Ro-De:specially abled friend.

Guru Teja Mannava[#1], Sai Bharath Attaluri[#3], Venkatesh Prasad Kalluri[#3], Surya Narayana Botcha[#4]

CS560: Big data apps and analytics

Dr. Yugyung Lee

University of Missouri-Kansas City

1. MOTIVATION:

In US there are approximately 1000000 people aged over 5 are unable to hear and more than half people are aged over 65 who are suffering from hearing problems. These people have been using some kind of things to assist them like machines and hearing dogs.These hearing dogs they are trained to listen all the important sounds and convey the message to the person. This concept has enlightened us to develop a robot that will assist these people and also provides more services than what a hearing dog can do. Ultimately we are succeeded in developing a robot which not only assists but also works as a security system for the person. Our robot can perform operations that are performed by hearing dog and also performs operations like speech to text, text to speech, two-way communication using Natural language processing and Our robot has a crash recovery function also.We have developed our Ro-De in IOS where we used spark to process our data stored in the database. Our Ro-De is just like a friend. Our Ro-De listens to what speaker says and then produces the data in the form of text or images based on the user requirement.It also recognizes the things using image classification and also suggests book to user by content recommendation which is developed in spark.It directly communicates with the user. It acts as a complete assistant to the deaf person.

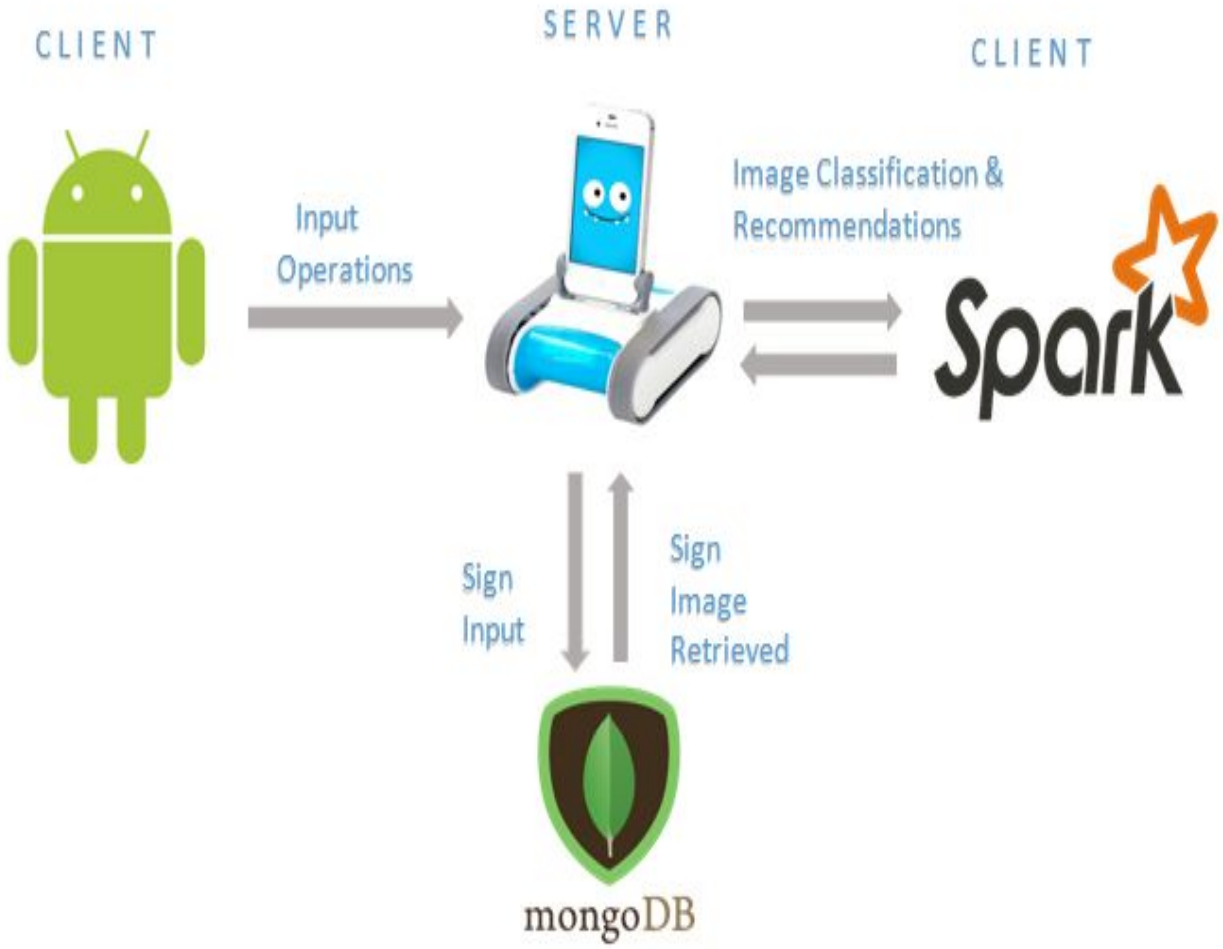
2. RELATED WORK:

There are several works which are specially designed to help specially abled people. Like there some robots which continuously monitor the patient health as a nurse.

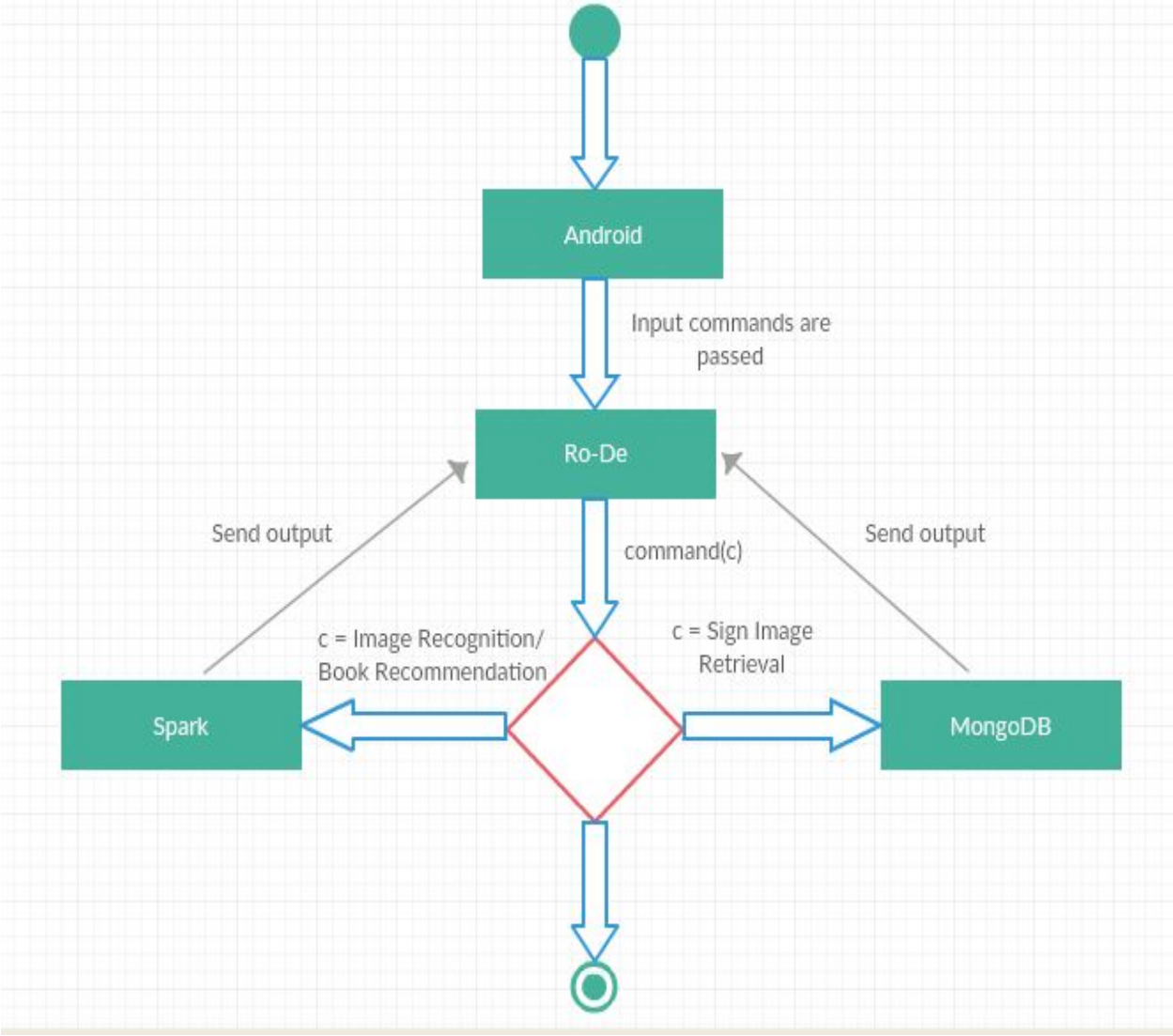
Aiko: A Japanese Robot that can communicate with deaf using sign language, there are several robots that help disabled people to help their abilities. There are no several works that actually detect a sound and recognize it.

3. ARCHITECTURE:

Android client connects and sends commands to Ro-De which is server, Ro-De performs actions based on the commands. If user gives command to get Book Recommendation/Image recognition it connects to Spark client and gets the result back .If user gives command to retrieve sign images , Ro-De connects with Mongo-Db to get Sign images of that command.



Activity Diagram :



4. SYSTEM FEATURES AND IMPLEMENTATION DETAILS:

We have implemented several features in Ro-De. Following are the features of Ro-De:

Peer-to-peer Movements:

Ro-De can be controlled by any client which is android. Client can be connected to Ro-De with Ip address and port address of Ro-De. After connecting client can send any command, based on the command robo moves. Speech to text service is used in the client, to get speech input from user and send text output to server Ro-De.

Basic movement commands:

Forward - move forward
Back - moves back
Left - turn left
Right - turn right
Slow - reduce its speed.
Fast - increases its speed.
Flash on- switch on its Flash.
Flash off- switch off its Flash.

Object Detection:

Ro-De can detects other robots and tries to communicate with it. By using front camera Ro-De continuously tries to recognize the objects that are in front of it. It can also detect basic household items such as lays and Powerade drink. Whenever it sees a "Stop" symbol, it detects it as "STOP" sign and performs stop action. Similarly when it sees "GO" or "LEFT" symbol, it performs "move forward" action and "turn left" action.

Color detection:

Ro-De has the ability to detect a color and notifies the user which color it has detected. whenever it detects the red color it notifies the user.

Crash Recovery:

Our Ro-De comes in built-in crash recovery feature, which has the ability to return to original position when it crashes.Here we are using accelerometer data to check whether the Ro-De is in crashed state or not. If yes, then it tilts its head to maximum angle possible(here 70), and then it gives a fast forward movement to come back to original position.

```
//get up in excess declination
if ((a <= 0.35 & a >= -0.35)& ((b <= 1.0 & b >= -0.99))& (c >=0.10 & c<= 2.10)){

    //light pulse
    [self.Romo3.LEDs pulseWithPeriod:-1.0 direction:RMCoreLEDPulseDirectionUpAndDown];
    //[self.Romo3 tiltWithMotorPower:1.0];
    [self.Romo3 tiltToAngle:120 completion:^(BOOL success) {
        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, 2 * NSEC_PER_SEC), dispatch_get_main_queue(), ^{
            self.Romo.expression=RMCharacterExpressionHoldingBreath;
            [self.Romo3 turnByAngle:0 withRadius:0.0 completion:^(BOOL success, float heading) {
                if (success) {
                    [self.Romo3 driveForwardWithSpeed:1.5];
                    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, 1 * NSEC_PER_SEC), dispatch_get_main_queue
                        [self.Romo3 stopDriving];
                        [mManager stopAccelerometerUpdates];
                    });
                }
            }];
```

Move to specified distance:

In this project we have included a feature which has the ability to move to a specified distance. When user gives input as 'move 3 meters' then Ro-De moves to that specified distance.We have done this by pausing the main execution by some time, which ultimately results in stopping at the distance. For example, if Ro-De speed is 0.6 m/s , then to cover 3 meters, the stop time must be equal to 5 seconds.

```
else if ([string isEqualToString:@"1 METER"] || [string isEqualToString:@"MOVE 1 METER"]) {
    [self.Romo3 driveForwardWithSpeed:0.6];
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, 1.667 * NSEC_PER_SEC), dispatch_get_main_queue(), ^{
        [self.Romo3 stopDriving];
    });
}
else if ([string isEqualToString:@"2 METERS"] || [string isEqualToString:@"MOVE 2 METERS"]) {
    [self.Romo3 driveForwardWithSpeed:0.6];
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, 1.667 * NSEC_PER_SEC), dispatch_get_main_queue(), ^{
        [self.Romo3 stopDriving];
    });
}
else if ([string isEqualToString:@"3 METERS"] || [string isEqualToString:@"MOVE 3 METERS"]) {
    [self.Romo3 driveForwardWithSpeed:0.6];
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, 5 * NSEC_PER_SEC), dispatch_get_main_queue(), ^{
        [self.Romo3 stopDriving];
    });
}
else if ([string isEqualToString:@"5 METERS"] || [string isEqualToString:@"MOVE 5 METERS"]) {
    [self.Romo3 driveForwardWithSpeed:0.6];
    dispatch_after(dispatch_time(DISPATCH_TIME_NOW, 8.33 * NSEC_PER_SEC), dispatch_get_main_queue(), ^{
        [self.Romo3 stopDriving];
    });
}
```

Sign Images for given speech:

For this concept we use Mongodb which is a NOSQL database . It is a document oriented storage and stores in the form of BSON format.It convert all formats into bson which makes it easy to integrate certain types of data belonging to applications. We use MongoLab in our project.MongoLab is a cloud database hosting Mongodb platform and is fastest growing in the world.

Along with speech-to-text and text-to-speech our robot provides images as output for the speech input.We store our images into mongodb. By using scala we convert the images into string format and push the data to mongolab. Here we use base64 to convert image to string. While retrieving the images the string data is converted back to the image format. We retrieve

the images in the iphone. So when the user sets it in the image retrieval mode and when a person says something, then based on the command or sentence it provides the respective images.

When user gives input as “Can you show Sign Images” then Ro-De gives reply as “Yes, what can i show” . Then as per user requested word Ro-De gets the Sign image from Mongo-db and displays back to user.

Image Recognition:

We have done image classification in Spark. Which plays a crucial role in determining the image .Spark generates a featured model based on the input data sets provided. If user gives input as “Recognize Image” then based on the featured model generated it will specify which class it belongs to.

Book recommendation:

We have also implemented book recommendations using Spark.Spark generates a model based on the previous ratings,comments,likes and dislikes spark will recommend which book is the best.If user gives input as “Recommend a Book”. Then Ro-De get connected to spark and sends command as “Recommend”, the spark then process and retrieves the Best book based on the model it has generated.

Interactive Communication:

Ro-De can perform interactive communications with the user. i.e based on the user choice it will selects the next command to speak. If user gives command as “can you sing”. Ro-De gives reply as “Yes i can sing, Do you want me to sing”. If user gives input as “Yes” then Ro-De sings a song. Else Ro-De asks user as “What can i do for you”.

Text-to-Speech :

For text-to-Speech we have used iOS inbuilt feature “AvSpeechSynthesizer”, which takes the String as the input and converts them as a speech.

Speech-to-text :

Google speech to text service is being used to parse the voice to textual data. using the socket client architecture, the Android device is going the send the data from the device to the Robot.Then the Ro-De will perform the action, based on the command sent by the device.

Accelerometer :

Accelerometer data gets the values of x,y,z coordinates based on the robo movement. Based on the accelerometer data, Ro-De can automatically detects whether it is in inclination or declination and increases/reduces its speed accordingly.

Dance:

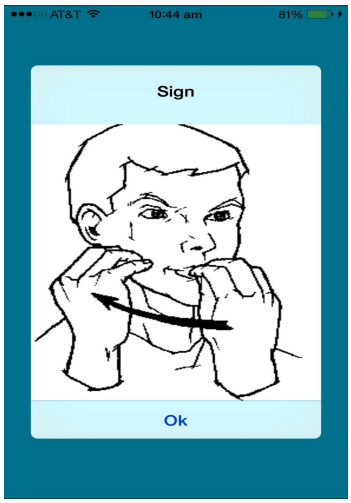
Ro-De tries to entertain the user and children's by singing and dancing. If user gives input to dance, then Ro-De Rolls.

```
[self.Romo3 turnByAngle:270 withRadius:0.1 completion:^(BOOL success, float heading) {
    if (success) {
        [self.Romo3 driveForwardWithSpeed:0.3];
        [self.Romo3 turnByAngle:-270 withRadius:0.1 completion:^(BOOL success, float heading) {
            if (success) {
                [self.Romo3 driveBackwardWithSpeed:0.3];
                [self.Romo3 turnByAngle:50 withRadius:0.05 completion:^(BOOL success, float heading) {
                    if (success) {
                        [self.Romo3 driveForwardWithSpeed:0.3];
                        [self.Romo3 turnByAngle:-40 withRadius:0.05 completion:^(BOOL success, float heading) {
                            if (success) {
                                [self.Romo3 turnByAngle:270 withRadius:0.1 completion:^(BOOL success, float heading) {
                                    if (success) {
                                        [self.Romo3 driveForwardWithSpeed:0.3];
                                        [self.Romo3 turnByAngle:-270 withRadius:0.1 completion:^(BOOL success, float heading) {
                                            if (success) {
                                                [self.Romo3 driveBackwardWithSpeed:0.3];
                                                [self.Romo3 turnByAngle:50 withRadius:0.05 completion:^(BOOL success, float headi
```

5.. RESULTS:

Sign Image:

If user gives input as “”Can you show me Sign Images” then Ro-De reply’s as “Yes, what can i show”, next if user gives command as “Show GoodBye” .It gets result from Mongo-bd and shows it to user.

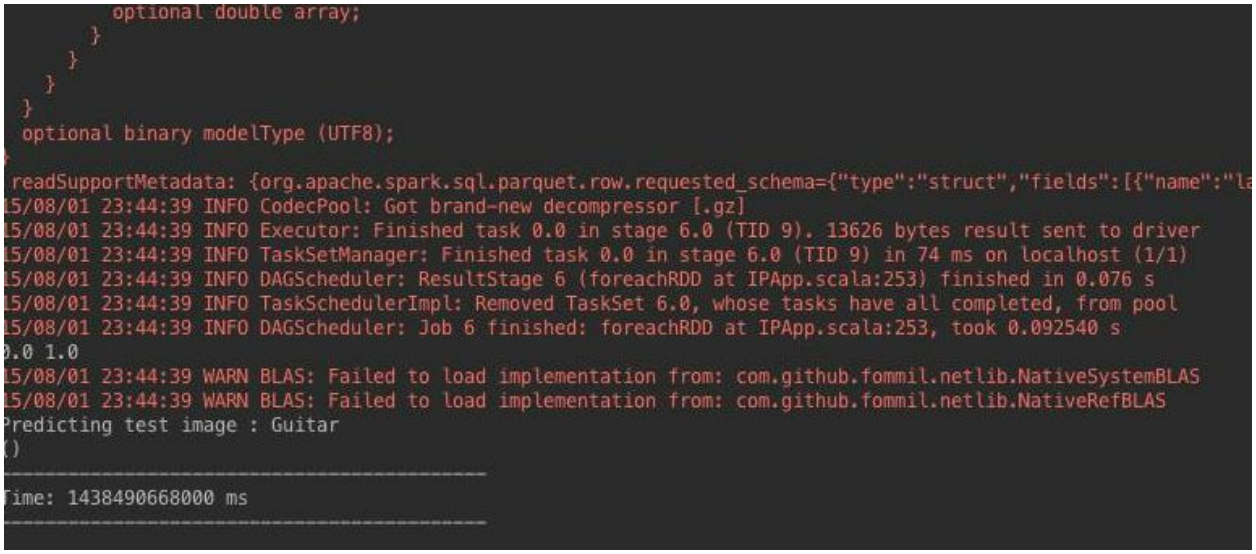


Recognize Image:

If user gives input as “Recognize image”, following is the image Ro-De is detecting to classify which group it belongs to.

Input Image:

Output in Spark:



Output from the spark, here it is “Guitar” is spoken by the Ro-De.

6. CONCLUSION:

Ro-De has provided the best performance in crash recovery and moving to specified distance , 90 % accurate result for recognizing image using spark image classifications based on the data set trained to get feature model. Ro-De provides accurate book recommendations in spark based on the data provided. Ro-De gets perfect clear input from client using google speech-to-text api. Our Ro-De can entertain children by singing songs and dance.

Machine learning provided optimized solutions for getting better result for image recognition and book recommendations in spark.

7. FUTURE WORK:

Our future work of Ro-De is to detect sounds around it, and to notify the user what that sound is about. Adding more datasets to the image classifications in spark.

8. REFERENCE:

[1] ROMO - http://www.romotive.com/

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

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

[4] Spark Recommendation -

http://spark.apache.org/docs/latest/mllib-collaborative-filtering.html

[5] TEST-TO- SPEACH - https://developer.apple.com/library/ios/documentation/AVFoundation/Reference/AVSpeechSynthesizer_Ref/index.html