |
| 30. | Integrating all the above features | Vaishnavi, Sarika, Gouri, Phanideep | 30 | Done |

**Conclusion:**

These are the features we developed for our project. By developing this application for our robot, it will learn to perform actions so, that it can be able to help physically challenged people in many ways in dong their work easily. Our project also connects to the big data platform which uses the large data set to analyze and give recommendations for them. We have trained our robot with various image datasets in such a way that it will speak out the image name when it is being tested with any specific image in the trained set. Robot is also trained in such a way that it can give the sentimental analysis by taking streaming twitter data.

**Future Work:**

Still we can add more features and make many features completely automated. For the big data analysis, we can give our personal ratings directly from android controller so that it will store them in Mongo Lab and dynamically retrieve them for recommendations. Also, using face recognition we can ensure security. Like, if we train with faces for only authorized users, it will act or perform actions only for them. It can completely ignore others. As there is a good scope for using both Spark and robots, we can really create many interesting features so that it can help physically challenged people in many ways.

**References:**

* Tutorial for Objective C language - <http://www.tutorialspoint.com/objective_c/>
* Openweathermap API - <http://openweathermap.org/api>
* Text to speech API - <http://tts-api.com/>
* Googlemaps SDK - <https://developers.google.com/maps/documentation/ios/>
* OpenCV - <http://opencv.org/>
* RoboMe SDK - <https://github.com/WowWeeLabs/RoboMe-iOS-SDK>

**Video Link:**

https://www.youtube.com/watch?v=jWGi8TD7KHQ