# INTEGRATION OF REAL-TIME SPEECH RECOGNITION AND ACTION IN HUMANOID ROBOTS

By

Katherine Yun

Senior Thesis in Electrical Engineering

University of Illinois at Urbana-Champaign

Advisor: Stephen E. Levinson

May 2019

## Abstract

Human speech and visual data are two crucial sources of communication that aid people in interacting with their surrounding environment. Thus, both speech and visual inputs are essential and should contribute to the robot's action to promote the use of the robot as a cognitive tool. Speech recognition and face recognition are two demanding areas of research: they represent two means by which intelligence behaviors can be expressed. In this thesis, we are interested in investigating whether a robot is able to integrate visual and speech information to make decisions and perform actions accordingly. The iCub robot will listen to real-time human speech from the user and point its finger at a person's face in an image as dictated by the user. In the following sections, our methods, experimental results, and future work will be further discussed.

Subject Keywords: humanoid robot, speech recognition, iCub motor control, face recognition.

## Acknowledgments

I would like to thank my thesis advisor, Professor Stephen E. Levinson, for his continual support and constructive advice throughout my senior research. I greatly appreciate the guidance and the insights he has provided in the fields of robotics and artificial intelligence. I would also like to thank Yichen Zhou, who completed the face recognition module for this research project. Also, I would like to thank Peixin Chang, who taught me how to use Yet Another Robot Platform (YARP) and the Gazebo simulator and constructed the initial stimulation environment for the iCub robot.

# Contents

# 1. Introduction

Language and vision are two essential parts of human cognition. We often take these cognitive functions for granted as they are basic functionalities of human beings. Vision guides our movements and allows us to match objects to existing concepts, while language is clearly a prerequisite for communicating with the environment and understanding higher-level concepts. Tasks as simple as recognizing objects and distinguishing speech from noise are still challenging to even the most advanced robots nowadays. At Language Acquisition and Robotics Group, we aim to develop intelligence robots with the capacity to learn and react to natural language, which this research project also investigates. In the past, the lab members have successfully completed projects on robot imitation learning, object recognition and piano key recognition. Thus, I would like to initiate a project that combines two important types of cognition functions: vision and speech. The goal of this research project is to integrate these two aspects of human communication with the robot's fine motor control. The iCub humanoid robotic platform and Gazebo simulation platform were used to develop an application which enables the robot to react to the speaker's commands and recognize faces.

## 1.1 Motivation

The aforementioned tasks might seem to be simple to us. Humans perform these tasks with almost no effort in their daily lives, but they still pose challenges to the robot. The motivation behind this research project is that we would like to program a robot that not only receives and processes audio input but also combines visual data with the speech commands to perform motor movements. This is important because a cognitive robot should have cognitive functions that are interactive and holistically connected.

## 1.2 Goals

The goal of this project is to implement an application for the iCub to perform the following tasks and interact with the environment in a Gazebo simulation. The program needs to first extract useful speech data from the microphone input stream and convert the audio data into text (name of the target). In addition, a face recognition module is required to recognize the target's face among other faces in an image. Last but not least, the iCub's fine motor control will enable the robot to move its index finger to point towards the target's face. The iCub will remain stationary until the next command is detected.

## 1.3 Platform and Simulation

The iCub robot is an open source cognitive humanoid robotic platform: it was developed at Istituto Italiano di Tecnologia (IIT) as part of the European Comission (EU) project. The iCub's software is built on Yet Another Robot Platform (YARP), which provides features that allow developers to organize individual connections. Moreover, it provides a communication system with various ports, hardware drivers, and modules needed by the robot's general control. The robot is equipped with various parts of the body, and each motor part has a number of degrees of freedom. On the physical robot, there are two PointGrey Dragonfly cameras as well as a microphone that take in visual and audio inputs. More information on the iCub's fine motor control will be discussed in the motor control section of the thesis.

This research project only involves simulation of the robot on the Gazebo simulation platform. The environment contains basic elements that are needed in the experiment, and the functionalities of the robot in the simulation exactly match those of the physical robot. In the Gazebo environment, I have placed two models besides the iCub robot model: a table and a huge wall that contains the gradient of a testing image. Both models are placed in front of the robot as shown in Figure 1. Thus, two .sdf files containing the features of the models are needed to insert

the models into the environment. The gradients of both models will appear in the images captured by the iCub's camera, which will be further processed to prepare for facial recognition.

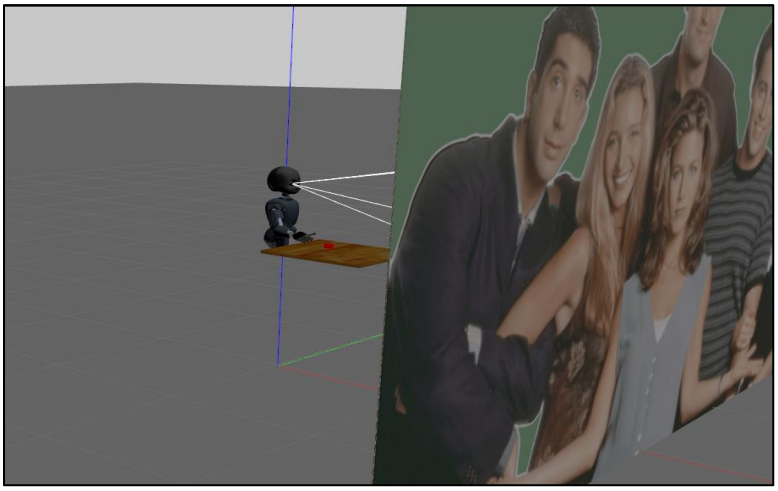

Figure 1: Project's Gazebo environment

## 2. Literature Review

The main challenge of this research project is that an algorithm is first needed for the real-time audio data capture while achieving a high accuracy in speech-to-text conversion. Iannizzotto et al. provides much insight on how techniques in computer vision, machine learning, and speech recognition can be combined for automation systems [1]. Besides the modules written for the iCub robot, an efficient human-robot interaction module is also required for this project: the program communicates with its users through terminal outputs and reminds the user when speech input is currently needed by the program.

In addition, a high-accuracy speech recognition model is needed to convert the input speech data into text. A classic model of speech recognition with recurrent neural networks (RNNs) is introduced in [2]. RNNs are inherently useful for classifying speech data, since their hidden states are a function of all previous hidden states. Convolutional neural networks (CNNs), on the other hand, are not suitable for sequential data, but a CNN-based system can be used to estimate phoneme class conditional probabilities with a standard Hidden Markov Model (HMM) decoder as discussed in [3].

More importantly, the program is expected to run in real-time, which requires pre-processing of input speech. [4] provides various language models and search algorithms that handle real-time speech input. “Continuous speech recognition” is given a clear mathematical formulation, since the search for target speech data often contains a unique acoustic pattern. Also, it’s important to further process the input data to remove noise from the background.

In addition to reviewing publications, I also utilized resources from several product websites and GitHub repositories including [5], [6], and [7].

Lastly, in order to fully understand the YARP plugins for the Gazebo simulator, I referred to [8] and its related documentation about running the iCub in the Gazebo simulation.

## 3. Methods

This project contains different interactive modules as shown in Figure 2. All sections of the program are necessary for the robot to complete its tasks. The modules will be discussed in details in this section of the thesis.

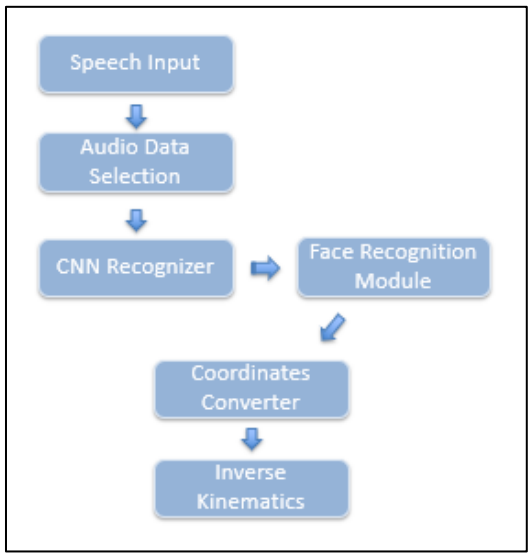


Figure 2: Function modules of the project

### 3.1 Speech Data Handling & Processing

The input speech will be acquired through the microphone of a laptop or an electronic device. To obtain the raw audio data, PyAudio, a cross-platform audio I/O library in Python, is used to simply record audio stream on Linux. The microphone instance will continuously store raw audio data in the buffer from the input stream until speech data is detected. The recognizer instance will detect speech data from the raw audio input by setting an energy threshold. Since background noise can greatly affect the robustness of this method, a function that dynamically adjusts the energy threshold based on audio input is included in this instance. This will calibrate the energy threshold with the ambient energy level to more effectively extract speech data from the microphone.

Snowboy, an embedded and real-time customizable hotword detection engine, is used to detect a single word from the speech input. A hotword is a key word or phrase that the program listens for to trigger the speech recognition module. Snowboy engine is now frequently used in real-time speech recognition engines such as Alexa and OK Google. The engine reads audio input for phrases until there is a phrase that is long enough and dynamically adjusts the energy threshold using asymmetric weighted average. After the speaking has stopped, it checks weather the unit energy (total energy/normalization factor) of the audio signal within the buffer is above the energy threshold. If so, this segment of audio data is passed on to the next module.

### 3.2 Speech Recognition Module

After the speech data of a single phrase is extracted, it is transcribed using the speech recognition module through trained neural networks. There exist many network models for speech to text transcription. Until the 2010's, the state-of-the-art for speech recognition models were phonetic-based approaches including separate components for pronunciation, acoustic, and language models [9]. There now exist a variety of recognition engines that enable automated speech recognition (ASR) online. In comparison, this project does not require a complete recognition engine that can transcribe almost every common word in the dictionary. Only the name of a person needs to be recognized, and it is assumed that the speaker will say one word at a time. Thus, the data set used for training the speech recognition module only needs to include the names of the six people who will appear in the test images. However, a higher-accuracy speech recognition network that recognizes names of people in the image is needed for this task [10].

The dataset includes a total of 240 recorded command utterances in which 32 utterances are used for training each name and 8 utterances are used for testing. Human-centric transformation for speech data is to compute Mel-frequency cepstral coefficients (MFCC). Either 13 or 26 different

cepstral features, as input for the model. After this transformation the data is store as a matrix of frequency coefficients (rows) over time (columns). The speech commands are preprocessed to have the same length and are stored as 98 by 13 matrices, where 98 represents the number of audio frames and 13 represents the number of frequency coefficients per frame.

Convolutional neural networks (CNNs) are used to train the data set. The speech recognition model used in this module consists of three 2D convolutional layers. Each CNN layer is followed by a max-pooling layer with rectified linear operator (ReLU) activation functions and local response normalization (Norm). At the end, there are two densely connected layers again with ReLU activation functions. The outputs are classified using SoftMax, which calculates the probabilities for each of the six labels. The loss function used in this model is categorical cross entropy and the optimizer used is Adadelta, which is an extension of Adagrad that seeks to reduce its aggressive, monotonically decreasing learning rate. Finally, the model will output an array with scores for six different classes (names for six targets in the test image). We will use the coordinates of the center of the target's face, with the highest output score, for the iCub's inverse kinematics. The network structure is illustrated in Figure 3.

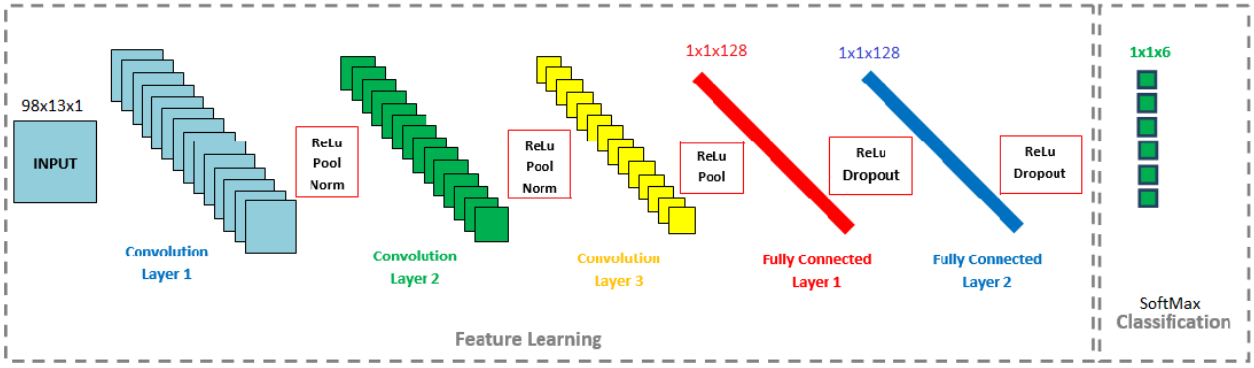


Figure 3: CNN architecture of the speech recognition module

One important thing to note is that if the user chooses to speak a word that is not included in the training set or the model fails to recognize the word spoken by the user, it will be automatically labelled as “unknown” and the face recognition module will the output an empty matrix, which will lead the program to terminate. If the energy is not great enough for the engine to recognize it as a speech input, the user will be prompted to repeat the word. After three unsuccessful trails, the program will terminate as no clearly spoken word is detected.

The automatic speech recognition for people’s names is often more difficult because names can have different spellings or even different pronunciations. To minimize errors caused by this, I made sure that the names involved in the training set only have one common spelling as well as pronunciation. As the first letter of the name is often capitalized, all labels given to the speech training data are in lowercase and only the outputs printed on the terminal are capitalized.

## 3.3 Face Recognition Module

The name of the target is determined by the speech recognition module and serve as an input to the face recognition module whose basic structure is shown in Figure 4. The module first recognizes faces in an image and outputs a label (the person’s name) to match each detected face.

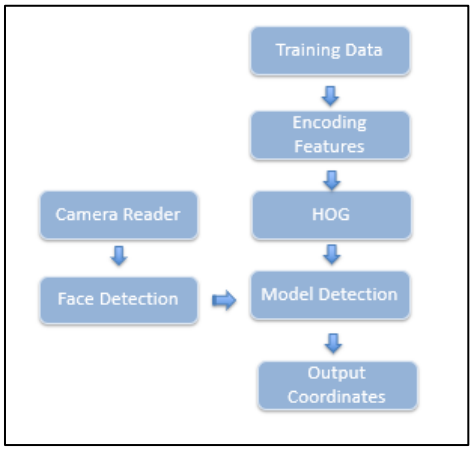

Figure 4: Flow chart of the face recognition module

Histogram of oriented gradients (HOG) technique is used as the foundation for this face detection task: occurrences of gradient orientation are found in localized portions of an image [11]. Thus, a dense grid of uniformly spaced cells is computed, and the accuracy is further improved by using overlapping local contrast normalization. An example of HOG features is shown in Figure 5.

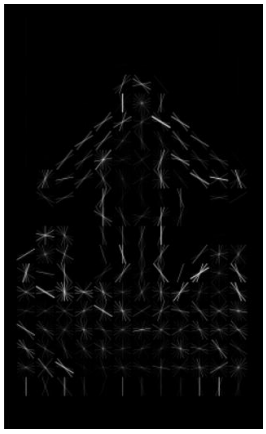

Figure 5: Example of HOG features

The model that is used for face classification is Very Deep Convolutional Networks for Large-Scale Image Recognition (VGG 16) as shown in Figure 6. VGG 16 consists of sixteen network layers: thirteen convolutions layers with 3*3 sized kernels and three max pooling layers [12]. The layers are fully connected at the end.

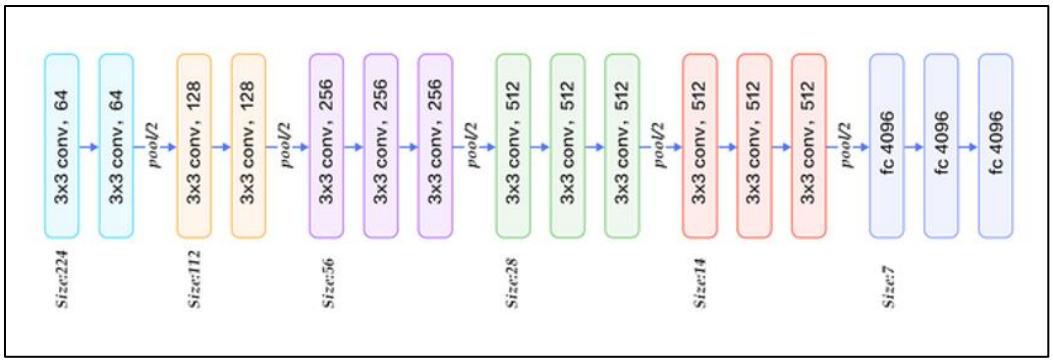

Figure 6: The network structure of VGG 16

The network takes features collected by HOG and outputs the coordinates of the target's face. OpenCV is used to draw a rectangle around the target's face and label the face with its corresponding name label in Figure 7.


Figure 7: Example output of face recognition module

The data set that is used for training and testing the model is collected from Bing Image Search API, which is an application that helps users scour the web for images. This data set only includes faces of the six actors and actresses who appeared on the television sitcom Friends. There are a total of 2400 images collected in this data set: 400 images are included for training each person's face, in which 320 images are used for training and validation and 80 images are used for testing. Only faces of the six actors and actresses will appear in the simulation. In other words, each person's face is only tested against the other five people, so a large data set is not required here.

## 3.4 iCub Motor Control

Motor control of the iCub robot in YARP is done through a device driver. As shown in Figure 8, the physical dimensions of the iCub are similar to that of a 3.5-year-old child. The robot has 53

actuated degrees of freedom organized as follows: 7 in each arm, 9 in each hand, 6 in the head, 3 in the torso/waist, and 6 in each leg. The head also has stereo cameras where eyes would be located on a human and microphones are on the side [13]. In this project, we only used one camera of the robot since depths of the objects in the environment are previously known. In addition to the camera, the right hand of the robot will be program to perform movements such as lifting and pointing at a specific direction.

The YARP Cartesian Interface enables the fine motor control of the arms as well as the legs of the robot directly in the operational space [14]. The user can program the robot to reach a specific configuration using inverse kinematics, expressed as a combination of a 3D point to be attained by the end-effector. One thing to note is that the default end-effector of the robot is the center of the palm and the desired orientation is given in axis-angle representation as shown in Figure 9. For this project, a different effector is defined to be located at the robot's index finger so that the direction at which the robot's finger points at can be controlled.

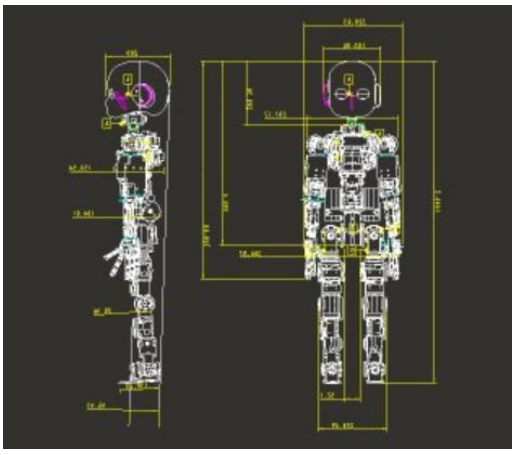

Figure 8: iCub motor schematics

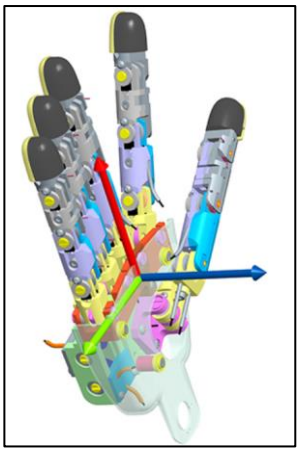

Figure 9: iCub's default end-effector orientation

Before performing inverse kinematics, the coordinates given by the face recognition module should be converted to the robot frame. First of all, the coordinates are converted to the picture

frame (the portion of the image that only contains the picture model), which is labeled in red in Figure 7, and then converted from the world frame to the robot frame.

The camera will capture an image with 480x640 pixels. We know the 2D coordinates $[X_i, Y_i]$ outputted by the face recognition module and the coordinates in the picture frame $[X_p, Y_p]$ are derived using the formula below:

$$X_p = \frac{Xi * image\_height}{480}$$

$$Y_p = \frac{Yi * image\_width}{640},$$

where image_height and image_width represent the size of the picture shown in the image captured by the camera and the 'p' subscripts represent the picture frame.

Next, the coordinates are converted to the 3D world frame according to the following equations:

$$X_w = model_w[x]$$

$$Y_w = \frac{model\_width}{2} - model\_width * \frac{Yp}{image\_width}$$

$$Z_w = model_w[z] + \frac{model\_width}{2} - \frac{Xp}{image\_height},$$

where the distance from the robot $X_w$ to the image remains constant and the 'w' subscripts represent the world frame. The size of the picture model is 4 x 5, and it's located at $model_w$ = [3, 0, 3], which is in terms of the world frame.

Finally, the coordinates will be converted from the 3D world frame to the 3D robot frame. The robot's position frame in respect to the world frame is shown in Figure 10. Thus,

$$X_r = - X_w$$

$$Y_r = - Y_w$$

$$Z_r = Z_w - 1,$$

where the ‘r’ subscripts represent the world frame.

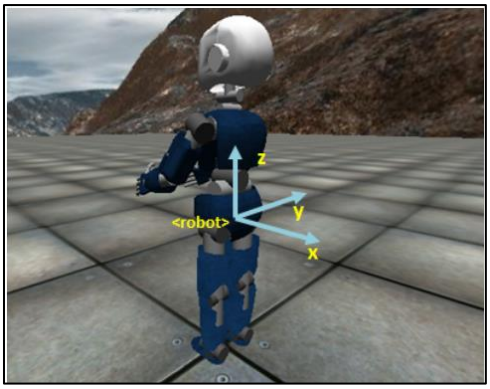


Figure 10: Robot’s position in respect to the world frame

These coordinates will then be used as the ending position for the iCub’s inverse kinematics. There is a function to call inverse kinematics in the Cartesian Interface [15]. The orientation of the end-effector (index finger) will always be in the positive $X_r$ direction so that only ending positions are changed when the robot’s arm performs its movement to avoid flipping the right hand.

## 4. Experimental Results

In this section of the thesis, the experimental results of the program implementation will be discussed. The experiments were run on the Gazebo simulator. The results were first recorded separately for each module (speech recognition and face recognition) in terms of recognition accuracy, and then the overall result is recorded every time the robot has completed its movement.

The final result of the program can be both inspected by printing the ending pose of the end-effector or visually compare the distance between the right hand of the iCub robot to the target's face in the image. An example of the final output with the speaker saying the name "Joey" to the microphone is shown in Figure 11. The program first needs to correctly identifies the speech data inputted by the user and transcribes the speech data into lowercase text (e.g. joey). Next, the label outputted from the speech recognition module will be forwarded to the face recognition module, which will recognize the face of the target based on this label. High accuracies in both processes are both crucial to the final result.

After coordinates outputted from the face recognition module are converted to the robot frame, the robot can then perform its movement through inverse kinematics and point its index finger towards the direction of the target's face as shown in the figure below. The index finger position appeared in the camera is boxed in red which should correspond with the coordinates in the picture frame.

Figure 11: An example of the program's output

The application can be unstable due to the following constraints. First of all, real-time speech recognition is largely affected by the current background in which the speaker is in as well as individual differences in pronunciation and speed. Thus, the training data set should be as large as possible to minimize the effect of individual and background. Secondly, the accuracy of face recognition is determined by the illuminance and image quality since the simulation is done in a Gazebo environment. As we have tested, the lighting condition of the environment and the image quality will substantially affect the accuracy of face recognition module.

Finally, all models including the iCub robot need to be inserted in the simulation environment before the first module, speech data processor. However, due to runtime constraints, models sometimes fail to be inserted to the environment which leads to an empty image outputted by the image grader, hence the face recognition module will not have a valid image to recognize in time. A solution to this problem is to set a wait time after each model is inserted but this will lead to an increase in overall runtime.

## 5. Conclusion and Future Work

This thesis focuses on the implementations of speech recognition, face recognition, and motor control in a humanoid robot. The iCub robot is able to listen to the speaker's input in real-time and convert raw audio data to a person's name in text format. Then, face recognition will be performed on a given image that includes that person's face, and the robot is able to point in the directions of the person's face in the image. The most challenging portion of this project is to integrate all required modules while having high accuracies in both speech recognition and face recognition. A failure in either of the processes will greatly harm the final result. Thus, this research project can be further improved to increase the accuracy in both modules to yield a more stable application.

In terms of future work, there are many aspects of this project that can be improved or modified. The implemented application can also be used as the foundation for a variety of new projects. As humans, we are able to recognize more complex sentence structures and perform a set of movements. In this specific project, only one hot word is detected before the program stops listening to the input stream, as only the name of one person is needed for face recognition module. Also, the robot only performs one movement: moving its index finger and arm from the neutral position to a position that points towards the target's face using inverse kinematics.

In future projects, the speaker can instead say a sentence that contains not only the name of a person but also a movement that corresponds with the name such as pointing using its right arm or nodding its head. This will of course increase the difficulty of the speech recognition module, but it is an example of a more complex cognition function that can be implemented on the robot.

## References


[1] G. Iannizzotto, L. L. Bello, A. Nucita, and G. M. Grasso, “A vision and speech enabled, customizable, virtual assistant for smart environments,” 2018 11th International Conference on Human System Interaction (HSI), 2018.

[2] A. Graves, A.-R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013.

[3] D. Palaz, M. Magimai.-Doss, and R. Collobert, “Convolutional Neural Networks-based continuous speech recognition using raw speech signal,” 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015.

[4] C. Scagliola, “Language models and search algorithms for real-time speech recognition,” International Journal of Man-Machine Studies, vol. 22, no. 5, pp. 523–547, 1985.

[5] KITT.AI, “Snowboy hotword detection.” https://github.com/kitt-ai/snowboy, 2018

[6] Robotology, “iCub.” https://github.com/robotology/icub-main, 2019

[7] Robotology, “YARP.” https://github.com/robotology/yarp, 2019

[8] "ICubForwardKinematics," 27 February 2014. [Online]. Available at: http://wiki.icub.org/wiki/ICubForwardKinematics. [Accessed 21 April 2019].

[9] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, “Applying Convolutional Neural Networks concepts to hybrid NN-HMM model for speech recognition,” 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012.

[10] V. Tikhanoff, A. Cangelosi, and G. Metta, “Integration of speech and action in humanoid robots: iCub simulation experiments,” IEEE Transactions on Autonomous Mental Development, vol. 3, no. 1, pp. 17–29, 2011.

[11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05).

[12] K. Simonyan and A. Zisserman, “VGG16 – Convolutional Network for Classification and Detection”, Neurohive, 2018. Available at: https://neurohive.io/en/popular-networks/vgg16. [Accessed 21 April 2019].

[13] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale and F. Nori, "An open-source simulator for cognitive robotics research: The prototype of the iCub humanoid robot simulator," in Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems, 2008.

[14] E. M. Hoffman, S. Traversaro, A. Rocchi, M. Ferrati, A. Settimi, F. Romano, L. Natale, A. Bicchi, F. Nori, and N. G. Tsagarakis, "YARP based plugins for Gazebo simulator," Modelling and Simulation for Autonomous Systems lecture notes in computer science, pp. 333–346, 2014.

[15] G. Metta, P. Fitzpatrick, and L. Natale, "YARP: Yet Another Robot Platform," International Journal of Advanced Robotic Systems, vol. 3, no. 1, p. 8, 2006.