Final Reflection for Mechatronics

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

April 6, 2023

1 Revision History

Date	Version	Notes
April 05, 2023	1.0	Everyone - Initial Final Reflection Draft

2 Symbols, Abbreviations and Acronyms

See SRS Documentation at https://github.com/kelhuynh/OpenASL/blob/main/docs/SRS/SRS.pdf

Contents

3 Changes in Response to Feedback

3.1 SRS and Hazard Analysis

Our team has made significant progress in improving our SRS document since the initial draft. Our project aims to develop a hand gesture translator for ASL, and we recognized the need to make changes to our SRS as our project evolved with the addition of new modules and the removal of some requirements that were not ideal for our system. Based on feedback from our TA, we added a new figure that outlines how each module and signal interacts with each other and what input and output each one requires. This helped us to better understand how the Machine Learning Module fits into our design and subdivide it into a smaller part, allowing us to create more specific requirements for the Gesture Detection Module that we realized we needed. This helped for testing as we could perform better testing and quality assurance for the VnV Report, for example, to ensure that the software meets the specified requirements and functions as expected. Additionally, we recognized the need to consider the user interface (UI) more thoroughly and included more assumptions as we discovered limitations within our project. For example, we assumed that users would only sign the available gestures within the system, as we realized the data requirements for machine learning to recognize dynamic movement gestures compared to static gestures that do not move and could not include as many gestures as we originally hoped for. We also added a traceability matrix to show how our different requirements depend on each other. Lastly, we included a section that delves more deeply into how ASL works in terms of grammar and linguistics, which will greatly improve the understandability of our project. These improvements have allowed us to develop a more comprehensive and accurate SRS document that better reflects the requirements of our project.

Relative to the feedback that was received from the TA and peers through Avenue and Github respectively. The hazard analysis has been improved to include risk mitigation for electric shock which was deemed necessary due to the involvement of electrical components with the system. It has also been reworked such that the recommended actions do not involve the end-user but are directed as what the system should do to indicate that the error has occurred or how we as designers should mitigate the risk. Furthermore, the roadmap also specifically indicates the risks and errors that might not be considered during the course of the project. These changes have helped with the development of the translator as it enabled us to determine which risks can be completely mitigated and which ones can be partially mitigated given the time constraints of the project. This resulted in much less wasted time pursuing fruitless endeavors allowing us to focus on bettering specific aspects of the translator.

3.2 Design and Design Documentation

MG (meng):

System Design (meng):

We made significant updates to the detailed design documentation based on the feedback provided by the TA. Our focus was on providing more comprehensive information on how to build the system. To achieve this, we added more modules that better reflect our updated program, such as modules for the UI and newer machine learning modules that were developed to better support dynamic gestures. In addition, we removed redundant modules that had no input and output as they were part of a larger module and not their own individual ones. These changes should make it easier for an independent developer to implement our modules.

3.3 VnV Plan and Report

In response to feedback from both the TAs and our peers, we modified various sections of our plan. One change we made to our plan was modifying several test cases. We changed the test cases to better represent the system, mostly through the initial state. Originally, many of our initial states described the users and their actions, when they should have described our system. We modified them to make it clearer as to what our system is doing at the beginning of these tests, and how these tests may be performed, by talking about the initial state of the program and the camera. We also updated our traceability matrix. As we have updated our requirements in the SRS document, we needed to remap which tests relate to which requirements. With these new requirements, we are able to better explain how the system can be tested and verify that we have achieved the requirements we would like.

In response to feedback and as part of our ongoing efforts to improve our project, our team has made significant advancements to the Verification and Validation (VnV) report document since the initial draft. We updated our tables to include the left and right bars and subdivided our testing into more specific sections so that it is more clear what it is we're testing. Due to changes in the SRS for our functional requirements we also added more tests to help go over more specific issues such as UI testing and data point collection. We also fixed our issue for the non-functional quality as we accidentally wrote the same non-functional quality twice. The accuracy quality now talks about how in the graph, by adding more data points for each word, the machine learning model is able to improve its accuracy as there is a greater database pool for it to draw information from and be able to differentiate the many gestures we have.These improvements have allowed us to create a more comprehensive and accurate VnV report that better reflects the requirements and objectives of our project.

4 Design Iteration (L011)

Over the course of the project, we went over many iterations of the final product. We started off the project by getting familiar with many concepts such as gesture recognition and machine learning since this was the first time many of us have tried using these concepts. This took a good period of time as the concepts were complex since we also needed to know how to combine gesture recognition and machine learning to build a classification system. The first item to be done at this point was simply getting hand joints recognized by the software so that we can create freeform data and shape it as we wanted for training a machine learning model.

Our first iteration of the translator was built primarily around static gesture recognition using Keras and Tensorflow for the construction of a sequential model using an input layer, one dropout layer, and 3 dense layers. This model started being trained with primarily static sign language gestures to which we achieved an accuracy of 98% in classification. As we worked towards motion gesture recognition, this accuracy value deteriorated immensely which made us realize that there was an issue with our implementation as well as dataset collection. Furthermore, around this point in time we learned that ASL was not solely hand gestures and involved a lot more such as facial expression and body language.

Factoring in these two issues, we came upon our second iteration of the translator which involved the use of a different model built also using Keras and Tensorflow but instead with three LSTM layers, two dropout layers, and three dense layers. This was because we realized that we need Long Short-term Memory (LSTM) to allow the model to classify gestures using prior input. We also expanded on how we collected data to train the model by making it so that we would take 30 input frames worth of data from video capture overlaid with joint coordinate data for the face, body, and hands to account for what we learned. This enabled us to build a much more accurate model when it came to motion gestures compared to the first iteration; however, it was lacking in accuracy when attempting to detect static sign language gestures.

With the second iteration's failure to meet our requirement of accuracy, we worked on a third iteration which was a combination of the best parts of its predecessors. We decided to combine and use the two models individually where we would use the first iteration's machine learning model solely for static gesture detection and the second iteration's model solely for motion gesture classification. This prevented either model from having a massive drop in accuracy since they weren't trained with the data that was causing issues in the first place.

5 Design Decisions (L012)

When we started this project, we first wanted to figure out if we could even use a camera to track hands. We would also need a machine learning model to determine what gestures are being performed. With this in mind, we did some research and found the media pipe library and tensorflow library for Python. These would allow us to use computer vision to track hand joints and a machine learning model to understand the gestures. For our first implementation, we had our system track the users hand joints in each frame, effectively tracking the static images in a video capture. We then used the hand joint coordinates from each frame as input to train our machine learning model to detect what hand gesture is being performed. This was a good starting point for us as it allowed us to prove that detecting hand signs using computer vision and machine learning was possible. However, we knew that most ASL gestures required motion, so our next step was to find a way to detect motion gestures. So for our next implementation, we tried to implement a new system that would track the hand joints through motion, but this did meet our accuracy requirement, as the model would have a difficult time guessing the correct gesture. Therefore, we tried a different method for tracking motion. Instead of using static hand joint coordinates, we would use multiple frames of data as the input data. This allowed our model to better recognize the motion gesture signs, but had a difficult time recognizing static hand gestures that had no movement. Since ASL consists of both static and motion hand gestures, we decided to combine both methods we used into one, which led to our final implementation. We decided to have different modes for both static and motion gestures, which gave us the ability to track both types of gestures.

6 Economic Considerations (L023)

By evaluating our product and the current market for ASL translation products, there is a market to launch the ASL translator. In order to bring our product to the market and deliver to potential customers, a proper product marketing process is essential. First of all, the ASL translator is positioned as high quality and highly accessible with accurate results.The target customers for our product are people who have hearing problems or require to use hand gesture translation to support their daily communication, people who cannot afford an in-person sign language translator and beginners who want to learn sign language on their own. And our initial target market is the Hamilton region and cities around Hamilton. Advertising and promotion are important methods to promote our product to attract more customers. For example, one of the promotion ways can be collaborating with the accessibility services in schools. The ASL translator is composed on a Raspberry Pi, a camera and the associated program to translate the hand gestures into the corresponding text and speech. The estimated cost for one unit is $400, and the price will vary according to the functionality and version of the Raspberry Pi and camera. By considering the one-time development cost of $1500 and rolling production costs of $75 per unit, the finalized product will be changed to $500 per unit ($25 profit margin). This means that at least 60 units need to be sold

before making profit. According to the research in 2022, approximately 357000 culturally deaf Canadians and 3.21 million hard of hearing people live in Canada. Therefore, we could approximate our initial potential users to be 5000 based on our target market areas and target customers.

7 Reflection on Project Management (L024)

7.1 How Does Your Project Management Compare to Your Development Plan

The team is able to follow the Development Plan we had created at the beginning of the academic year well and has a good workflow. The meetings were scheduled weekly and we decided to have a "flexible" meeting time, which means that different meeting time and length would be set depending on the workload and team members' availability. And the meetings were either conducted on Microsoft Teams and in person based on team members' agreement and flexibility. The communication plan was also well followed, team communication was managed and conducted using Microsoft Teams and Facebook Messenger. Communication related to the source codes was conveyed through the use of Github issues and pull requests. And all the team members performed their assigned roles well. The overall workflow for the team was managed by using Gitlab and Github, every team member worked on the assigned tasks first and then committed their content onto the documents. At the end of the project period, the team could look up the opened issue, solve them and update the documents. During the project, we had applied all the technologies planned in the Development Plan, and they are important tools that help us do the work efficiently with high quality.

7.2 What Went Well

Overall, our project has laid a solid foundation for a machine learning module and algorithm that can recognize American Sign Language (ASL) gestures. By providing a diverse dataset to train the model, our program has demonstrated the ability to categorize any ASL word or gesture, including those that involve grammar and linguistics using facial expressions. With the integration of MediaPipe, our program enables ASL users to fully express themselves and translate their language to non-ASL users. Notably, our algorithm is not limited to ASL and can be extended to other sign languages globally with a diverse dataset. As we continue to expand our project to include new gestures, we have the potential to contribute to a more inclusive society that values and respects the deaf community.

In terms of our project management, we were able to stay on top of our work and make good progress on our project. While we did not meet 3 times a week as we said we would in our development plan, we still met regularly at least once a week, and often more when there were major deadlines. During these meetings, we would discuss what needed to be done and

figure out what we want to add or change with our project. This allowed us to separate the work to get it done in a timely manner.

7.3 What Went Wrong

Our project encountered two significant issues that prevented us from achieving our ideal system. The first issue involved our original algorithm for training the machine learning model, which was unable to accurately determine dynamic gestures that required hand movement. Our previous method involved recording data points of hand joints by clicking a key, which was insufficient for gestures with changing hand orientation and posture. After consulting with ASL users and conducting further research, we developed a new method that recorded frames of the user performing a dynamic gesture. While this approach was successful for dynamic gestures, it rendered the recognition of static images difficult. Unfortunately, we could not integrate these two methods into a single program, requiring the user to manually select the appropriate model. As a result, our system was unable to capture the full language of ASL.

Our second issue was related to hardware limitations when using a Raspberry Pi for portability. The processing power of the Raspberry Pi was insufficient to handle large amounts of image data, resulting in a slower frame rate that impaired the machine learning model's ability to accurately predict hand gestures. Despite our attempts to use multi-threading and overclocking to improve performance, we were unable to achieve the desired frame rate for our final product.

7.4 What would you Do Differently Next Time

One area where we could have improved our approach is in the development of our machine learning algorithm. Specifically, we could have leveraged existing hand gesture translation datasets provided by Google, which would have saved us a significant amount of time and resources. Instead, we spent countless hours manually recording millions of key points for each gesture, only to discover that we needed even more data to accurately categorize similar hand positions. This process was time-consuming and often resulted in trial and error training, which can be unreliable given the unpredictable nature of neural networks and their tendency towards non-linear relationships and overfitting. By using an existing dataset, we could have focused on refining and optimizing our model parameters to improve accuracy, rather than spending valuable time on data collection.

One thing we could have done differently is make better use of Git and the various version control tools it has. None of us have ever had to use or learn about Git before, but after having been exposed to it, we realize how powerful it can be when working on software collaboratively. We often struggled combining different implementations of code, especially

when it came to the tex files for the documents. We had to deal with several merge issues as things would sometimes get overwritten. Using the knowledge we have now, if we had made use of branching and merge requests during development, we could have had an easier time combining our work.