UNIVERSITY *of* MARYLAND
EASTERN SHORE

SCHOOL *of* BUSINESS AND TECHNOLOGY
Department of Engineering and Aviation Sciences

Design of Sign AI:
An AI-Based ASL Translation Tool

Josheb Dayrit

Advisor: Dr. Zhang

Spring 2020

List of Contents

List of Tables

List of Tables

Abstract

By the end of the project, summarize the project into short text and put here.

1. Introduction

The following project seeks to develop an ASL (American Sign Language) translation tool for deaf persons. In addition, it will be hosted on a microcomputer, like the 32-bit Raspberry Pi. Finally, a deep learning model will be trained to recognize ASL through pre-processed sEMG (surface electromyography) data collected by sEMG electrodes. Needless to say, each part in the development process is crucial – doubly so for the deep learning model, which is in charge of gesture classification and translation.

As far as additional hardware is concerned, there will be an LCD providing the user with an interface on which spoken words that are converted to text are displayed. Communication is a two-way street, after all; the LCD enables the user to not only express, but to also understand. The end product is expected to offer its deaf users a “give” as well as a “receive” functionality. That is, it facilitates two-sided conversation, where the user (who is assumed to be deaf) is able to speak as well as be spoken to. To increase the convenience and portability of the end product, it will also be mounted on a pre-made, adjustable armband. This mounting should be possible since the microcomputer where the deep model is hosted is light enough. In general, microcomputers such as the Raspberry Pi are not heavy, so weight is a non-issue.

For Sign AI, ease of use as well as translation accuracy are of utmost importance. Ease of use is incorporated in the design process through accessories which increase portability and convenience. Meanwhile, translation accuracy hinges on the deep learning model and its ability to differentiate between different types of EMG data. In the end, what Sign AI aims to offer is a portable and reliable ASL translator that creates a user-friendly experience by automating the translation process. Everything is done in the backend and little input is required from the user.

1.1 Backgound/Motivation

Different means of communication have existed since time immemorial. While speech stands as the most common mode of communication, people are also able to communicate through writing. It is no surprise that unique modes of communication are as plenty as they are ubiquitous. In technology, computers use a scheme of 0s and 1s to perform complicated tasks and display the outputs; in nature, all scores of animals are able to encode meaning in the noises they produce. For modern humans, speech is intrinsic to communication. However, the two are not at all exclusive to each other. In a paper titled "The Origin of Human Multi-Model Communication," two researchers (Stephen Levinson and Judith Holler) from the Max Planck Institute for Psycholinguistics and Donders Centre for Brain, Cognition and Behavior stated that "the modern human communication system is, on a biological time scale, a recent innovation" [1]. Citing the Gestural Theory of Language Origin, Levinson and Holler believed that the initial humans (the "cavemen," or Homo troglodytes) must have developed and employed a gesture-based language. They also believed that changes in anatomy triggered a gradual but inevitable shift from gesture-based communication to vocal communication in humans.

In a society that now communicates (primarily) through speech, it will be difficult for deaf persons to assimilate – and, more importantly, interact – with their non-deaf peers. According to the Gallaudet Research Institute, in year 2006, "nearly 10,000,000 persons (Americans) are hard of hearing and close to 1,000,000 are functionally deaf" [2]. This is also corroborated by more recent (2012) statistics from the Center of Disease Control and Prevention (CDC), which state that "approximately 15% of American adults (37.5 million) aged 18 and over report some trouble hearing" [3]. While the CDC clarifies that "the overall annual prevalence of hearing loss (has) dropped slightly" [3] with the passing of time (from 16% in 1999 to 14% in 2012), additional findings from the National Deaf Center on Post-Secondary Outcomes reveal an alarming truth about deafness: "In 2014, only 48% of deaf people were employed, compared to 72% of hearing people" [4].

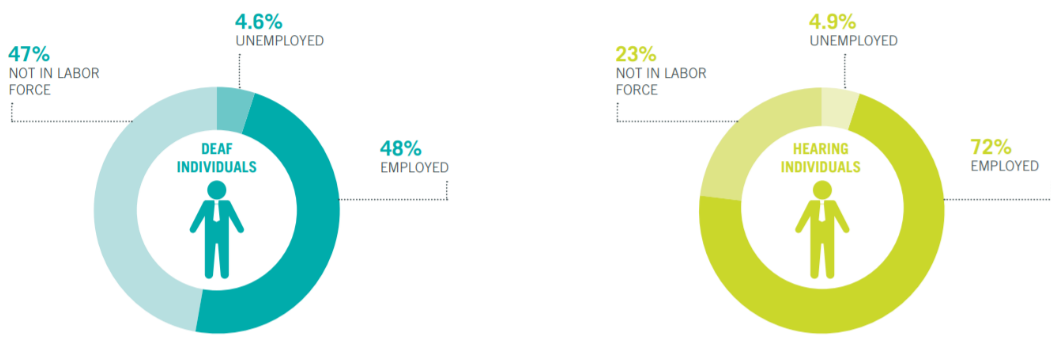

Figure 1: Rates of employment in 2014, according to the National Deaf Center on Post-Secondary Outcomes

Unfortunately, disparities in opportunities and (to some extent) outcomes are due in large part to physical disabilities, as shown by the countless statistics on deaf people. Since deaf people are not capable of speech, they choose to communicate through gesture-based languages such as ASL (American Sign Language). Because of this, however, prospects for work are limited for deaf people. The harsh reality is that many employers are unwilling to hire deaf employees. While the ADA (American Discrimination Act) outlaws discriminatory practices based on race, gender, or disability, it does not prevent them entirely. In fact, in an American study which closely scrutinized successful discrimination lawsuits made against companies, the EEOC (Equal Employment Opportunity Commission) concluded that "young people who are deaf or hard of hearing may be advised to anticipate some resistance from employers" [5].

As outrageous as it sounds, even if a deaf candidate applies for a position while possessing qualifications that meet or exceed a company's standards, the factor that determines their employment is whether or not they require excessive accommodation. From a corporate perspective, hiring employees who will only place undue burden on a company and its resources is generally not worth the cost. This idea also aligns with the EEOC's secondary conclusion, which states that disabled applicants who "articulate well their qualifications" *and* provide long-term work-arounds or solutions to their disabilities are "more likely than others to find success in the world of work."

One solution that Sign AI brings to the table is the function of a basic, real-time, AI-based ASL translation tool. Deaf people face many barriers of entry regarding employment, but communication is, by far, the most pressing issue. In the workplace, collaboration among employees is important. Typically, company projects are completed not by a single person, but by teams of people making collective contributions. A pre-requisite to effective collaboration is effective communication, especially in the nascent stages of a project, during which first impressions are gathered, and ideas are put forth for evaluation by others. For deaf persons, communication comes in the form of sign language; however, for non-deaf persons, this is not the case. Deaf persons have a hard time expressing their ideas to non-deaf persons, because sign language is fundamentally different to spoken language. To bridge the gap between sign language and spoken language, a linguistic interface is needed between the two languages to translate one to the other. Sign AI strives to fulfill this role through a trained AI, a TTS (text-to-speech) feature that will speak out the meaning of various ASL gestures, and an STT (speech-to-text) feature that will translate from spoken word back to text.

1.2 Objective

The objective of Sign AI is to, quite literally, give a voice to deaf persons. The inability to speak properly is one disadvantage of deaf persons that either prevents them from fully pursuing specific career paths or leads them to be unjustifiably discriminated against in their search for employment. This inability stems from the fact that speech is a product of both the ears and the mouth; the mouth produces the sounds, while the ears regulate them. Without coordination from both the mouth and ears, it becomes increasingly difficult to produce intelligible speech. Many non-deaf employers simply do not have the time to learn sign language for their deaf applicants. In the hiring process, most are likely to turn their attention to the non-deaf candidates, even if the deaf candidates have the necessary qualifications. Such discrimination is, by all accounts, unfair. However, there is some wisdom to be found in it, even if morally questionable. All businesses are utilitarian in that their goal is to maximize productivity while minimizing expenditure of any sort (human, financial, etc.). If a prospective employee demands more from a company than their fellow competitors, that company will likely settle for those competitors in hopes of making things easier on themselves. While Sign AI cannot shift this mindset, it can remedy, to an extent, the one disadvantage encumbering deaf persons when competing with their non-deaf peers.

1.3 Design Requirements

The design requirements for Sign AI fall into 2 distinct categories: physical and non-physical. The non-physical part of Sign AI deals with requirements for AI itself (which is responsible for the translation), while its physical part deals with requirements for hardware (the “package” on which the AI is hosted) as well as any necessary software.

Non-Physical Design Requirements:

1. The final product must be able to differentiate between gestures with an accuracy of 70-90 percent. The reason behind this accuracy range lies in the previous successes of similar ASL projects as well as their scope. Past ASL projects have reached accuracy values close to and even beyond the specified accuracy threshold. However, it should be noted that the scope of these projects were more limited than Sign AI in terms of the number of gestures classified.

   One example is a project done by professors from a Pakistani institution. In a paper titled "American Sign Language Translation through Sensory Glove," they attempted to use flex and contact sensors alongside an accelerometer to recognize differences in the activation of muscles when performing ASL gestures. "The glove was found to have an accuracy of 92 percent," the paper's abstract read.

   Like other projects, however, the sensory glove had glaring limitations, particularly in the number of gestures that it can translate. While the glove is capable of translating the alphabet with 92 percent accuracy, that is the extent of its translation. The limited scope of the project might have played a part in achieving its high accuracy, as having to do less translation work would mean there are less sources of error. If the glove were to process 100 gestures as opposed to 26 gestures, realistically-speaking, it would have a harder time differentiating between them, especially if some gestures do not produce distinct measurements. The takeaway is that, the more gestures there are to classify, the greater the risk for misclassification becomes. This is why the aim of Sign AI is not 90 percent accuracy, but, rather, 70-90 percent accuracy. The tolerance is necessary to account for the to-be-determined increase in misclassification due to the wider scope of the project.

2. Sign AI is a word-based translator, giving translations word-by-word. Its ability to translate from ASL to English will span 100 words. This design requirement was inspired from Dr. Bill Vicars, who is both an ASL professor, as well as a deaf person. On his website called "HandSpeak," he compiled a list of the 100 gestures in the ASL language that are used with the most frequency. A deep learning model will be trained to recognize and classify these gestures. Then, a script will store the classification and read it out through a voice synthesizer from a TTS (text-to-speech) software.

Physical Design Requirements:

1. TTS software, alongside a speaker for sound amplification, will speak out translated gestures.
2. The translation tool should enable deaf users to have basic conversations with their non-deaf peers. This means that, in addition to being able to translate ASL to English, Sign AI must also be able to convert speech into text that can be perceived by the deaf user. What is required for this is STT (speech-to-text) software, the reverse of TTS (text-to-speech) software. STT software, first requiring a microphone to pick up audio, will then encode said audio to words.
3. An LCD display will also be required to display the words on a screen, where they can be read by the deaf user.
4. Finally, the translation tool (the hardware) will be mounted on a pre-made armband for portability purposes.

1.4 Design Constraints

Although design constraints are few in number due to the "hands-off" nature of an AI project, undoubtedly, they do exist. One constraint lies in the ASL language itself, while others in hardware and cost.

1. Some gestures are not arm-based or hand-based, which means that they cannot be measured by the MyoWare muscle sensors. In ASL, pronouns such as "he," "she," or "they" are expressed through a singular gesture: pointing a finger. With the current setup of sensors, there is no way to determine if a gesture meant for a pronoun is referring to a "he," "she," or "they."
2. Connection to the Internet is required for the deep learning model. The alternative would be to install TensorFlow and other software locally, which is ill-advised since storage space is limited in microcomputers.
3. Regarding expenses, the project will cost $300 in total. The cost of each MyoWare sensor is $37.99, while the cost for other hardware like the LCDs, speakers, microphones, or A/D (analog-to-digital) converters varies depending on the vendor. Each MyoWare will require a A/D converter, as the Raspberry Pi does not accept analog inputs. In addition, a speaker and microphone will be needed for performing (respectively) text-to-speech and speech-to-text. Lastly, an LCD will be necessary for displaying speech-to-text outputs.

1.5 Design Method (Approach)

It should be noted that the classification process hinges upon not only the accuracy of the neural network (AI), but also the reliability of the data that is being used to train it. This data originates from the sensors, and the particular brand of sensors that will be implemented is the MyoWare Muscle Sensor developed by Advancer Technologies. The definition of “reliable data” changes from project to project. In the case of Sign AI, however, it is data that can capture (via EMG measurements) the disparate characteristics of each gesture.

Naturally, in a project that is wholly dependent on data, the first step is to collect it. Then comes the pre-processing stage, where EMG data is formatted in a way that can be passed on to a neural network for training. Data collection is expected to be a major bottleneck in the development of Sign AI, as it is a time-consuming endeavor. Not to mention, the tasks of collecting data and training the neural network cannot be pipelined because the neural network takes in the collected data as an input. As a solution, pseudo-data will be randomly generated in place of real sensor data.

Before the pseudo-data is generated, however, a plan has to be drafted relating to how it will be implemented into the project. To recap, the end goal is to develop a neural network that can differentiate between various types of data. Considering this, the pseudo-data must represent a wide range of data. One idea was to plot a sample of 10 points on an x-y graph, with each bounded by a pre-determined y-limit. First, the graphs will be generated by a script; then, they will be formatted accordingly for use of the neural network. These procedures encompass the second step.

The third step would be the design of the neural network itself. Architectures for neural networks are plenty, and some even specialize in solving niche problems. For example, there exists the recurrent neural network (or RNN, for short) which is used for pattern recognition is text and speech. When it comes to image processing, the ConvNet (or convolutional neural network) was developed to fulfill the need. Due to its universal application in the realm of image processing, the standard ConvNet design (found on the TensorFlow website) will be used as a base by the Sign AI. Following the third step is the fourth, which is exporting the finalized neural network on a Raspberry Pi and assembling the necessary hardware for the text-to-speech and speech-to-text functionalities.

2. Project Description

2.1 System Description

In progress.

2.2 System Diagram

The system diagram is displayed in Figure 2. To begin, the elements underneath the Raspberry Pi are the software programs installed on it. The hardware elements on the left (i.e. the microphone and the sensors) provide the Raspberry Pi with inputs, while the hardware elements on the right (i.e. the speaker and LCD display) actualize the outputs.

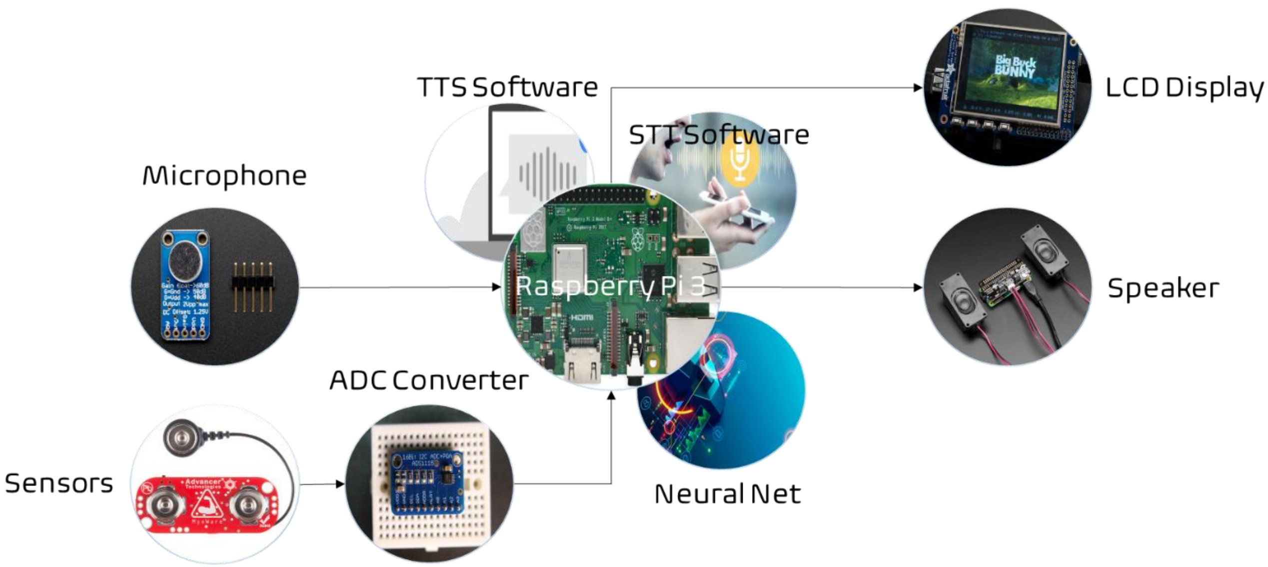

Fig. 2. The system diagram

2.3 System Functions

The system diagram can be decomposed into 3 separate sub-systems: the sensor-to-Pi subsystem, mic-to-display subsystem, and AI-to-speaker subsystem. Each subsystem, when put together, is responsible for facilitating the flow of data throughout the overall system.

Unlike the Arduino, the Raspberry Pi 3 does not have an innate method of processing analog signals. In fact, it does not even have analog pins; every pin on a Raspberry Pi is for digital information. Therefore, an ADC converter is required to connect data coming from the MyoWare muscle sensors to the Raspberry Pi. Obviously, some data will be lost when the analog-to-digital conversion is performed. However, this is a necessary evil due to the limitations of the Raspberry Pi.

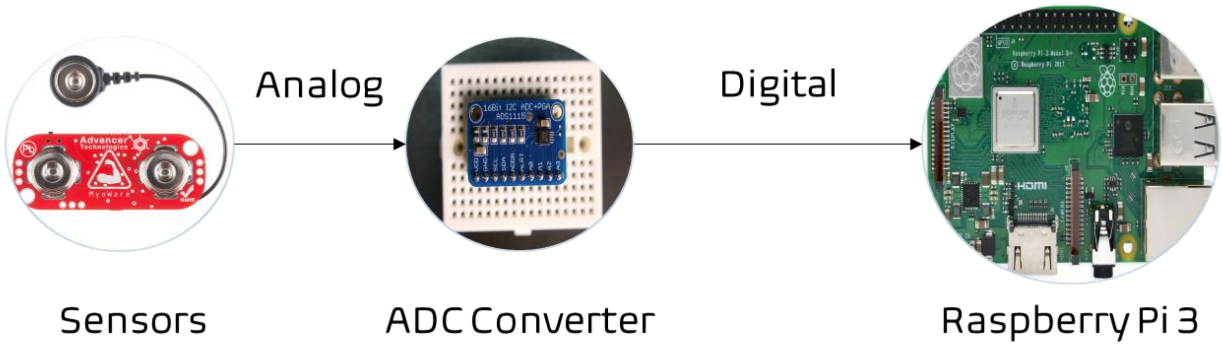

Fig. 3. The sensor-to-Pi subsystem

One of the primary functions of the Sign AI is to allow deaf users to be able to understand their non-deaf peers through a speech-to-text function. This speech-to-text function is achieved through the use of a microphone (which encodes audio input into a data file) and a speech-to-text software (which processes the data file in question and returns a text file). First, an audio file will flow from the microphone to the Raspberry Pi. Then, audio-to-text conversion happens once the speech-to-text software is allowed run by the Raspberry Pi; the output is a text file. In the final step, this text file will appear in the LCD display for the deaf user to read.

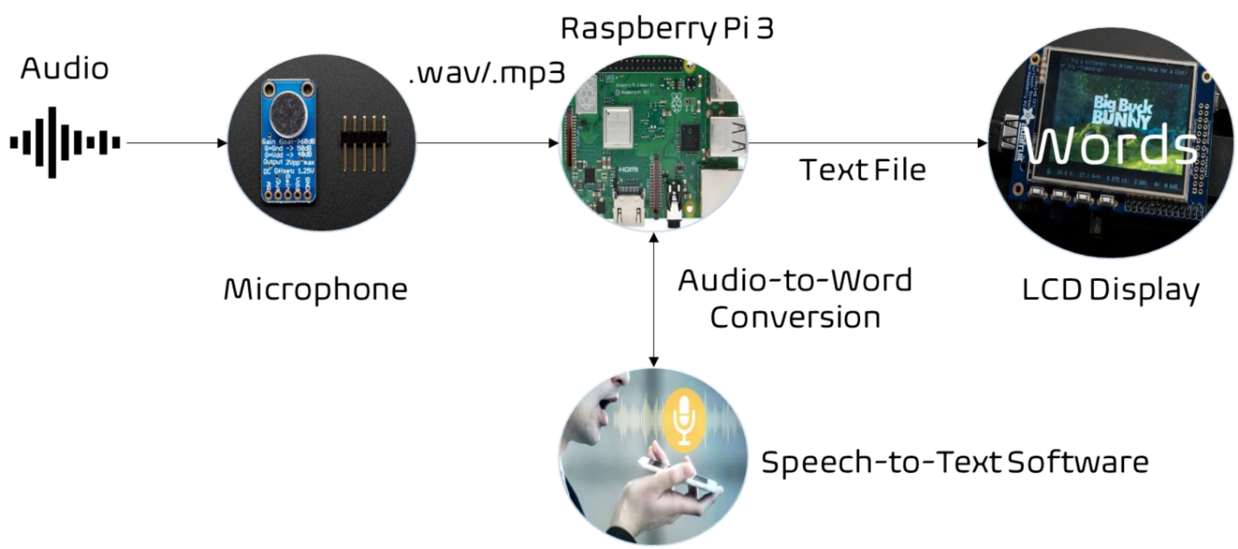

Fig. 4. The mic-to-display subsystem

Last but not least is the subsystem involving the neural network and speaker; both components are required to perform text-to-speech conversion. Once it receives input from the muscle sensors, the neural network then classifies the data that is fed into it as one of the gestures in its database. After this process, the neural network packages its classification into a text file that is subsequently read by the Raspberry Pi. Flaring to life afterwards is the text-to-speech software, which processes the most recent file and returns an audio file that is broadcast and amplified by the speaker.

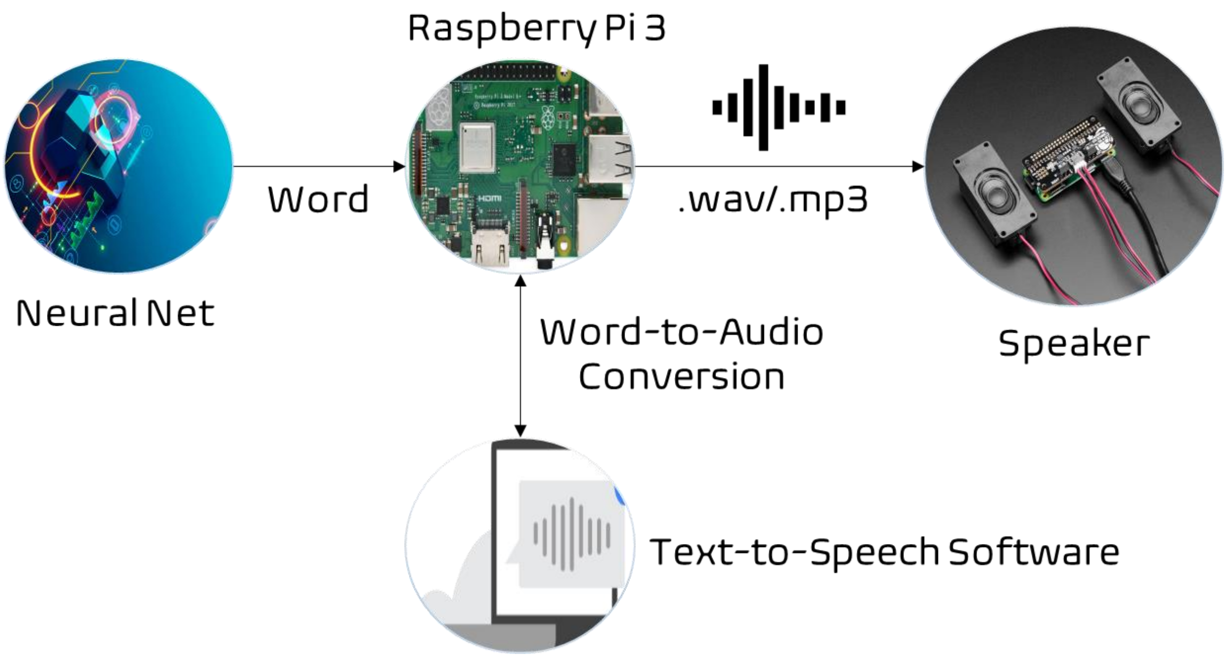

Fig. 5. The AI-to-speaker subsystem

3. Implementation Plan

3.1 Tasks

- Task 1: Design and execute a *general* plan for the classification of graphical pseudo-data through a TensorFlow model.
    - Subtask 1.1: Seek out a resource that allows the storage and access of files on the cloud. Because micro-computers like the Raspberry Pi 3 have limited disk space, it would be ill-advised to store files on them. Instead, project-related files like scripts, software, or program libraries will be stored (and, later, accessed) on the cloud.
    - Subtask 1.2: Seek out an IDE (integrated development environment) that grants access to a cloud GPU suitable for deep learning projects. This will speed up the training process immensely.
    - Subtask 1.3: Write Python script to generate, plot, and save pseudo-data as an image file.
    - Subtask 1.4: Write a script to cycle through the image files and convert them to normalized arrays of RGB or grayscale values.
    - Subtask 1.5: Write a script to initialize, design, train, and evaluate a TensorFlow model.

- Task 2: Develop a TensorFlow model for the classification of EMG data based on the model in Task 1.
    - Subtask 2.1: Recruit test subjects for data collection.
    - Subtask 2.2: Devise an efficient and accurate method of visualizing the collected data.
    - Subtask 2.3: Write a script to visualize, format, and classify the collected data in real-time.
- Task 3: Implement a text-to-speech function.
    - Subtask 3.1: Write a script to enable data flow between the neural network, speaker, Raspberry Pi, and text-to-speech software.
    - Subtask 3.2: Make the necessary hardware connections between the speaker, sensors, ADC converters, and the Raspberry Pi.
- Task 4: Implement a speech-to-text function.
    - Subtask 4.1: Write a script to enable data flow between the microphone, LCD display, and speech-to-text software.
    - Subtask 4.2: Make the necessary hardware connections between the microphone, LCD display, and Raspberry Pi.
    - Subtask 4.3: Design a simple GUI for the LCD display.

3.2 Team Organization

Team member 1, 2, …. Make sure all tasks/subtasks specified in last section are clearly assigned to **a** team member (if a subtask is too much for one person, split it to more subtasks). Later he/she will be evaluated based on the evaluation and completeness of tasks he/she is responsible for.

3.2.1. Responsibility of Team Member 1.

Task 1, Subtask 2.2, …

3.2.2. Responsibility of Team Member 2.

…

3.3 Timeline/Milestones/Delivery Plan

Please prospect the timeline to deliver the results of each task/subtask.

Please schedule your project to no more than 22 weeks from now.

Table 1. PROJECT TIMELINE AND DELIVERY PLAN

Time	Task	Comments	Responsible Personnel
Week 1	Start Subtask 1.2	Caller display coding. Three weeks needed.	Albert Einstein
Week 3	Finish Subtask 2.2	Delivery of the CAD file of the vehicle frame design.	Isaac Newton
…			
Week 22	Finish task 4.6, 9.10, …	System finalization and delivery. Finish all documentations and ready for presentation.	…

4. Implementation

For each task/subtask, create a section and add tech details of how it is implemented.

4.1 Implementation of Task 1.

…

4.1.1. Implementation of Subtask 1.1

…

4.2 Implementation of Task 1.

…

5. Conclusion (Discussion and Future Plans)

By the end of the project, conclude the project and your learning experience.

Acknowledgment

If you get help or support from someone else (besides the team member and the advisor) and want to show your appreciation, put here (**do not include the advisor**).

Appendix

You can put reference info here, including: i) specs of components used in the system, ii) source code (must be here but not in the body text), iii) CAD figures, etc.

A. Component Specs

1. Specs of Arduino Due

...

2. Specs of Raspberry Pi

…

B. Source Code.

1. Source Code of Graphic User Interface

…

2. Source Code of Robotic Arm

…

REFERENCES

[1] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino*, et al.*, "Corona: System Implications of Emerging Nanophotonic Technology," in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, 2008, pp. 153-164.

[2] X. Zhang and A. Louri, "A multilayer nanophotonic interconnection network for on-chip many-core communications," in *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, 2010, pp. 156-161.

[3] C. Batten, A. Joshi, J. Orcutt, A. Khilo, B. Moss, C. Holzwarth*, et al.*, "Building manycore processor-to-DRAM networks with monolithic silicon photonics," in *High Performance Interconnects, 2008. HOTI '08. 16th IEEE Symposium on*, 2008, pp. 21-30.

[4] Y. Pan, P. Kumar, J. Kim, G. Memik, Y. Zhang, and A. Choudhary, "Firefly: illuminating future network-on-chip with nanophotonics," in *IEEE/ACM Intl. Symp. on Computer Architecture (ISCA)*, 2009, pp. 429-440.

[5] N. Kirman, M. Kirman, R. K. Dokania, J. F. Martinez, A. B. Apsel, M. A. Watkins*, et al.*, "Leveraging Optical Technology in Future Bus-based Chip Multiprocessors," in *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*, 2006, pp. 492-503.

[6] J. M. Cianchetti, C. J. Kerekes, and H. D. Albonesi, "Phastlane: a rapid transit optical routing network," in *Proceeding of: 36th International Symposium on Computer Architecture (ISCA)*, 2009, pp. 441-450.

[7] A. Shacham, K. Bergman, and L. P. Carloni, "Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors," *Computers, IEEE Transactions on,* vol. 57, pp. 1246-1260, 2008.

[8] A. Shacham, K. Bergman, and L. P. Carloni, "On the Design of a Photonic Network-on-Chip," in *First International Symposium on Networks-on-Chip, 2007. NOCS 2007*, 2007, pp. 53-64.

[9] M. Kwai Hung, Y. Yaoyao, W. Xiaowen, Z. Wei, L. Weichen, and X. Jiang, "A Hierarchical Hybrid Optical-Electronic Network-on-Chip," in *VLSI (ISVLSI), 2010 IEEE Computer Society Annual Symposium on*, 2010, pp. 327-332.

[10] D. Ding and D. Z. Pan, "OIL: a nano-photonics optical interconnect library for a new photonic networks-on-chip architecture," presented at the Proceedings of the 11th international workshop on System level interconnect prediction, San Francisco, CA, USA, 2009.

[11] A. Joshi, C. Batten, K. Yong-Jin, S. Beamer, I. Shamim, K. Asanovic*, et al.*, "Silicon-photonic clos networks for global on-chip communication," in *Networks-on-Chip, 2009. NoCS 2009. 3rd ACM/IEEE International Symposium on*, 2009, pp. 124-133.

[12] D. Vantrease, R. Schreiber, M. Monchiero, M. McLaren, N. P. Jouppi, M. Fiorentino*, et al.*, "Corona: system implications of emerging nanophotonic technology," in *Proc. 35th IEEE/ACM Int'l Symp. Computer Architecture (ISCA)*, 2008, pp. 153-164.

[13] L. Zhang, E. Regentova, and X. Tan, "A 2D-Torus Based Packet Switching Optical Network-on-Chip Architecture," presented at the *IEEE International Symposium on Photonics and Optoelectronics* (SOPO 2011), Wuhan, China, 2011.

[14] L. Zhang, E. E. Regentova, and X. Tan, "Packet switching optical network-on-chip architectures," *Comput. Electr. Eng.,* vol. 39, pp. 697-714, 2013.

[15] G. Huaxi, X. Jiang, and W. Zheng, "A novel optical mesh network-on-chip for gigascale systems-on-chip," in *Circuits and Systems, 2008. APCCAS 2008. IEEE Asia Pacific Conference on*, 2008, pp. 1728-1731.

[16] G. Huaxi, X. Jiang, and Z. Wei, "A low-power fat tree-based optical Network-On-Chip for multiprocessor system-on-chip," in *Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE '09.*, 2009, pp. 3-8.

[17] Y. Yaoyao, X. Jiang, H. Baihan, W. Xiaowen, Z. Wei, W. Xuan*, et al.*, "3-D Mesh-Based Optical Network-on-Chip for Multiprocessor System-on-Chip," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on,* vol. 32, pp. 584-596, 2013.

[18] A. Shacham, K. Bergman, and L. P. Carloni, "Photonic networks-on-chip for future generations of chip multiprocessors," *IEEE Trans. Computers,* vol. 57, pp. 1246-1260, 2008.

[19] A. W. Poon, F. X. Xu, and X. Luo, "Cascaded active silicon microresonator array cross-connect circuits for WDM networks-on-chip," in *Proc. SPIE*, 2008, p. 689812.

[20] M. Lipson, "Compact Electro-Optic Modulators on a Silicon Chip," *IEEE Journal of Selected Topics in Quantum Electronics,* vol. 12, pp. 1520-1526, 2006.

[21] M. Lipson, "Guiding, modulating, and emitting light on Silicon-challenges and opportunities," *Lightwave Technology, Journal of,* vol. 23, pp. 4222-4238, 2005.

[22] T. Xianfang, Y. Mei, Z. Lei, J. Yingtao, and Y. Jianyi, "Wavelength-routed optical networks-on-chip built with comb switches," in *Photonics Conference (IPC), 2013 IEEE*, 2013, pp. 46-47.