Software Requirements Specification for Mechatronics: ASL Translator

Team #20, OpenASL
Robert Zhu zhul49
Zifan Meng mengz17
Jiahui Chen chenj194
Kelvin Huynh huynhk12
Runze Zhu zhur25
Mirza Nafi Hasan hasanm21

October 19, 2022

Contents

Revision History

Date	Version	Notes
October 05, 2022	1.0	Everyone -¿ Initial SRS Draft

1 Reference Material

1.1 Terms, Abbreviations, and Acronyms

Term, Abbreviation, or Acronym	Description
A	Shorthand for Assumption
ASL	Shorthand for American Sign Language. It is a form of sign language primarily used in the US and in parts of Canada
CFR	Shorthand for Camera Functional Requirement
CMC	Shorthand for carpometacarpal. This is the joint that connects your thumb to the rest of your hand
DIP	Shorthand for distal interphalangeal. This is the joint on your finger just before where your fingernail is
IP	Shorthand for interphalangeal. This is the joint just before where your fingernail on the thumb is situated
PIP	Shorthand for proximal interphalangeal. This is the next joint up your finger from where the knuckles are
MCP	Shorthand for metacarpophalangeal. This is the joint situated roughly where your knuckles are
ML	Shorthand for Machine Learning
MLFR	Shorthand for Machine Learning Functional Requirement
NFR	Shorthand for Non-Functional Requirement
CV	Shorthand for computer vision, computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos.
OpenCV	Shorthand for computer vision, computer vision is an interdisciplinary scientific field that deals with how computers can gain high-level understanding from digital images or videos.
TensorFlow	An open-source framework developed by Google, which enables machine learning, deep learning, and other statistical and predictive analytics

2 Introduction

2.1 Purpose of the Project

The purpose of our project is to create a device that will translate sign language gestures into their corresponding words or phrases. This will require the creation and development of a computer vision system alongside a machine learning model that will be used to recognize the hand motions, as well as a Raspberry Pi that will speak the word or phrase. The user will perform the sign language motion that will be captured by our computer vision system through a camera, and processed by our machine learning model and spoken through our Raspberry Pi.

2.2 Scope

2.2.1 In-Scope

The goals for our project are listed in the following table. The primary goals for our project include

- Accurate hand motion recognition: Tracking and recognition of the user's hands
- Real-time translation: Recognition and translation of user's hand gestures with minimal delay

Goals	Desciption
Reliable and Accurate Translations	The Sign Language Translator requires extensive training on the sensors to capture precise hand motion and ignore any human error on the user's part. The processing unit should be able to identify each letter within the American Sign Language using the data collected and transmit dialogue accurately to the user's request.
Real Time Translations	User's should never be required to wait an extensive period of time for the device to process their hand motion and provide a translation. The Sign Language Translator should simulate a real time conversation between regular people to deliver a seamless transition for other parties during presentations or social interactions.
Ease of Use	The user experience is crucial for a communication device. The Sign Language Translator should require minimal time and effort to set up. Once set up, the device should not require much maintenance or updates. Most importantly, the device should not hinder the user's ability to perform the gestures and hand motions of sign language.
Affordability	The Sign Language Translator should be affordable for the end users as to reduce the need of requiring an actual translator to accompany the user during their tasks. The device should remain functional whenever it is required to be used, and the hardware components of the device should be simple and cost-effective.
Customizable to User	As with language, different people might have a certain way of pronouncing a phrase or word and likewise the same could be said with Sign Language with slightly different gestures. The device should be able to adapt to the user and recognize the unique motions instead of forcing the user to slow down for the device.

2.2.2 Out-of-Scope

The stretch goals for our project are listed in the following table. These goals are out-of-scope for our project. They may or may not be achieved depending on our progress and time remaining in the academic year.

Stretch Goals	Desciption
Portable	The final device, while requiring OpenCV to scan and process hand motion, should become more portable and lightweight for the user to move around, so as to not interfere with the user's regular activities. The translator text to speech should become an application on all phone brands as for any user with the required equipment to be able to begin using.
Expanding to Different Languages	As a universal sign language does not exist at the moment, there exists deaf/mute individuals who use another form of sign language other than the American Sign language. These include the British, Australian and New Zealand Sign Language (BANZSL), the Chinese Sign Language (CSL), Arabic Sign language, and much more. The device should be able to understand and translate these new hand motions and generate a translation in their native language for this product to be used on a global scale.
Sign Language Education	The final device should be able to recognize the different hand motions and gestures of sign language in order to accurately translate them. This would make the device an excellent educational tool for those looking to learn sign language. The device could provide feedback and tell users how to improve their gestures using it's accurate hand tracking to help teach those unfamiliar with sign language.
Non-real Time Translations	The final product should be able to extract and recognize hand gestures from photos or videos uploaded by the users. In this case, if the users find online photos or videos related to sign language, they can upload them to application/software to acquire text-based translation. This could help the users learn sign language from online sources.

2.3 Usual Operations

The ASL translator will translate sign language gestures into their corresponding words or phrases to help the daily communication for people who have hearing problems. The camera on the device will actively detect the location and the motion of the hand gestures of the users standing in front of the camera. Then the computer vision and machine learning algorithm in the device will accurately translate the motion of the hands and output the correct English words or phrases. Then, the speaker on the Raspberry Pi which is connected to the translator will “speak out” the words or phrases.

2.4 Users and Stakeholders

The stakeholders for our project are people who have hearing problems and need to use sign language for their daily communication. This can also include various accessibility services for various companies, whether that be in education or entertainment. This can also be individuals looking to learn sign language as the device can be used as a practice tool to validate their signing. Our project can benefit anyone or anything that requires a sign language interpreter.

3 Project Constraints

3.1 Constraints

The project is constrained by the following:

- Translate a subset of the American Sign Language (ASL)
  - Training a machine learning model to encompass the entire ASL would be very time-inefficient for the time constraint of the project
- The project expenses cannot exceed $750 CAD
  - Additionally the project cannot be an off-the-shelf solution and be cost-efficient to attain our goal of affordability
- The project must be completed during the course of the academic year
  - This serves as a time constraint for the project and is also a requirement set by the course
- The area of detection is limited
  - Since there is only a specific field of view that can be detected by the camera, the users have to stand at the specific location to ensure the hand motion can be detected by the camera

3.2 Assumptions

Identifier	Assumption	Rationale
A1	Lighting will be sufficient enough for hand joint detection by camera	The CV system will require a camera to detect motion. That camera must be able to see the user clearly, which will require adequate lighting conditions
A2	The user is signing ASL and not any other sign language	The translator will only be able to recognize ASL and for such reasons will not recognize any other sign language which could result in a mistranslation or inaccuracy
A3	The camera or the device containing the camera is portable	Since the ASL is invented to help daily activities, the camera that is used to detect the hand motion should be compact and light

3.3 Undesired Event Handling

In the case of any undesired event, the device should alert the user of the failure or error that occurs. This will be crucial during testing as we will need it to ensure our project functions the way we would like it to. Some potential undesired events could include.

- Camera error
- Text-to-Speech failure
- Faulty training models

More undesired events will be added as development continues

4 Context Diagrams

Figure 1: Context Diagram

5 Functional Decomposition

5.1 Data Flow Model

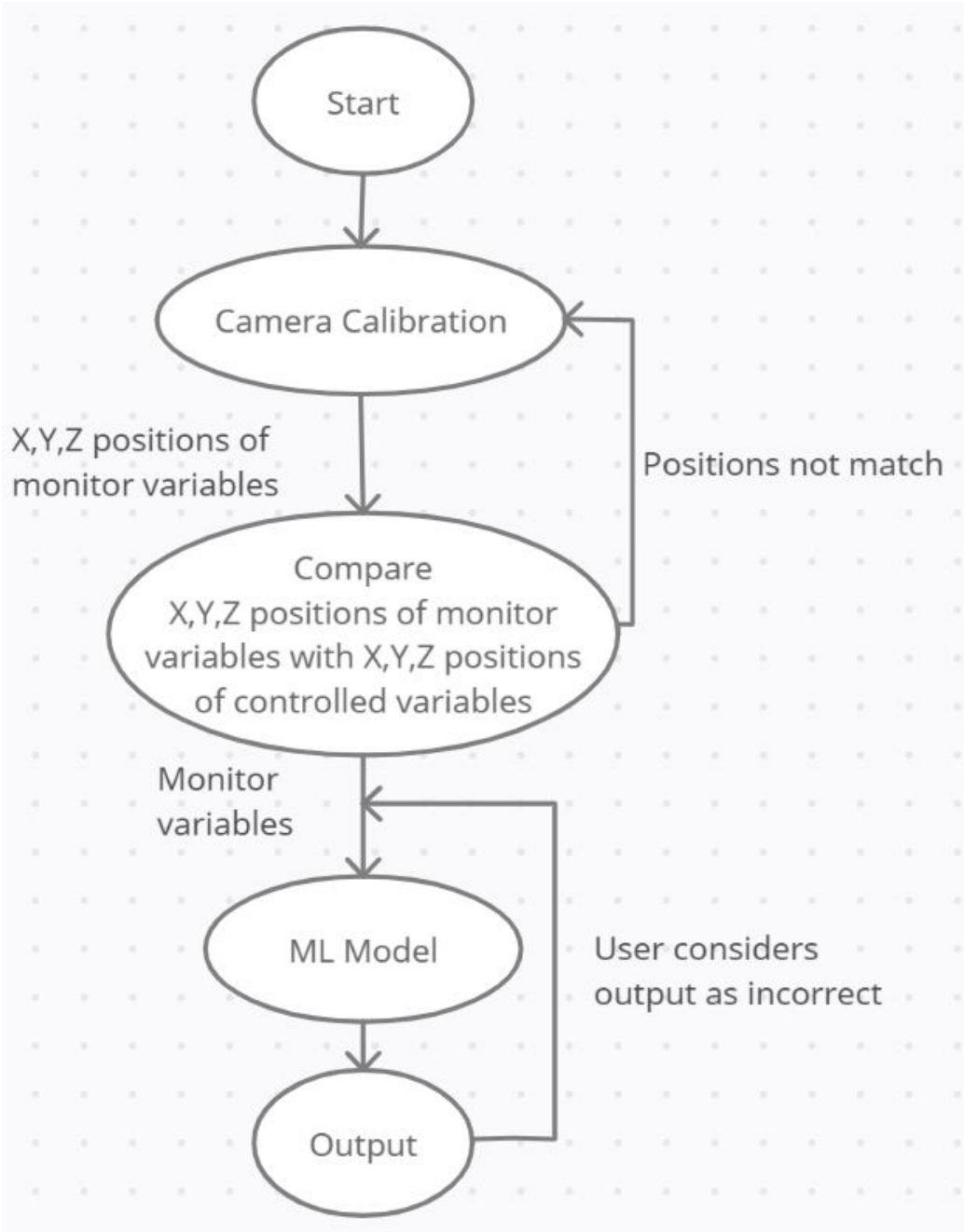

Figure 2: Data Flow Model

5.2 Monitor and Controlled Variables

5.2.1 Monitor Variables

Variable	Description
0. WRIST	Wrist
1. THUMB_JOINT_0	CMC joint of the thumb
2. THUMB_JOINT_1	MCP joint of the thumb
3. THUMB_JOINT_2	IP joint of the thumb
4. THUMB_TIP	Tip of the thumb
5. INDEX_JOINT_0	MCP joint of the index finger
6. INDEX_JOINT_1	PIP joint of the index finger
7. INDEX_JOINT_2	DIP joint of the index finger
8. INDEX_JOINT_TIP	Tip of the index finger
9. MIDDLE_JOINT_0	MCP joint of the middle finger
10. MIDDLE_JOINT_1	PIP joint of the middle finger
11. MIDDLE_JOINT_2	DIP joint of the middle finger
12. MIDDLE_JOINT_TIP	Tip of the middle finger
13. RING_JOINT_0	MCP joint of the ring finger
14. RING_JOINT_1	PIP joint of the ring finger
15. RING_JOINT_2	DIP joint of the ring finger
16. RING_JOINT_TIP	Tip of the ring finger
17. LITTLE_JOINT_0	MCP joint of the little finger
18. LITTLE_JOINT_1	PIP joint of the little finger
19. LITTLE_JOINT_2	DIP joint of the little finger
20. LITTLE_JOINT_TIP	Tip of the little finger

Variables above are monitor variables for the left hand, monitor variables for the right hand are similar, the only difference is variable names have _R at the end. For example, CMC joint of the thumb for the right hand is THUMB_JOINT_0_R.

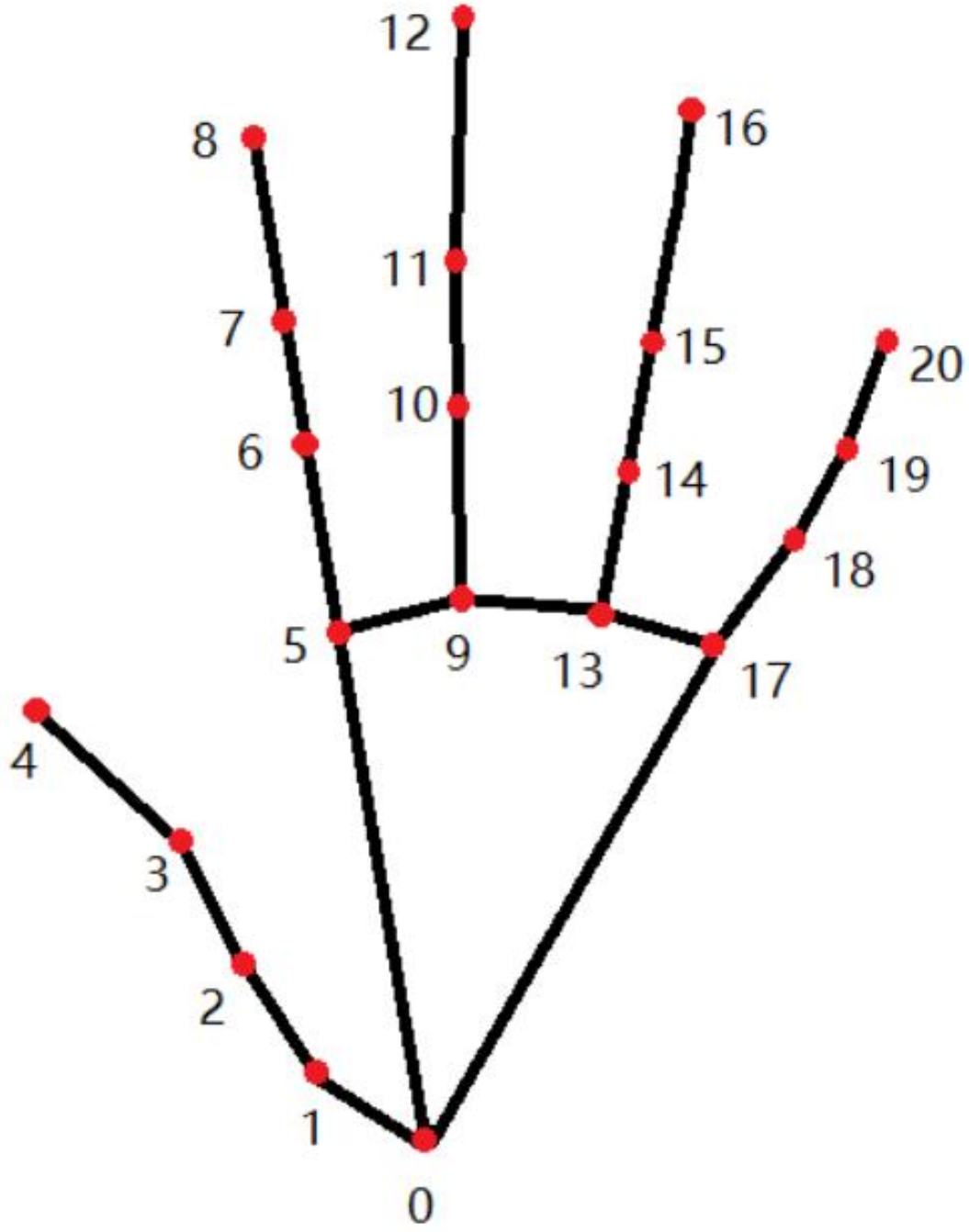

Figure 3: Hand Variables

5.2.2 Controlled Variables

Variable	Description
WRIST_CAL	Calibration point for wrist on the camera screen. The x, y, z location of this point must align with the x, y, z location of variable "WRIST" for successful calibration
INDEX_JOINT_0_CAL	Calibration point for MCP joint of the index finger on the camera screen. The x, y, z location of this point must align with the x, y, z location of variable "INDEX_JOINT_0" for successful calibration
LITTLE_JOINT_0_CAL	Calibration point for MCP joint of the little finger on the camera screen. The x, y, z location of this point must align with the x, y, z location of variable "LITTLE_JOINT_0" for successful calibration

6 Functional Requirements

6.1 Camera Functional Requirements

Identifier	Requirement	Rationale
CFR1	User hand gestures should be recognized and converted into input for the system	This is the primary and only way that the end user engages with the system. This is to ensure that their signing is picked up by the camera within a certain degree of accuracy
CFR2	The camera must be able to relay its vision back to the program	This enables validation and testing on the development end. In addition to understanding what needs to be corrected to ensure accurate user input

6.2 Machine Learning Functional Requirements

Identifier	Requirement	Rationale
MLFR1	The program should be able to recognize user hand joints	This enables the ML model to compare and match user hand gestures to ASL
MLFR2	The program should output the x, y, and z coordinates of each joint relative to the camera	This will enable system calibration and aid in enhancing predictive accuracy as the training data set will be primarily static images in contrast to the dynamic input from the end product
MLFR3	The program should recognize up to two hands in the input	The complexity of sign language calls for two hands to enable effective communication. Tracking one hand should also be considered as there are words in sign language that require the use of a single hand
MLFR4	The program should be able to process data in real-time	The translator should relay the relevant translation within a reasonable amount of time to ensure conversation fluidity
MLFR5	The program must be able to calibrate the camera	This is to ensure that the image being processed is undistorted and recognizable to the program to prevent inaccuracies and incorrect output from the ML model
MLFR6	The program should be calibrated to match the speed of the signer	The translator should be able to keep up with the user or the likelihood of a mistranslation will increase
MLFR7	The ML model should be easily trainable	This is how the ML model should learn sign language to use in processing. Making it easily trainable should enable expandability as well. In addition, this will enable the program to adapt to users' specific signing habits and allow for manual correction for the future

7 Functional Requirement Change Likelihood

7.1 Camera Functional Requirements

Identifier	**Likelihood of Change**	**Rationale**	**What May Be Changed**
CFR1	Unlikely	Input component of the system	Input may be changed to sensor instead of a camera
CFR2	Very unlikely	Enables testing and validation for the system	N/A

7.2 Machine Learning Functional Requirements

Identifier	Likelihood of Change	Rationale	What May Be Changed
MLFR1	Very unlikely	Key processing component of the system	N/A
MLFR2	Very unlikely	Enables testing and validation for the system	N/A
MLFR3	Unlikely	Subject to time constraint. ML model accuracy may be sub-par for two hand input	Tracking might only be possible with one hand
MLFR4	Unlikely	Subject to time constraint. Refer to 2.2.1	Data processing in real-time may be difficult and delays might have to be used to ensure translation is as accurate is possible
MLFR5	Very unlikely	Key processing component of the system	N/A
MLFR6	Unlikely	Key implementation aspect. Refer to 2.2.1	Dependent on the processing speed of the program. The speed at which a user can input sign language might be reduced consequently
MLFR7	Likely	Key implementation aspect. But expandability of the program is subject to time constraint and memory required	The ML model may only accommodate a subset of ASL for the sake of time constraint and space saving

8 Non-functional Requirements

8.1 Accuracy Requirement

Identifier	Requirement	Rationale
NFR1	The ASL translator should translate the sign language accurately	Since this device is used to help people who have hearing problems to communicate in their daily lives, the accuracy of the translator must ensure the normal communication
NFR2	The hand gestures of the users should be clear	In order for accurate translation by the ASL translator, the users should perform the sign language standardly and clearly

8.2 Useability Requirement

Identifier	Requirement	Rationale
NFR3	Ease of use	SThe device should be easy-to-use by the users, they should be able to use the translator directly since everything is preset
NFR4	Understandability	The instructions of the ASL translator should be clear and understandable by all the users
NFR5	Update the database timely and easily	Since new English words appear every year, the database should be updated yearly within half an hour

8.3 Portability Requirement

Identifier	Requirement	Rationale
NFR6	The size of the ASL translator should be portable	Since the ASL translator is used to support the daily communication of people who are hard of hearing, it should be carried easily by the users

8.4 Culture Requirement

Identifier	Requirement	Rationale
NFR7	Cultural requirement	As with language there are a variety of different forms of sign language that start from a common language, but over time different groups of people adapt to fit into their culture through variations of common English words. While it is unlikely to create a data set for every distinct variety of ASL for example, the machine learning function MLFR7 allows for easability in training for future updates to properly include differences in grammar and phonology

9 Phase-in Plan

Date to be Completed by	Objective	Description
Oct, 31, 2022	Get OpenCV to recognize Hands	In order for out project to work, first we must be able to recognize the joints in the hand
Nov, 30, 2022	Build a machine learning model	In order to recognize the different hand motions of ASL, we will first need to build a machine learning model that can recognize ASL
Dec, 31, 2022	Integrate the machine learning model and the OpenCV system	With both a machine learning model and hand recognition, we can begin to interpret ASL hand motions into words
Jan, 31, 2023	Build and connect Raspberry Pi to OpenCV system to provide real-time translation	With a working ASL translation system in place, we can begin to use text-to-speech to say the words
Feb, 28, 2023	Move OpenCV system and machine learning model onto Raspberry Pi	We can begin to try and make the entire system portable by moving it entirely to the Raspberry Pi

10 References

11 Appendix

11.1 Reflection

Kelvin will be responsible for learning TensorFlow (the framework for building a machine learning model):
The project will require the use of machine learning to generate a model for ASL such that user input in the form of sign language can be matched to a word in the dataset. In this aspect, it can be assumed that the goal of the project is to progress past machine learning and touch on deep learning for real-time image processing. It is also crucial that we are able to learn how to optimize a deep learning model such that it can fit onto a small device such as a Raspberry Pi. One approach to learning TensorFlow and how the framework can be used to build a model is simply reading through documentation. Although this is a viable way to learn TensorFlow, it is also not ideal as it becomes harder to ask questions and/or have visual examples for verification. Another approach would be to ask others who have dabbled or work with TensorFlow regularly, this way you can figure out the specific nuances of the framework and learn practical optimization techniques. The most ideal outcome would be to combine the use of these two approaches such that it is more applicable to the project.

Robert will be responsible for learning ASL:
Learning ASL and forms of sign language is a cornerstone requirement for the Capstone project. As with any spoken language, each pronunciation is fluent and experienced to communicate with other people. The Sign Language Translator needs to be tested for real world application, which is especially fast for ASL as each hand motion can range from a phrase to a letter and they require proper grammar to articulate the sentence. There are two practical ways to learn a language and that is through in-person interaction with an individual who understands the language fluently and online video tutorials in ASL. Online video tutorials will be the primary approach and the group does not include an individual who understands ASL. However, here at McMaster, the Student Accessibility Services are able to provide some insight to answer any questions about grammar and structure. This will not be the primary use as they often require appointments, so repeated practice through Youtube and forums should satisfy testing the Sign Language Translator in its accuracy and speed.

Jiahui will be responsible for learning mediapipe (the framework for incorporating OpenCV to machine learning):
For the project, the ASL translator should be able to use the camera to detect the motion of the hands and translate them into corresponding sentences, and technologies such as OpenCV and machine learning need to be involved to achieve the goals. Since the translator involves processing of video, which is time-series data, the framework Mediapipe will be used for building machine learning pipelines to meet the needs. One of the advantages about the Mediapipe is that it can work on various platforms such as Desktop/Server,

Android, iOS, Raspberry Pi etc. In particular Mediapipe Hands, which is a high-fidelity tracking solution for hand and finger will be adopted in the project. One of the approaches to acquire the knowledge of Mediapipe and machine learning is using the github website for Mediapipe, the website contains detailed information about the introduction, solution and tools for users who are new to Mediapipem and it is useful for beginners. In addition, many tutorial videos can also be found online to help learn the knowledge. Another approach to acquiring knowledge about machine learning is to ask professors who have a machine learning background. Online learning should be the main approach for learning Mediapipe, since it is difficult to find people who are professional in Mediapipe, and online tutorials are very detailed which can help solve most of the problems, and the study time can be flexible. Therefore, the online learning approach will be pursued for this project.

Nafi will be responsible for learning Raspberry Pi:
For our project, we will be using a Raspberry Pi to translate the hand gestures into actual words using text to speech.Eventually, we would like to have our entire project on a Raspberry Pi for portability, but as a first step, we will be looking to connect our OpenCV program to the Raspberry Pi. This will be the primary hardware component for our project. Combining software, and hardware components to create something is a core principle of mechatronics engineering. Learning how to work with a Raspberry Pi will help us learn about both hardware and software design and how both components interact and work with each other. One practical way to learn about Raspberry Pi would be to find tutorials online. Raspberry Pi has numerous tutorials for all sorts of projects online that may help with our project, or at least teach us more about how the board works. Another practical way to learn could be to ask people who have worked with or know about Raspberry Pi boards. The main method will likely be online tutorials, mainly because it is the simplest way to learn. It may be difficult to find people who know how to help us with our specific project.

Zifan will be responsible for learning OpenCV:
The main function of this project is to use computer vision to process and recognize ASL, in order to use machine learning framework and build models, it is important to understand how OpenCV works. OpenCV is a cross-platform computer vision and machine learning software library, it consists of a series of C functions and a small number of C++ classes, it provides interfaces in languages such as Python, Ruby, and MATLAB, and it can implement many general algorithms in image processing and computer vision. Some knowledge about image processing needs to be learned for understanding OpenCV, this involves the basics of linear algebra. digital image processing, signal processing, Fourier transform, wave function, etc. After that, one approach to learn OpenCV is to find online courses and tutorials, since OpenCV is an open source library and it is widely used, there are numerous resources about it, from theories to applications. Another approach can be communicating with people who are familiar with OpenCV, most people that work on CV use OpenCV, so it is not hard to find a professional. Both approaches will be used for this project, so that we can learn OpenCV efficiently.

Runze will be responsible for learning CI/CD:
For the project, each team member has to work on some parts of the coding independently and help the others to understand the code made by them. Therefore it is essential to learn Continuous Integration and Continuous Delivery, which is a coding philosophy and set of practices that can drive the team to frequently implement small code changes and keep delivering and deploying them for tests. Having a consistent integration and delivery process encourages all team members to commit code changes more frequently, which can lead to better collaboration, comprehension, and code quality. One approach to learn CI/CD is to find online courses and tutorials, since CI/CD is the fundamental cornerstone of DevOps, there are a lot of online resources and tools about it. Another useful approach is communicating with software developers who are experienced with this coding philosophy, many professional programmers need to apply the idea of CI/CD when working in teams, so it would be easy to get help from them. We can combine the two approaches, and it is supposed to be easy and helpful to carry on.