ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

An Intelligent Assistant Using Sign Language

Team #27

QIANZHONG CHEN (qc19@illinois.edu)
HANWEN LIU (hanwenl4@illinois.edu)
HAINA LOU (hainal2@illinois.edu)
YIKE ZHOU (yikez3@illinois.edu)

Sponsor: Prof. Yang, Liangjing
TA: Xiaoyue Li

May 23, 2023

Abstract

This project proposes a novel intelligent assistant to help people with speech or hearing impairments communicate and seek help. The intelligent assistant includes a bionic hand of 17 degrees of freedom (DOFs) and an innovative neural network that recognizes American Sign Language (ASL). The users can prompt a question in ASL, and the assistant would recognize the problem and search for the answer online, answering and helping the user with ASL co-generated by the microcontroller unit and the bionic hand. Meanwhile, the answer would be demonstrated on a digital screen for inspection.

Keywords: American Sign Language, MediaPipe, Self-designed dataset, Switch Model, GRU, Attention, Assistive robots, Bionic hand, Generative AI.

Contents

1 Introduction

1.1 Problem

An Intelligent Assistant (IA) is software that can provide services and interact with the user, typically by performing automated tasks and assisting with daily activities. With the advent of computer vision and natural language processing technologies and the emergence of innovative home accessories, intelligent assistants have revolutionized how people interact with technology. Most intelligent assistants use Voice User Interface (VUI) as a primary means of communication. Some of the most prevalent examples include Siri from Apple, Cortana from Microsoft, and Alexa from Amazon. VUI provides several advantages, such as hands-free operation, faster input, and greater convenience. However, VUI is not always suitable for people with hearing or speech problems, hindering their ability to use these intelligent assistants. People with hearing or speech impairments often face significant challenges in accessing information, participating in social interactions, and performing daily activities. Therefore, developing technologies that meet their unique needs and facilitate communication and engagement can significantly improve their quality of life.

1.2 Contribution

We propose to develop an intelligent assistant that uses sign language as its primary communication standard. Sign language will enable people with hearing or speech impairments to interact with intelligent assistants effectively. By leveraging the latest advancements in computer vision and natural language processing, our intelligent assistant will recognize sign language and respond in real time, making it a powerful and accessible tool for a broader range of users. Our intelligent assistant using sign language consists of four subsystems: Input and output subsystem, Gesture recognition subsystem, System and control subsystem, and Bionic hand subsystem. The input and output subsystem includes a camera that receives input from the user through sign language and a displayer that reveals the interaction between the user and the intelligent assistant to those who do not understand sign language. The gesture recognition subsystem receives the visual signal from the input and output subsystems. The gesture recognition subsystem utilizes *Mediapipe*[1], developed by Google, to collect the relevant position for the wrist in real time. Then, it extracts features with two methods, relative coordinate and rescaling, to maintain a small network. The system and control subsystem receives the decision of the gesture, which the bionic hand needs to do from the gesture recognition subsystem. It translates it to Pulse-Width Modulation (PWM) signals to control the movement of servo motors. Our control system uses Microcontroller Unit (MCU) to output signals and an advanced computing unit to deploy the machine learning model.

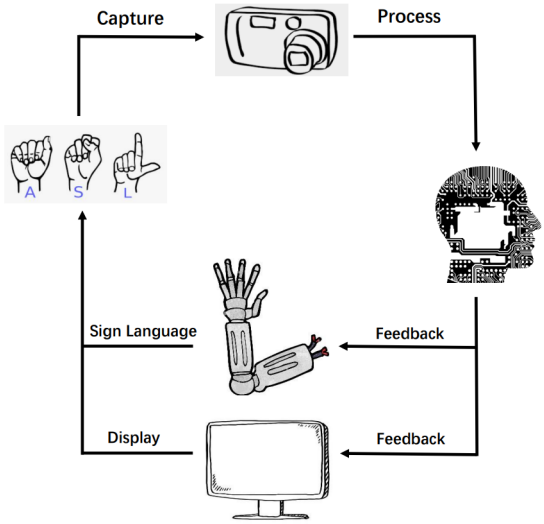


Figure 1: Visual Aid

1.3 Related Works and Information

1.3.1 Bionic Hand

Bionic hands combine cutting-edge technology with human adaptability and functional potential, representing a significant prosthetic advancement. A bionic hand is a complex mechanical limb designed to restore a lost hand's function that mimics the natural hand's complex movement and flexibility. A combination of sensors, actuators, and advanced control systems enables upper limb amputees to regain the ability to perform everyday tasks with great precision and ease. This revolutionary technology opens up new possibilities to enable amputees to regain their independence, improve their quality of life and integrate seamlessly into society. In this article, we will explore the fascinating world of bionic hands, delving into their structure, function, and the transformative impact they have on the lives of those who use them. A general introduction to bionic hand is provided in [2]. George et al. [3] highlighted the importance of tactile, proprioceptive, and kinesthetic feedback in enhancing the user's control and dexterity and the challenges associated with providing realistic sensory sensations. In [4], the authors described non-invasive and invasive technologies for conveying artificial sensory feedback through bionic hands and evaluated the technologies' long-term prospects. More emphasis was put on materials in [5] and examined the use of flexible and lightweight materials, as well as the development of bio-compatible interfaces for seamless integration with the user's residual limb.

1.3.2 American Sign Language

American Sign Language (ASL) is a rich and expressive visual language used primarily by deaf communities in parts of the United States and Canada. Unlike spoken language, which relies on sound, American Sign Language uses gestures, facial expressions,

and body movements to convey meaning and communicate effectively. With its unique grammar and syntax, American Sign Language is a complete and unique language with its linguistic structure.

In the deaf community, American Sign Language facilitates communication and fosters cultural identity. It is a means of communication for deaf or hard-of-hearing people, enabling them to engage in dialogue, express ideas, and participate in various social and professional environments. Fig.2 [6] demonstrated the fundamental expression of 26 letters and ten numbers in ASL.

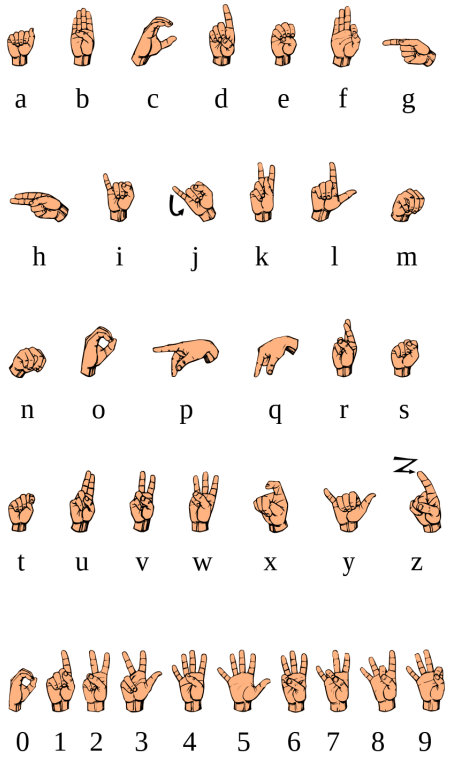


Figure 2: Basic expression of 26 letters and 10 numbers in ASL

1.3.3 Vision Recognition

Considering the cost-effectiveness and convenience, we proposed adopting computer vision-based techniques to detect objects rather than sensor-based ones. However, most computer vision-based methods consisting of gesture segmentation and hand shape estimation have high demands on high computing power, which indicates a high delay in a real-time recognition scenario within limited computing resources. Thus, we select MediaPipe[1], an open-sourced framework developed by Google, to detect the users' body movements. Compared to other object detection models like You Only Look Once (YOLO) [7], MediaPipe's computational cost on real-time recognition is relatively cheap. Also, considering our application scenarios that at most one user can interact with our product simultaneously, MediaPipe's single object detection suits our requirements. Fur-

thermore, MediaPipe can be deployed on personal computers and embedded platforms, like the Jeston Nano we used in the project. Figure 3[8] shows the hands' landmarks returned by the framework.

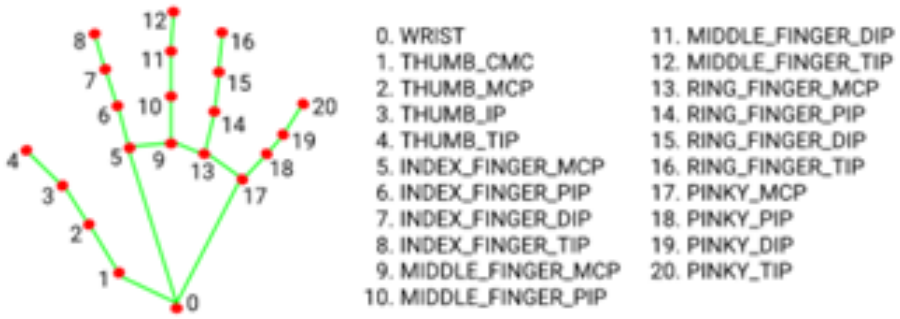


Figure 3: Hand's Landmarks

1.4 High-level Requirements

Build an end-to-end model using the Mediapipe framework combined with different machine learning models, including Support Vector Machine (SVM), Long Short-Term Memory (LSTM), and Gate Recurrent Unit (GRU). To implement a good interaction experience, the time used by the user doing sign language to display the dialogue and the response of the bionic hand should be at most 30 seconds. The bionic hand can move free and fluently as designed, all of the 12 degrees of freedom fulfilled; the movement of a single joint of the finger does not interrupt or be interrupted by other movements; the bionic hand could work for one hour in a roll and two years in total.

1.5 Block Diagram

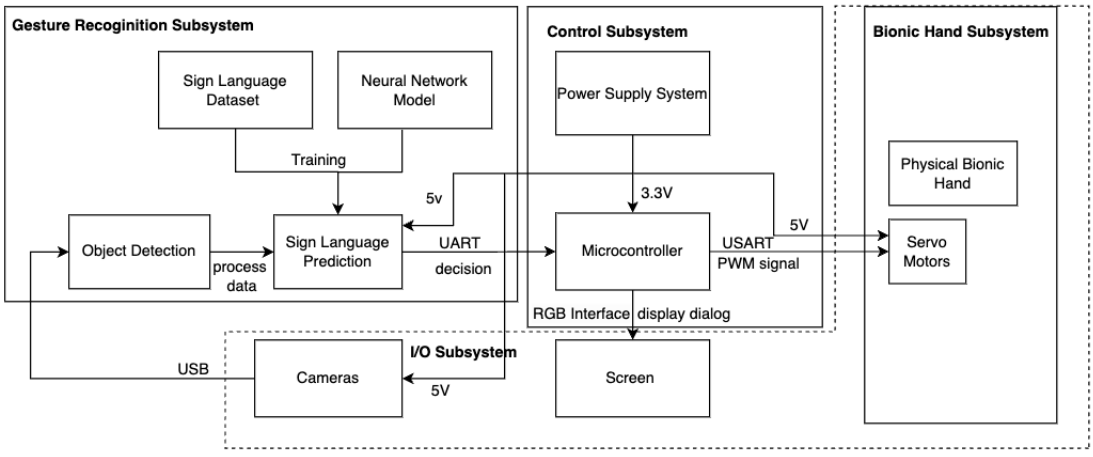


Figure 4: High Level System Overview

2 Design

2.1 Physical Design

In this part, we demonstrate the system overview in Fig. 5 and Fig. 6. Our system includes one bionic hand, one camera for gesture recognition, one STM32 microcontroller, one NVIDIA Jetson Nano developer kit[9], and one Liquid Crystal Display (LCD) screen to visualize the information.

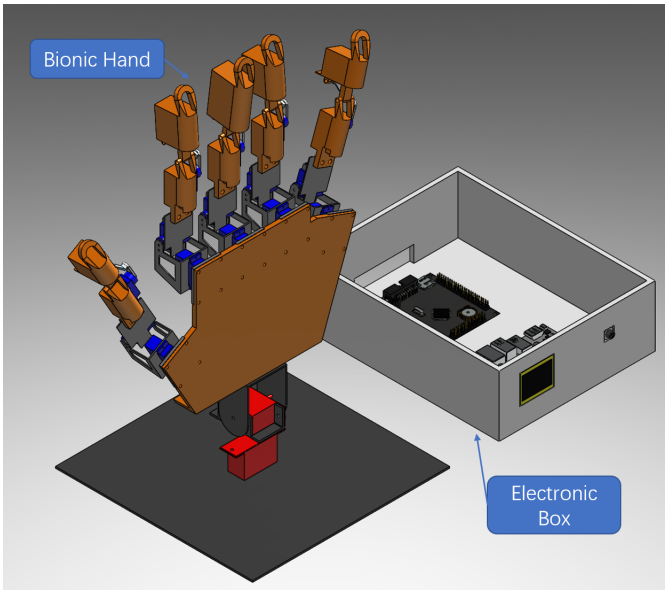


Figure 5: Overview of the System Physical Design

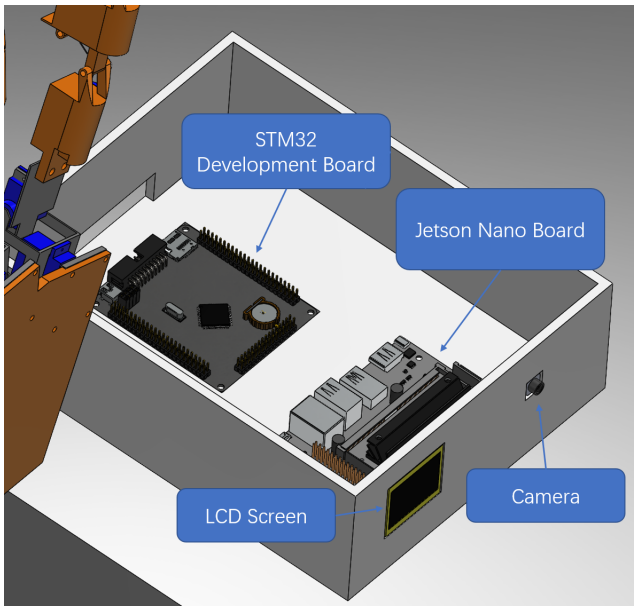


Figure 6: Detailed Layout of Electronic Box

2.2 Bionic Hand Subsystem

2.2.1 Description

The bionic hand subsystem comprises a bionic hand and an electrical platform, forming a system with 17 DOFs. The bionic hand subsystem is responsible for delivering the motion planned by the control subsystem and interacting with the user directly. From bottom to top, the hand has a moveable platform, five fingers with 3 DOFs each, and a palm. The combination of fingers' movements will form different gestures. The bionic hand comprises 71 parts, as shown in Fig.7

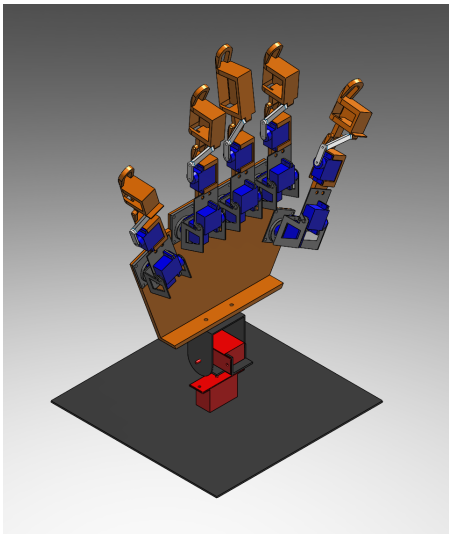


Figure 7: Engineering Drawing of the Bionic Hand

2.2.2 Moveable Base Platform

Two RDS-3115 digital servo motors drive the moveable base platform, hold the plastic bionic hand, and provide two extra DOFs. The platform is an off-the-shelf product.

2.2.3 Finger

The finger can be divided into two parts, a small platform with 2 Degree of Freedoms (DOF) acts as the underneath part of the finger, and the fingertip is held on the platform and driven by a four-bar linkage. Each finger uses 3 MG-90 small servo motors. The platform can bend the finger and rotate to the left and right. For the tips, the MG-90 servo motor is directly fixed inside the fingers, and its output shaft will connect the finger's moveable part with a linkage and drive the finger to rotate around the joint. Therefore, the finger part, motor, and linkage will form a basic 4-bars link system and move smoothly. The small platform is an off-the-shelf product, and we manufactured other parts with 3D printing using PLA material.

2.2.4 Palm

The palm is designed to hold the pedestal of five small platforms, which also fixes the assembly position of five fingers. The design of the palm simulated with human-beings real hand structure and shape, also leaving each 3-DOFs finger sufficient space to conduct motion in 2 directions. The palm was manufactured with 3D printing using PLA material.

2.2.5 Requirements and Verification

All of the requirements have been achieved with verification items. For detailed information, please check Table. 3 in Appendix A.

2.3 System & Control Subsystem

2.3.1 Description

The control subsystem is aimed at translating from sign language predictions to PWM signals and delivering the signals to 17 servo motors, making it possible to communicate and control the whole system. It is like the bridge between a high-level decision-making system and a mechanical part. It contains one development board with an STM32 microcontroller and a computing unit that supports real-time gesture recognition. The block diagram of the control system is shown in Fig. 8.

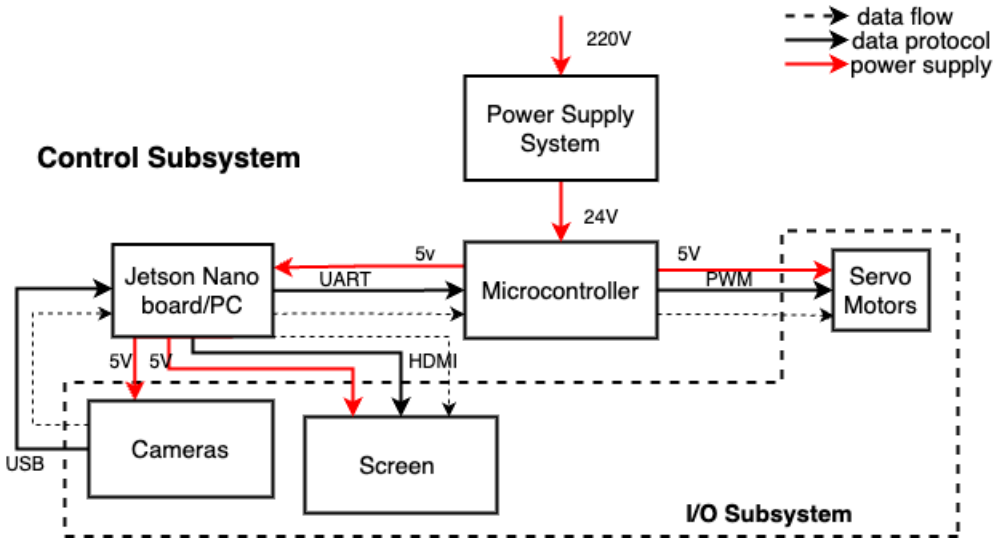


Figure 8: Block Diagram of Control System

2.3.2 Connection with Other Subsystems

- Connection to the power source: Our development board is driven by 24V.

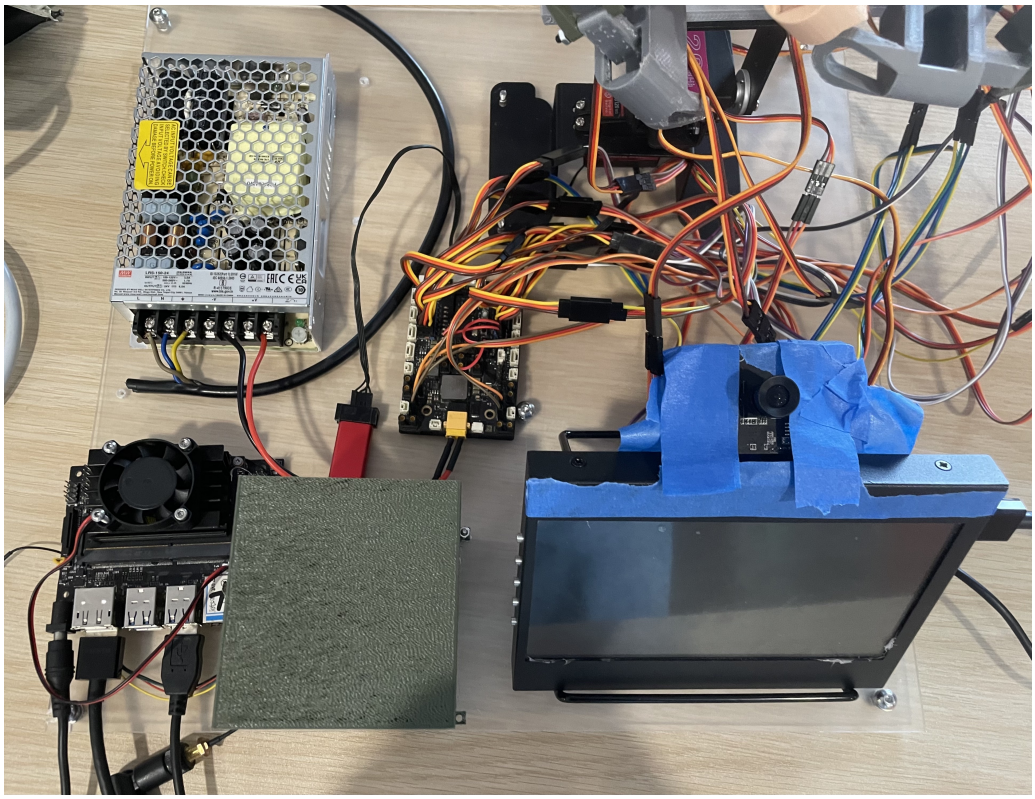


Figure 9: Top View of Control System

- Connection to the gesture recognition subsystem: Receives gesture predictions from machine learning model.
- Connection to the bionic hand subsystem: Output 17 PWM signals to each servo motor to control the motion of the bionic hand.
- Connection to the input and output Subsystem: The camera will be connected to our computing unit to capture the gesture, and LCD will be connected to an upper computer via USB.

2.3.3 Microcontroller & Development board

We choose Robomaster Development Board Type A as our development board. It has a powerful microcontroller STM32F427IIH6 which can be operated on ChibiOS real-time operating system. It controls the motion of 17 servo motors on the bionic hand by sending PWM signals. The specific motion signal should be generated based on the decision from the computing unit Jetson Nano or personal laptop through Universal Asynchronous Receiver/Transmitter (UART). We use pin PD12 - PD15 to output 4 PWM signals generated from Timer 4, pin PH10 - PH12 and pin PI0 to output 4 PWM signals generated from Timer 5, pin PI2, and pin PI5 - PI7 to output 4 PWM signals generated from Timer 8 and use pin PA8, PA9, PE13 to output 3 PWM signals generated from Timer 1. Those 15 signals are used to control the 15 servo motors on five fingers. The two bigger servo motors

are controlled by PWM signals from pin PA2 and PA3 generated by Timer 9.

Design Alternatives
There are several design alternatives related to the microcontroller and development board. We plan to use Arduino boards at the beginning. However, our development boards have more powerful processors than Arduino boards, which can be beneficial when controlling many servo motors. Arduino outputs 3.3 V, which is smaller than servo motors' 5 V operating voltage. Arduino has less memory and performs less efficiently doing calculations. Besides, considering the expandability, we choose a development board since we want to further develop a smart, intelligent lighting system for our project. For the development board, it is easier to add new components like the Bluetooth HC05 module in the future.

2.3.4 Computing Unit

We use Jetson Nano as our computing unit in this project. Jetson Nano is a small, powerful processing unit that can run multiple neural networks in parallel for applications like image classification, object detection, and speech processing [10]. We use this platform to deploy our sign language recognition model. It will intake data from the camera unit and perform computing onboard after getting the result from the gesture recognition subsystem. The computing unit, powered by a development board, will send all decisions to MCU via UART.

Design Alternatives
We test both Jetson Nano and our laptop as our computing unit. The performance depends on their computational resources. Personal laptops have good performance in both static and dynamic gesture recognition processes. We use GPU on Jetson Nano to run our trained model, it has excellent performance for static gesture recognition.

2.3.5 Control Program

The workflow of our control program is shown in Fig. 10. Three threads are used in the control program, including the communication, hand control, and one main thread. The communication thread listens for messages transmitted from the upper computer, which will be stored in a shared buffer. The hand control thread continuously loops through the message in the shared buffer and extracts each character. It then finds the corresponding PWM signal outputs for 17 motors in the matrix, which contains all motor position data for 27 movement sets. For letters like S, M, N, and T, the order of finger bending is crucial to succeeding since we want to hide the thumb into four other fingers to complete those gestures. The order for the movement of each finger is also stored in one matrix in our program. After deciding the movement order, the hand control thread starts the PWM driver and output signals.

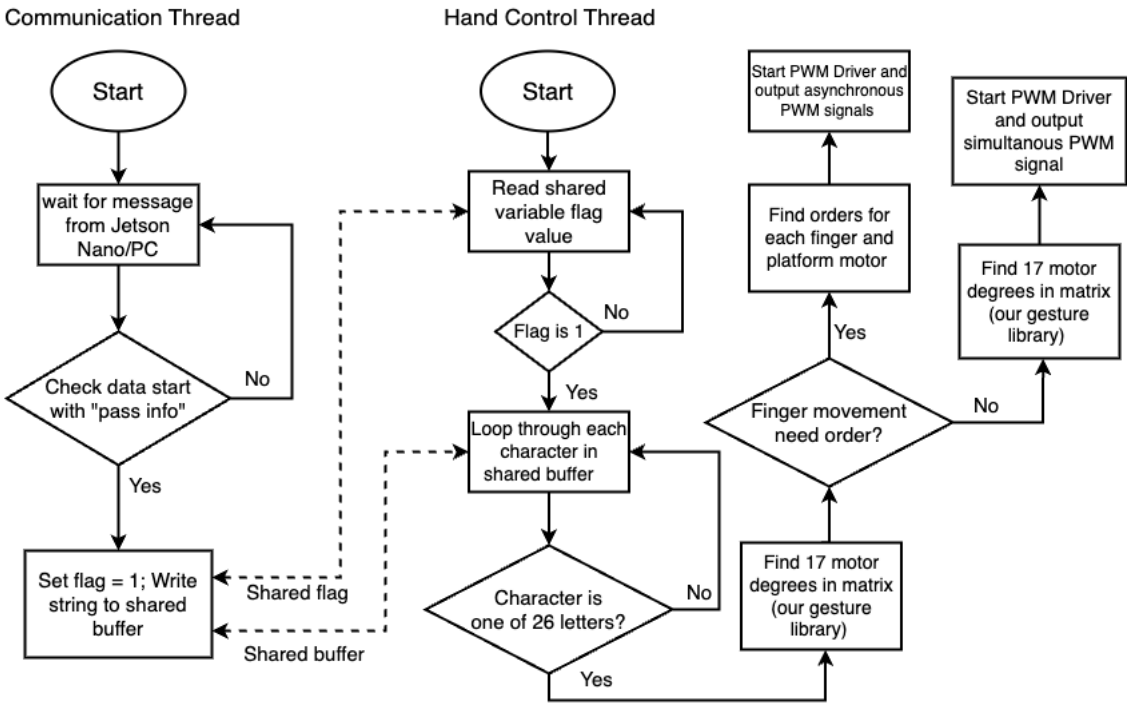


Figure 10: Workflow of Control Program

2.3.6 Requirements and Verification

All of the requirements are achieved with verification items. For detailed information, please check Table. 5 in Appendix A.

2.4 Input & Output Subsystem

2.4.1 Description

The input and output subsystem includes one camera and one screen. The camera module captures the user's hand gesture as input data. The screen displays sign language dialogs to other people as text to help people who do not know sign language learn the conversation between the intelligent assistant and the user.

2.4.2 Connection with other subsystems

- Connection to the power source: The camera and screen are connected to a 5V power source. Jetson Nano board supply the camera and screen power.
- Connection to the control subsystem: The camera inputs hand gesture data through a USB wire, and the screen displays sign language dialogues.

2.4.3 Camera

A USB camera is used for capturing real-time hand gestures of the user. The input resolution should be 3280*2464 with a 79.3 field of view.

2.4.4 Screen

When a user uses sign language to communicate with our assistant, not only our bionic hands will give users feedback, but also the dialog between users and the assistant will be displayed on screen as text. It can be connected to the development board via HDMI.

2.5 Gesture Recognition Subsystem

2.5.1 Description

The gesture recognition subsystem is mainly used to extract features from the images passed by the camera and then feed them into a pre-trained model for further prediction. Our ultimate goal is to design a real-time detection subsystem that can be as fast as possible within the constraints of limited computing power. This part will discuss how each part in the following workflow is designed to fit the requirements and improve our models compared to the existing methods.

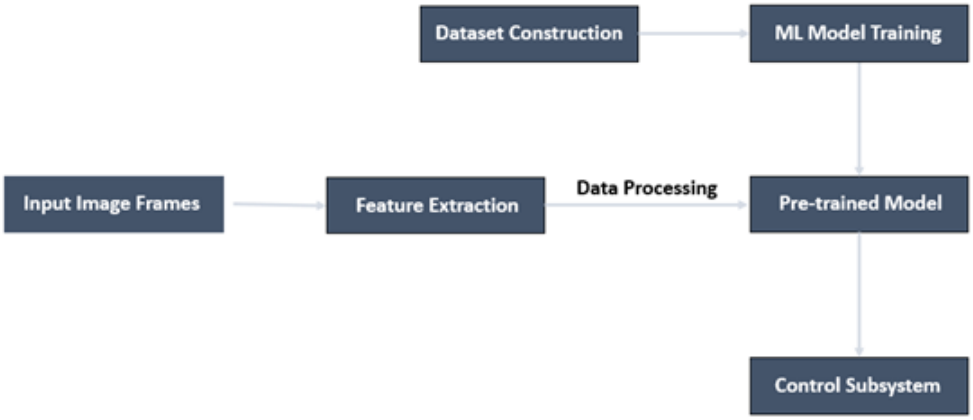


Figure 11: Gesture Recognition Subsystem Workflow

2.5.2 Object Detection and Feature Extraction

After using MediaPipe to obtain the 3D coordinates, we still need to process data before feeding them into our machine-learning model. For static recognition, since we recorded our dataset, we have a small amount of data compared to other open-source datasets on the internet. Thus, we put forward two different features to help the model learning. The first feature is relative coordinates. Because we cannot predict the position of the user's hand, our recognition subsystem must determine the same ASL gesture shown up in different positions with the same meaning. Furthermore, the relative coordinates will take

the point on the wrist as the reference. In this case, the subsystem can eliminate the effect results from the position factor. Besides the position, we cannot fix the user's hands' distance from the webcam, which may cause different palm sizes. In addition, different users' palms sizes must be different. To help our model better learn this situation, we proposed the second feature of rescaling, which will rescale the length between two points. The rescale factor η is determined by the area of the smallest rectangle that can enclose the hand. We first calculate the average palm area in training data as S_{avg} and the real-time detected palm area as S_{det}. Then, η could be calculated by Equation 3. We could adjust the length between two points by multiplying the rescaling factor. The validation of those two features is in the next section.

$$\eta = \sqrt{\frac{S_{avg}}{S_{det}}} \tag{1}$$

2.5.3 Static Recognition Model

We will not utilize complicated neural networks in static recognition tasks because traditional machine learning models can achieve well enough results. We adopted Support Vector Machine (SVM), which performs better in high-dimensional space than other traditional models[11]. The optimization problem solved by SVM can be described as two equations:

$$\min_{(b,d,e)} \frac{1}{2} w^T w + C \sum_{i=1}^{n} d_i \tag{2}$$

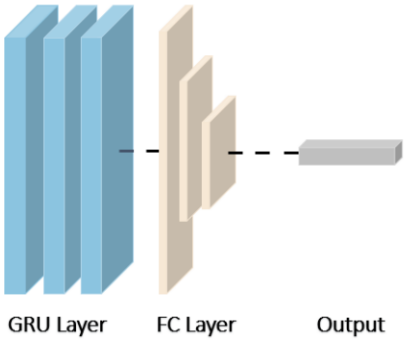
$$y_i(w^T \phi(x_i) + b) > 1 - d_i \tag{3}$$

The kernel function is the key to SVM. For nonlinear models, we need to use nonlinear mapping to transfer the data from low to high dimensions and then perform linear classification in the high-dimensional feature space. However, the computational complexity will grow exponentially as the number of variables increases. However, the kernel function can solve this problem. Its computation process is done in low-dimensional space. Then it represents the classification effect in high-dimensional space so as to avoid the complex computation in high-dimensional space. After testing the model in the different kernels, we found linear kernel fits our dataset best. The detailed table will be attached in the verification section.

2.5.4 Dynamic Recognition Model

In the case of dynamic recognition, we proposed adopting recurrent neural networks (RNNs), which contain memory storing information from previous states' computations. Thus, they can deal with time series and sequential data. While traditional RNN may occur problems of gradient vanishing, we first adopted Long Short-Term Memory (LSTM), a variant of RNN, to prevent the potential risk. However, due to the limitation of computing resources and lack of data, we changed LSTM with Gate Recurrent Unit (GRU),

which performs similarly to LSTM but with fewer parameters. Compared with LSTM, the substantial changes in the GRU network are integrating Forget Gate and Input Gate in LSTM into an Update Gate, whose role is to decide what to discard old information and what to add new information. Thus, GRU also can filter out useless information and capture the long-term dependencies of input data. Nevertheless, the number of parameters is reduced by a third, and in some cases, the dropout loop can be omitted[12]. Fig12 demonstrates the GRU structure we developed. There are three GRU layers, followed by three Fully Connected layers. Furthermore, after a softmax function, the model will output the possibility of each class.



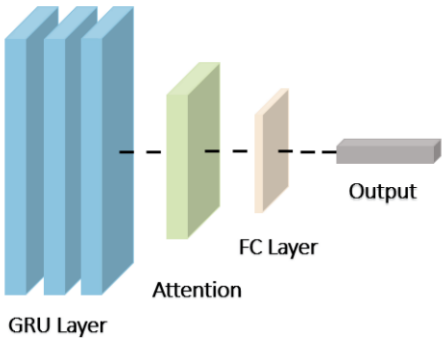
Layer	Output Shape
GRU1	(None, 30, 32)
GRU2	(None, 30, 64)
GRU3	(None, 32)
Dense1	(None, 32)
Dense2	(None, 16)
Dense3	(None, 6)

Figure 12: GRU Structure

Finally, we propose the structure of GRU with the Attention mechanism. The attention mechanism allows us to concentrate limited attention on crucial information, thus saving resources and getting the most helpful information quickly. In our task, some segments of movements play a decisive role in indicating their meaning; this mechanism will allocate more computing power to this part of the movements. By introducing attention, we can further reduce the number of parameters and simplify the model, which means the model is more straightforward and has the following merits: First, our system will have a shorter response time in a real-time recognition scenario. Secondly, it can save more computing resources so the model can be deployed on develop board or Web with less cost if needed. Thirdly, a simpler model means that it has shorter training time and thus higher training effectiveness, reducing the time cost of retraining. Fig13 demonstrates our developed GRU + Attention structure. There are also three GRU layers, followed by one Attention layer and one Fully Connected layer.

2.5.5 Switch Model

As we do real-time sign language detection, low latency has become our pursuit of giving users a better experience. In the last part, we have mentioned both static and dynamic models. Theoretically speaking, all the recognition can be completed by dynamic recognition. However, the latency will become significant if we keep using dynamic due to the high model complexity. At the same time, although the static model has a low delay in detection, it cannot handle all the situations.



Layer	Output Shape
GRU1	(None, 30, 4)
GRU2	(None, 30, 8)
GRU3	(None, 4)
Attention1	(None, 4)
Dense1	(None, 6)

Figure 13: GRU + Attention Structure

Thus, we put forward a switch model consisting of static and dynamic recognition to classify all the ASL successfully and get faster responses within limited computing resources. The basic logic is that we prefer SVM to do the task for sign languages that can be recognized statically. We will use the dynamic one to accomplish the classifications only for those that cannot be recognized by static recognition.

3 Verification

3.1 Bionic Hand Subsystem

In modern mechanical and product design, Computer Aided Engineering (CAE) and product iteration are essential to design verification practices. We conduct dynamics simulation for our four-bar linkage and statics simulation toward critical parts' structural reliability. Throughout the whole project, we have had five significant design iteration versions.

3.1.1 Computer Aided Engineering

The core system can be abstracted to a four-bar linkage for motion smoothness. Therefore, detailed kinematics and dynamics simulation is necessary. Initially, we referenced the simulation tool used in UIUC TAM 212 course[13]. Due to the space limitation, we had the presupposed ground link length (g) and input link length (a). By changing the output link length (b) and floating link length (f), we have combinations of four-bar linkages that can deliver satisfied trajectories on the output node, which links to the finger part driven by the motor. We conduct dynamics simulation for those candidate four-bar linkages using the MATLAB program developed in UIUC ME 370 lab and project. By calculating the position, velocity, acceleration, and torque on the output node for the whole cycle, we chose the best four-bar linkage that suffers little impact and vibration. The kinematic and dynamic simulation results are shown below in Fig.14 and Fig. 15. The formulas we used are listed below:

$$\text{Grashof index:} \quad \boldsymbol{G} = \boldsymbol{s} + \boldsymbol{l} - \boldsymbol{p} - \boldsymbol{q} \geq 0 \tag{4}$$

$$\text{Validity index:} \quad \boldsymbol{V} = \boldsymbol{l} - \boldsymbol{s} - \boldsymbol{p} - \boldsymbol{q} \geq 0 \tag{5}$$

where $\boldsymbol{l}$ is the longest link, $\boldsymbol{s}$ is the shortest link, and $\boldsymbol{p}$, $\boldsymbol{q}$ are the rest two links.

$$\begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ -R_{12y} & R_{12x} & -R_{32y} & R_{32x} & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & R_{23y} & -R_{23x} & -R_{43y} & R_{43x} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & u & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} F_{12x} \\ F_{12y} \\ F_{32x} \\ F_{32y} \\ F_{43x} \\ F_{43y} \\ F_{14y} \\ T_{12} \end{bmatrix} = \begin{bmatrix} m_2 A_{CG2x} \\ m_2 A_{CG2y} \\ I_{CG2}\alpha_2 \\ m_3 A_{CG3x} - F_{Px} \\ m_2 A_{CG2y} - F_{Py} \\ I_{CG3}\alpha_3 - R_{Px}F_{Py} + R_{Py}F_{Px} \\ m_4 A_{CG4x} \\ 0 \end{bmatrix} \tag{6}$$

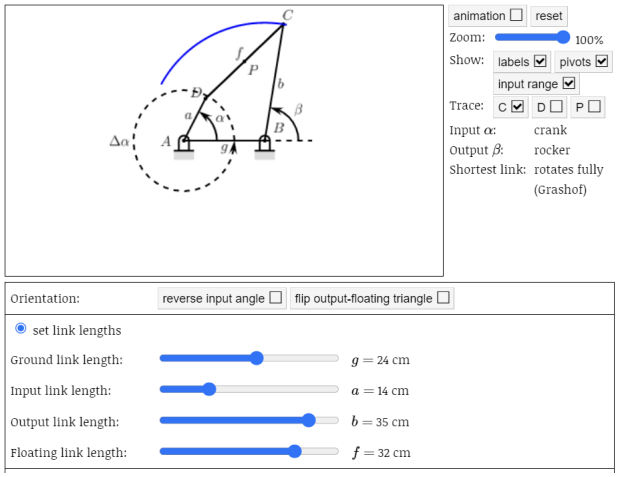


Figure 14: Kinematics Simulation Results of Proposed Four-bar Linkages

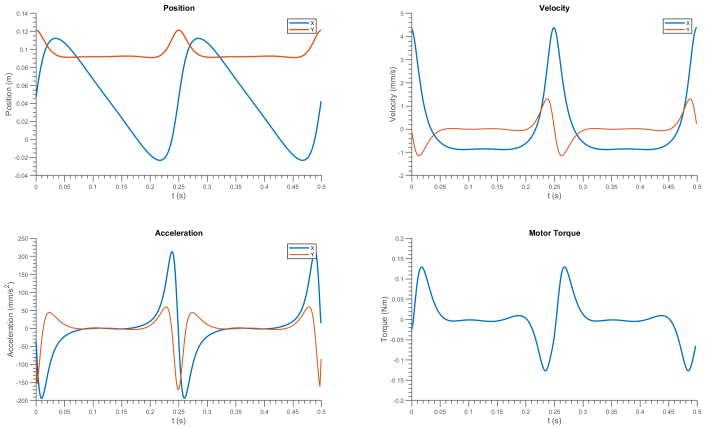


Figure 15: Dynamics Simulation Results of Proposed Four-bar Linkages

For critical parts, we conduct statics simulation and failure analysis with CAE software to verify and improve our design. For instance, as demonstrated in Fig .16, the palm we initially designed has right-angle sides, which may cause stress concentration and even yield under impact. Afterward, we strengthen the part and fillet the right-angle sides. The optimized design performs much better under the same load conditions shown in Fig. 17.

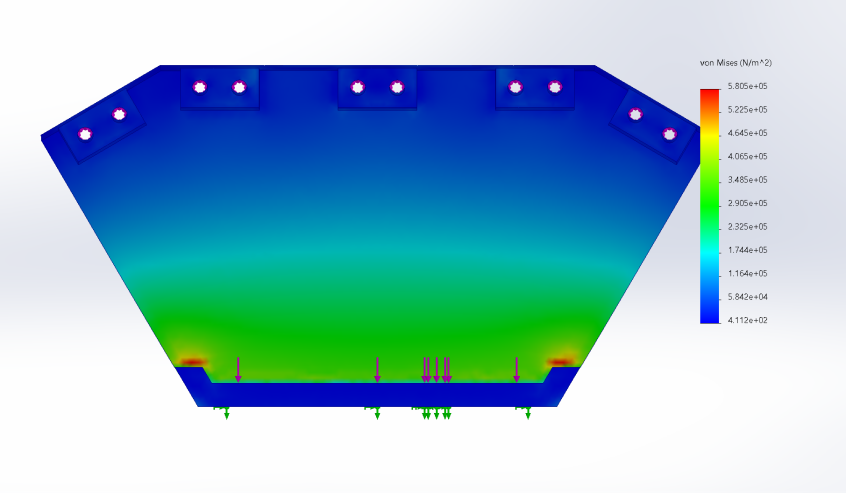


Figure 16: Statics Simulation for Initial Palm

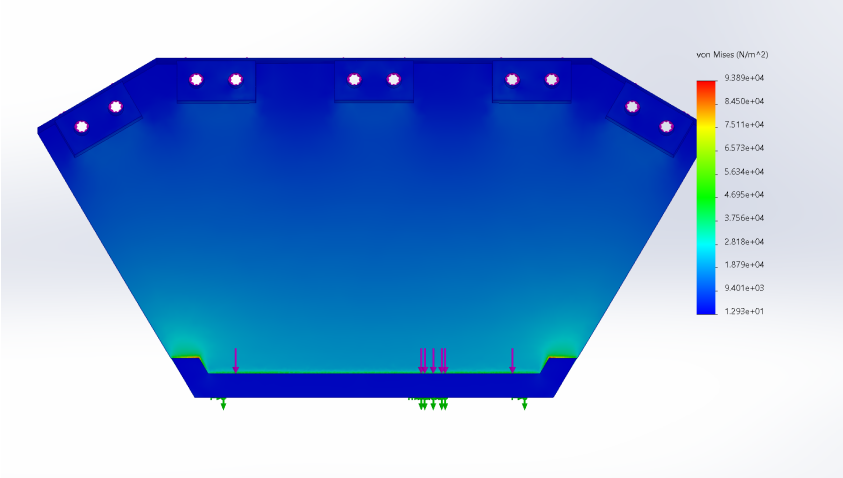


Figure 17: Statics Simulation for Optimized Palm

3.1.2 Iterations

As shown in Fig.18, along the project, we iterated five major versions of the design in total. Each discarded version failed for specific reasons. This section demonstrates a brief discussion of each discarded version's drawbacks.

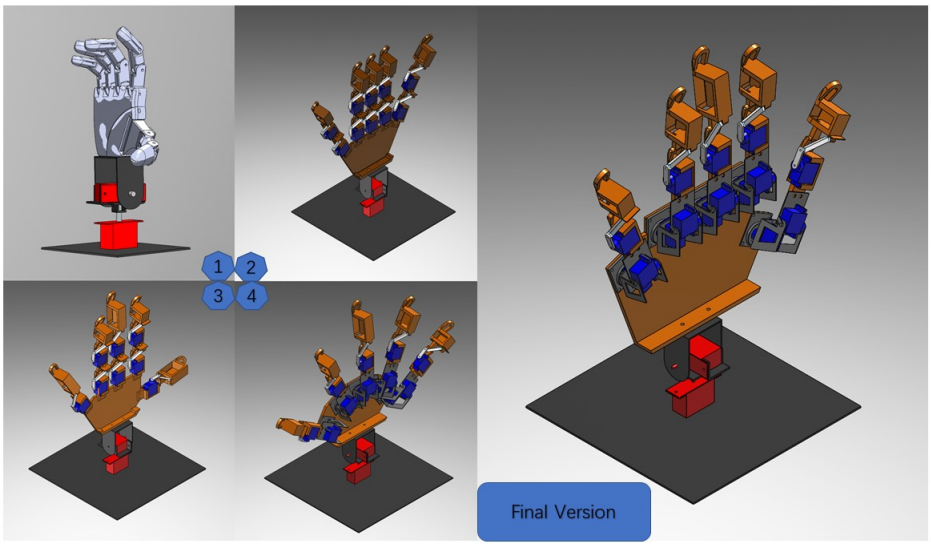


Figure 18: Design sketch of all of our bionic hand versions

Version 1 uses strings to drive and bend each finger. However, with this design, the assignment and management of strings are critical problems, and it cannot fulfill the independent control of each finger joint. What is worse, the finger cannot rotate to the left and right on the palm surface like everybody's hand. Because of the problems, we did not finish manufacturing this version and turned to the new design. Starting from Version 2, we gave up strings and used servo motors to drive the finger parts to rotate directly. On the Version 2 hand, we had six servo motors to fulfill 6 DOFs, all using four-bar linkages to drive and rotate. The motion of each finger is reliable, but different fingers very quickly collide and conflict with each other due to the poor design of the palm. In Version 3, we optimized the palm design for this version and fixed the hand on a 2-DOFs platform. To represent gestures of 26 letters in ASL, we found that we must have an extra DOF on each finger to rotate on the palm surface centered with the linking axis. We started to use a small platform in Version 4 to fulfill the function that the finger can rotate on the palm surface centered with the linking axis. Meanwhile, the 2-DOFs small platform also replaces the base part of each finger to make the finger bend for a greater angle. Due to the increasing size of the small platform compared with the initial design, the palm needed to be larger to hold every finger and avoid collision.

3.2 Control Subsystem

The verification of the control subsystem includes three parts. For the controlling and communication parts, we set the requirement that the delay from the microcontroller receiving the decision message from the upper computer to outputting the corresponding PWM signal should be less than 1 second. Furthermore, the development board should get correct messages from the upper computer for the serial communication part. To verify the requirements, we directly transmit letters from the upper computer to develop-

ment board using serial communication Python script to control programs. In the control program, We set the LED to turn green when the development board receives a message. To compute the delay between receiving the message from the upper computer and hand movement, we can record the time the green light turns on and the hand starts to move. It turned out that the green light and the movement of our hands happened simultaneously. We show our verification result in Table. 1

Table 1: Verification for Time Cost and Accuracy

Letter	L	A	S	E	V
Input Times	20	20	20	20	20
Correct Message Times	20	20	20	20	20
Cost Time	less than 1s	less than 1s	less than 1s	less than 1s	less than 1s
Accuracy	100%	100%	100%	100%	100%

The second requirement is that our product has the ability to output 27 separate sets of signals, standing for a whole alphabet and a default position, output 17 stable PWM signals simultaneously at 50 Hz frequency, and output asynchronous PWM signals output for performing letters M, N, S, T. The verification process includes testing all letters from A to Z and asking people who understand ASL to identify each letter from our hand movement. The results are successfully shown in Appendix A 22

3.3 Gesture Recognition Subsystem

3.3.1 Static Recognition model

This part will compare SVM with different kernel functions and some other traditional machine-learning models. Furthermore, discuss the performance of SVM with linear kernel in real-time recognition scenarios. Fig19 shows the accuracy of different traditional machine-learning models based on the open-sourced dataset[14] before and after adopting the two features we proposed. After comparison, all models' accuracy increased after data processing. The result shows that the feature we set is reasonable and feasible. Among all those models, SVM with linear kernels outperformed others, and we will use it in our static recognition. The evaluation merit of frames per second (FPS) is adopted to test the performance in real-time scenarios. In the case of CPU Intel i5-9300HF, the average FPS of running MediaPipe's hand-detection of solution for 60 seconds is 20.7 FPS. While SVM is working, the value is 14.7 FPS.

3.3.2 Dynamic Recognition model

In this part, GRU with the structure mentioned above will be treated as a baseline and has a comparison with the structure we proposed with the Attention mechanism. Furthermore, discuss the FPS on real-time recognition. We will evaluate the results using

Figure 19: Model Comparison

performance matrices, including accuracy, precision, recall, and F_1 score, to evaluate our outcome quantitatively. Accuracy represents correctly predicted labels from the whole dataset, as shown in Equation 7.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

Precision measures the number of actual positives in all the positives predicted by the model, which is a good measurement when the cost of False Positive is high. Equation8 gives the mathematical formulation. Recall calculates the number of actual positives predicted correctly by our model, which is a good measurement when the cost of a False Negative is high. Equation9 gives the mathematical formulation. The F_1 score combines Precision and Recall and represents both properties. Equation10 gives the mathematical formulation.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

$$F_1 Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{10}$$

In the above equations, TP, TN, FP, and FN represent True Positive, True Negative, False Positive, and False Negative, respectively. Confusion Matrices are also proposed to understand the models' performance better. For the baseline GRU, the merits and the total number of parameters are as shown in Fig20:

However, after introducing the Attention mechanism, the total number of parameters has been reduced by 93.53% while the performance is even better, Fig21 shows the result:

Frames Per Second (FPS) is also adopted to test the model's performance in real-time scenarios. In the case of CPU Intel i5-9300HF, the average FPS running MediaPipe's pose-detection of solution for 60 seconds is 14.06 FPS. While the simple GRU is working, the value is 8.1 FPS, but GRU with the Attention mechanism's value has increased to 9.7 FPS, which is a noticeable improvement.

Merits	Output Shape
Accuracy	99.07%
Precision	99.24%
Recall	99.24%
F1 Score	98.72%
Parameters	5,7942

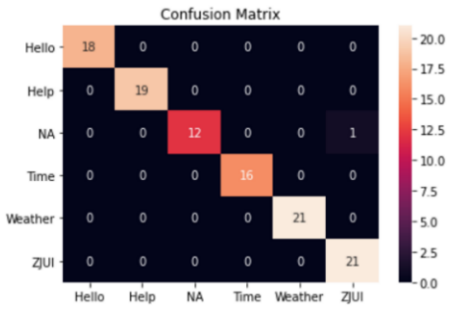


Figure 20: GRU's Merits and Confusion Matrix

Merits	Output Shape
Accuracy	100.00%
Precision	100.00%
Recall	100.00%
F1 Score	100.00%
Parameters	3750

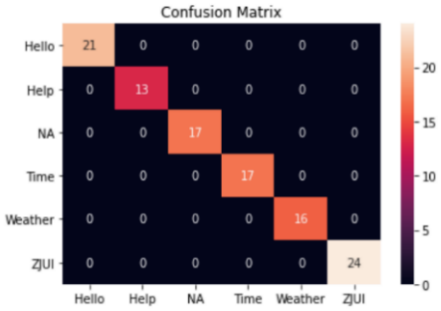


Figure 21: GRU + Attention's Merits and Confusion Matrix.png

4 Conclusion

4.1 Accomplishments

The final bionic hand subsystem has reached aesthetic and functionality expectations. The bionic hand subsystem can perform 17 DOFs fluently and non-interferingly. Combining 17 DOFs can easily express all the letters and numbers in ASL. The bionic hand achieved all the listed requirements and performed better than the design for Requirements 2 and 3. Also, taking advantage of good controlling code programming, the expression and indication of each letter is clear and distinguishable. The reliability and durability of the bionic hand are also satisfied, making our demonstration a big success. For the control subsystem, the control program for outputting 27 separate sets of signals, standing for a whole alphabet and a default position, is developed. Besides, serial communication between the upper computer and the development board is achieved. The reliability and efficient data transmission is guaranteed.

An ASL recognition subsystem in real-time scenarios is accomplished for the gesture recognition subsystem. Also, we collected the dataset and put forward two different features considering our small amount of data. Moreover, after comparison and analysis, we selected the static machine-learning model and improved the baseline model of GRU by introducing the Attention Mechanism, which significantly improves performance.

4.2 Uncertainties

Due to the limited iteration and tuning time at the final stage, we did not have a chance to have a better design at the axle part of the four-bar linkage and used a half-screw and normal nut. The ill-considered design costs more effort in maintenance and examination. It could be improved by changing the nut to a clamp nut or using a bearing.

Though introducing the Attention Mechanism improves the model's performance, the specific weights allocated to each input vector is still vague to us.

4.3 Future Work

Our intelligent assistant has the potential for further development into a comprehensive smart home system, leveraging the existing hardware on the development board. We can achieve a practical example by integrating an HC05 Bluetooth module into our board, using wireless communication to send instructions and control the lightning devices.

In the gesture recognition subsystem, we only deploy our system into personal computers and Jetson Nano. In the future, we would like to develop an App on smartphones or integrate it into the Web as an online accessible tool. Also, the proposed model could be better, and we will dedicate ourselves to consistently improving it for better performance.

5 Cost & Schedule

5.1 Cost

As engineers, all teammates' labor is valued and should be evaluated according to the total time to complete the project. Our group decides to work at least 15 hours per week per person on designing, testing, and validating our project and at least 3 hours per week to document our work. The graduate Research Assistant at the University of Illinois will get paid 40$ per working hour. Therefore the total labor cost of our team will be:

$$4 \cdot \frac{\$40}{hr} \cdot \frac{18hr}{week} \cdot 10week \cdot 2.5 = \$72,000$$

At the same time, the cost of the prototype is estimated at $235.3 in total:

Table 2: Prototype Cost Estimation

Part	Cost per piece	Piece	Total
Electrical Platform (2 DOF)	$20	2	$40
Bread Board	$2.5	1	$2.5
Servo Motor (SG90, 9 gram)	$1	20	$20
Steal Rod (ϕ 2.8mm * 1m)	$5	1	$5
M3/M4/M5 Bots & Nuts	$2.5	2	$5
3-mm Bearings	$0.4	2	$0.8
Jatson Nano Development Board (B1)	$150	1	$150
Camera	$12	1	$12
Total		30	$235.3

Therefore, our total cost will be estimated at 72,235.3$.

5.2 Schedule

week	Hanwen Liu	Yike Zhou	Haina Lou	Qianzhong Chen
3/20	Set group rules, distribute work, and read embeded system development manual	Extract Features of an open-sourced dataset by MediaPipe. Train and evaluate several machine learning models.	Buy hardware materials and learn embedded RTOS development process	Iterate and refine the mechanical design. Manufacture the Version#0 finger and combine with servo motor for simple demonstration
3/27	Construct our static sign language dataset and find effective features	Complete the static recognition model and test it on a real-time scenario	Implement a small demo: programming to output PWM signals and drive one servo motor	Confirm the mechanical design, pace up to manufacture the bionic hand
4/3	Deploy pre-trained model on the device to test the performance	Build and train LSTM based on open-sourced dataset	Output different signals to schedule all 24 motors	Continue to manufacture the bionic hand and purchase the electrical platform holding the hand
4/10	construct our dynamic sign language dataset	Build and train GRU based on open-sourced dataset	Test the motors which are installed in bionic hand to realize different hand gestures	Work with Haina to tune the motor and control code, turning the simple motion of fingers to gesture

week	Hanwen Liu	Yike Zhou	Haina Lou	Qianzhong Chen
4/17	Work on embedded programming to make dialog display on LCD	Adjust the model's structure and parameters on our dataset	Connect LCD to STM32 development board and continue work on other different hand gestures	Continue to work on gestures, manufacture the electronic components box, get ready for system test
4/24	Work with Haina to connect Jetson Nano board to STM32, and perform different hand gesture based on prediction	Explore more models that may yield good outcomes	Help Yike deploy model on Jetson Nano board. Work with Howie to connect Jetson Nano board to STM32	Tune and refine the details of system's mechanical parts, start preparing final demo
5/1	Combine and test all the subsystems. All together			
5/8	Prepare mock demo and the final report draft. All together			
5/15	Prepare functionality demonstration video and the final report. All together			

6 Ethics & Safety

When designing a product, safety is our top priority. To ensure "the safety, health, and welfare of the public"[15], it is vital to notify users of potential hazards and minimize the possibility of systematic danger caused by misuse of our work. This means guaranteeing that users are aware of any potential risks associated with using our intelligent assistant and are given clear instructions, warnings, and possible solutions when danger happens. In addition, it is essential to implement safety features and safeguards that can help prevent accidents and minimize the consequences and risks of any incidents that could happen.

As engineers, we are responsible for addressing unforeseen risks to ensure the safety of users. Regarding the movement of the fingers controlled by pulling on strings, we must implement safety features to prevent any harm caused by misuse. This could include providing clear instructions on how to use the product and warnings and possible solutions when danger happens. Additionally, we could design the strings to minimize the possibility of jerking or twisting, such as using more vital strings or providing guides for proper string movement. We could include safety covers or guards around the strings to avoid tangling or wrapping around the user's neck or other body parts.

For the sharp or protruding parts of the product, we must mitigate the risks by adding protective covers, smoothing the edges, or even redesigning the product. We could also provide clear instructions on handling the product to prevent harm.

In case of a system malfunction or short circuit, we could incorporate safety features to minimize the consequences and risks of any incidents that could happen. We should use materials less likely to cause harm from accidents, such as fire-resistant materials. Additionally, we could design the product with an emergency shut-off feature that could quickly disconnect the power source in case of a malfunction.

The Occupational Safety and Health Administration (OSHA) [16] provides guidelines for ensuring workplace safety, while the Federal Communications Commission (FCC) [17] sets standards for electronic devices' safety.

The soul of design is to help people's life easier. As a society, striving for respect, inclusivity, fairness, and equilibrium is vital, ensuring everyone has access to the tools and resources they need to live fulfilling lives without "discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression"[15]. Our core is to help people with hearing and speech problems could also interact with intelligent assistants.

References

[1] C. Lugaresi, J. Tang, H. Nash, *et al.* "Mediapipe: A framework for perceiving and processing reality." (2019), [Online]. Available: https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf.

[2] H. Basumatary and S. M. Hazarika, "State of the art in bionic hands," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 2, pp. 116–130, 2020. DOI: 10.1109/THMS.2020.2970740.

[3] S. J. Bensmaia, D. J. Tyler, and S. Micera, "Restoration of sensory information via bionic hands," *Nature Biomedical Engineering*, pp. 1–13, 2020.

[4] J. A. George, D. T. Kluger, T. S. Davis, *et al.*, "Biomimetic sensory feedback through peripheral nerve stimulation improves dexterous use of a bionic hand," *Science Robotics*, vol. 4, no. 32, eaax2352, 2019.

[5] C. Chen, J. Sun, L. Wang, *et al.*, "Pneumatic bionic hand with rigid-flexible coupling structure," *Materials*, vol. 15, no. 4, p. 1358, 2022.

[6] "American sign language." (2023), [Online]. Available: https://en.wikipedia.org/wiki/American_Sign_Language#/media/File:Asl_alphabet_gallaudet.svg (visited on 05/13/2023).

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. "You only look once: Unified, real-time object detection." (Jun. 2016).

[8] Google. "Hands." (2021), [Online]. Available: https://google.github.io/mediapipe/solutions/hands.html (visited on 03/08/2023).

[9] "Jetson nano developer kit." (), [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (visited on 05/23/2023).

[10] N. Developer. "Jetson Nano Developer Kit." (2021), [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (visited on 03/08/2023).

[11] A. Halder and A. Tayade, "Real-time vernacular sign language recognition using mediapipe and machine learning," *Journal homepage: www. ijrpr. com ISSN*, vol. 2582, p. 7421, 2021.

[12] G. H. Samaan, A. R. Wadie, A. K. Attia, *et al.*, "Mediapipe's landmarks with rnn for dynamic sign language recognition," *Electronics*, vol. 11, no. 19, p. 3228, 2022.

[13] Matthew-West. "UIUC TAM 212 Course Website." (2021), [Online]. Available: http://dynref.engr.illinois.edu/aml.html (visited on 03/23/2023).

[14] AKASH. "ASL Alphabet." (2018), [Online]. Available: https://www.kaggle.com/datasets/grassknoted/asl-alphabet (visited on 03/24/2023).

[15] IEEE. "IEEE Code of Ethics." (2020), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/08/2023).

[16] "Law and regulations." (2023), [Online]. Available: https://www.osha.gov/laws-regs (visited on 03/24/2023).

[17] "Federal communications commission." (2023), [Online]. Available: https://www.fcc.gov (visited on 03/24/2023).

Appendix A

Table 3: Requirements and Verification for Bionic Hand Subsystem

Requirement	Verification	Status (Y or N)
1. Every finger has 3 DOFs	A. Using CAD to assembly and validate the design. B. Print the finger and check the smoothness of motion.	Y
2. Finger can bend for more than 60 degree	A. Using "motion study" function on CAD software to have a throughout dynamic simulation.	Y
3. Motion of finger is finished within 600 ms	A. Connect the servo motor, linkage, and finger to test and record the motion time.	Y
4. The palm can steadily fix five fingers and avoid vibration	A. Using FEA software to have statics simulation and analysis on palm part under various situations. B. Test the strength of palm part before assembly.	Y
5. Motion of each finger would not cause intervention with others	A. Conduct dynamics simulation to validate the design.	Y

Table 4: Requirements and Verification for Gesture Recognition Subsystem

Requirement	Verification
1. The accuracy of the recognition should be above 90%	A. Divide the dataset into the training dataset (80%) and testing dataset (20%). B. Taking the strategy of cross-validation to train and evaluate the model. C. Input real-time video captured by the camera to check the performance of the trained model.
2. The model's whole response time should shorter than 500ms	A. Group members perform different sign movement and record the time from the end of the action to the display result formore than 20 times. Calculate to get the mean valueas the final response time.

Table 5: Requirements and Verification for Control Subsystem

Requirement	Verifications	Status
1. The delay from microcontroller receives decision message from Jetson Nano board to output corresponding PWM signal should be less than 1 second	A. set the LED light on development board to turn green when it receives message from PC/Jetson Nano B. record the time duration between the LED light turning on and hand movement	Y
2.Ability to perform ASL hand gestures a.Output 27 separate sets of signals, standing for a whole alphabet and a default position. b.Output 17 stable PWM signals simultaneously at 50 Hz frequency. c. Asynchronous PWM signals output for performing letters M, N, S, T	A.record every movement of bionic hand for each letter in alphabet B.show the recording to someone who knows ASL, and test if he can identify the letter in ASL	Y
3. The accuracy of data transmission between upper computer and development board should be 100%	A. in control program running on development board, echo back the received message to upper computer. B. check the message whether it is the same as message we transmit	Y

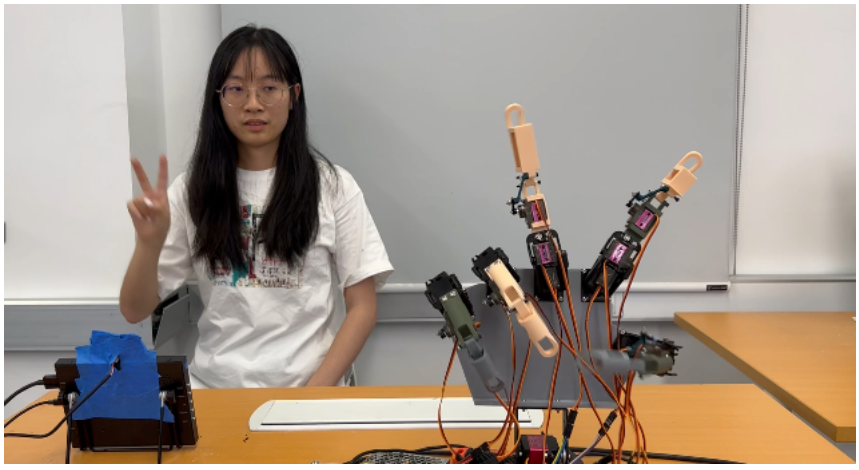


Figure 22: Verification for Control System