American Sign Language Translator for Inclusive Virtual Meetings

Master of Engineering Capstone Report
University of California, Berkeley

Irina Hallinan [EECS], Isadora Smith [BIOE], Ashley Zhang [EECS]
Advisor: Prof. Brian Barsky [EECS]

May 2023

Abstract

The rise of popularity in virtual meetings have left Deaf and hard of hearing people out of the conversation. An existing solution to join online meetings is to hire an in-person translator; however, translators require time and money. We developed an alternative no-cost solution. We conducted user research and gathered feedback from over 120 American Sign Language (ASL) users. Based on our findings, we designed and built the Sign Language Assistant for Meetings (SLAM) application. The application is used as a real-time, ASL to English text translator. Users can download SLAM as an extension to their Zoom meetings. Users sign ASL into the built-in computer camera and that information is sent to the backend. The backend utilizes the Sign pose-based transformer for word-level sign language recognition (SPOTER) machine learning model [1] to detect key points on the user's hands, face, and body. We trained the model to recognize over 1000 signs and output the corresponding English translation with the highest statistical probability. The final translation is displayed as captions for the other members of the virtual meeting, integrated into the Zoom client. For future work, SLAM can be expanded to translate spoken English to ASL. One possible idea is to include a virtual avatar in the virtual meeting which will sign ASL to the Deaf and hard of hearing user as English is being spoken in real-time. This will complete the two-way communication that is typically done by an in-person ASL interpreter.

Contents

Chapter 1

Discovering the Needs of American Sign Language (ASL) Speakers and Overview of Current ASL Recognition Technologies

1 USA Deaf Community and Their Needs

According to the World Health Organization, the hearing threshold for normal hearing is within the 20 dB range in both ears [2]. People who are hard of hearing are outside of that 20 dB range. They can still hear, but it comes with difficulty and they typically need assistive devices such as live captioning or hearing aids. People who are deaf cannot hear at all or have a very small threshold for hearing. They rely on sign language to communicate and the US has a specific language called American Sign Language (ASL). It is a language that utilizes signs created by the hands, face, and body.

In the US, there are approximately 37 million deaf and hard of hearing people [3], which accounts for 11 percent of the population in the 2020's. There are approximately 1 million ASL users [4], which is not mutually exclusive with the hearing population. Many deaf people have family and friends who learn ASL to communicate with their loved ones. This has led ASL to be in the top 10 most commonly used languages in the US [5]. A vibrant deaf community in schools, work places, and society at large creates an inclusive environment that is beneficial to all.

People who are Deaf or hard of hearing have several obstacles in their day to day life, such as inclusion in schools or workplaces. Over the pandemic, virtual meeting platforms like Zoom and Google Meets rose in popularity due to their convenience and ability to keep schools and workplaces running. However, these platforms are not designed to be inclusive to the deaf or hard of hearing community. The reliance on spoken mode of communication makes it difficult for Deaf people to join online meetings without an in-person translator. Hiring an ASL interpreter costs time and money. Therefore, the Deaf and hard of hearing community is left out of these meetings, which further perpetuates the obstacles they endure.

2 Overview of ASL Recognition Machine Learning Models

In ASL, we use 5 parameters to describe how a sign behaves: 1) hand-shape, 2) location, 3) movement, 4) palm orientation, and 5) facial expression or non-manual signals (Figure 1.1). As a result, an accurate recognition model needs to take all 5 parameters into consideration. In this section, we discuss works related to American Sign Language recognition. We divide the works into two types: static gesture recognition (SGR), and real-time action recognition (RTAR). Table 1.1 summarizes these models.

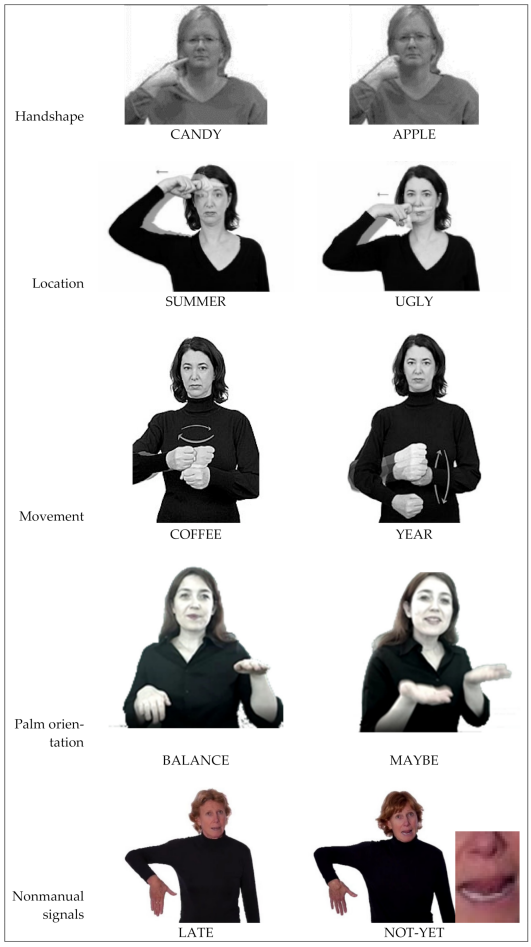

Figure 1.1: The five parameters of ASL. Image credit: [6], Figure 8.

2.1 Static Gesture Recognition for a Single Image

The first category of ASL recognition work can recognize the user's static gestures. Specifically, this kind of tool is used for recognition of static hand gestures. Saha *et al.* [7] proposed a novel image processing sign language detection framework that employs the MAdaline network for classification purposes. The main focus of this framework is on two aspects. This model introduced an advanced feature set with seven distinct features. In addition, the model eliminated the process of cropping out unnecessary background images, which reduced system complexity. This framework

Model	Year	Type	Remarks
MAdaline	2016	SGR	It can recognize 26 English alphabet letters. Accuracy is more than 93%.
H3DF	2013	SGR	It can encode the 3D shape information from depth maps. It can recognize digits from 0 to 9 and letters from a to z without j and z. Accuracy is more than 91%.
I3D	2017	RTAR	Reaching 80.9% accuracy on HMDB-51 dataset and 98.0% accuracy on UCF-101 dataset.
SPOTER	2022	RTAR	Outperform the prior state of the art with a relative improvement of about 4%.

Table 1.1: Summary of ASL recognition machine learning models

has been implemented to recognize the 26 English alphabet letters from 'A' to 'Z' of standardized ASL and the results had an accuracy of more than 93%.

In addition to the MAdaline Neural Network, Zhang *et al.* [8] used the Histogram of 3D Facets (H3DF) to explicitly encode the 3D shape information from depth maps. This H3DF descriptor, which is based on the depth map, offered two advantages over previous 2D image descriptors. In contrast to previous 2D global descriptors, such as the Histogram of Oriented Gradients (HOG), it applied a compact global representation to characterize a depth image. Two types of experiments have been conducted to test the H3DF descriptor. Two public data sets, the NTU Hand Digits data set and the ASL Finger Spelling data set, containing depth maps of hand gestures. The results showed that the H3DF descriptor can recognize the hand gesture with a mean accuracy of more than 91%.

2.2 Real-Time Action Recognition for Video Inputs

In comparison to detecting static gestures, recognizing movements with a temporal component is more challenging. In this part, we look into the research that aims to recognize real-time actions and translate them into spoken English or text.

Carreira *et al.* [9] proposed a novel Two-Stream Inflated 3D ConvNet (I3D) that is based on 2D ConvNet inflation. This model allowed for the learning of spatial-temporal feature extractors from video while utilizing successful ImageNet architectural designs, including their parameters. The authors demonstrated that I3D models significantly outperformed the state-of-the-art in action categorization after pre-training on kinetics, reaching 80.9% accuracy on HMDB-51 data set and 98.0% accuracy on UCF-101 data set. Li *et al.* [10] fine-tuned the I3D model based on ImageNet and Kinetics and tested it using the Word-Level American Sign Language (WLASL) video data set, which contains 2000 words performed by over 100 signers. The results showed the tuned I3D model achieved better performance compared to the other image-appearance based models.

Bohacek *et al.* [1] subsequently presented the SPOTER system, which utilized a pose-based transformer model for word-level ASL recognition. The recognition was based on the estimation of the human body's pose in the form of 2D landmark locations. The authors presented a robust pose normalization scheme that consid-

ered the signing space and processed hand poses independently of the body location. Additionally, they introduced several body pose augmentations that improved the accuracy even further, including a novel sequential joint rotation augmentation. They validated their system on two data sets, WLASL100 and LSA64. The results showed that the SPOTER model outperformed the prior state-of-the-art models with a relative improvement of about 4%.

3 Current ASL Technologies and Novelty of SLAM

To enable us to design a novel application, we conducted an analysis of the current assistive technologies available to ASL users. Based on these findings, we designed our app to target the biggest pitfalls of the current technologies.

The most prominent use cases for SLAM include online meetings for school, work, job interviews, and connecting with friends and family. Since it is an extension to the Zoom marketplace, SLAM is only compatible with that platform. However, for future work, it can be expanded to any platform such as Google Meets, Microsoft Teams, or Skype. SLAM can also be used on mobile devices. This enables Deaf or hard of hearing users to communicate with people who don't know ASL.

Current ASL apps on the market include Hand Talk and ASL Translator. Both apps translate English text to ASL and use an avatar to sign ASL to the user. They are stand-alone assistive technologies and don't integrate into a larger platform. These translator apps don't address the other side of the communication, which is translating ASL to English text. Jeenie is another app that enables users to hire an in-person translator to join their online meetings. However, as previously addressed, this requires time and money. Additionally, if the meeting is personal or has sensitive information, an in-person interpreter may not be desirable.

There are many live captioning services and apps that capture voice and translate sound into written English. Zoom and Google have these technologies built into their platforms. However, these meeting software platforms don't take ASL into account. Therefore, this diminishes the effectiveness of these live captioning services for ASL speakers because the captions are in English rather than in ASL. There are currently no technologies that automatically translate ASL to English text that are compatible with online meeting platforms. After market analysis, we found that there are many pitfalls to the current technologies. SLAM would be a novel solution that aims to address some of these pitfalls.

4 Human-centered Design and ASL Community Findings

We followed the Human-Centered Design (HCD) techniques in our approach to build the prototype application that translates American Sign Language (ASL) into English. HCD is a set of principles centered on understanding the potential user prior to designing a solution. In the context of building a software application, HCD is known as User-Centered Design (UCD) [11]. The HCD principles are used in a variety of disciplines from designing software applications in healthcare [12], [13] to consumer products [14]. The universal approach of HCD is to design for the user with the user in mind, using a collection of social-behavioral methodologies. The

goal of HCD is to gain a deep understanding of the target user and to create a product that solves an existing need. We don't directly ask the users what kind of application they would want. Instead, we observe the users in their environment and come up with possible solutions to their existing challenges. The specific challenge we identified is the inability of people who rely on ASL to participate in virtual meetings with people who rely on spoken language. After establishing the challenge, we prototyped our solution as a web application. Then, we tested our solution with users and iterated the design based on user feedback.

The specific methods of Human-Computer Interaction we employed include contextual inquiry [15], rapid prototyping, iterative design, and user testing. In each step of the design process we focus on the target audience. Our initial target group is the Deaf, hard-of-hearing, and people who are learning ASL, between the ages of 20 and 30, who reside in the United States.

4.1 Observation of the ASL Community and its Influence on SLAM

The iterative steps of the Human-centered design approach are 1) observe, 2) design, 3) test, and 4) iterate, shown in Figure 1.2. In the first step, we observed the current habits of users by interviewing them and asking them open-ended questions about the way in which they communicate in ASL. In the second step, we described our target audience and designed the application. In the third step, we tested our prototype with users. Finally, we implemented the feedback we received into the next iteration of the design.

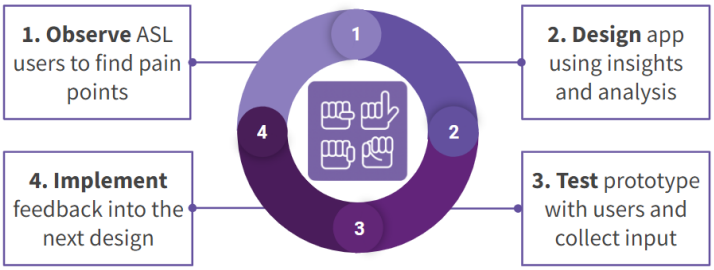

Figure 1.2: Human-centered design approach.

As part of the observation step in the HCD approach, we conducted interviews in order to better understand the target audience. We interviewed two people over the phone and email: one person working at the NorCal Services for the Deaf and one person working at the Deaf Community Services of San Diego. The interview questions focused on the current usage of technology as an aid in communication. Conversations over the phone lasted approximately 30 minutes. Sample questions included specific questions about current translation applications and general exploratory questions about the Deaf culture and community. An example interview question is: "Are you currently using a translator app?". The interview participants gave us a glimpse about how users currently communicate with hearing and non-hearing people and their existing challenges.

4.2 Online Surveys with Members of the ASL Berkeley Community

In addition to interviewing members of the ASL Berkeley community, we conducted two online surveys. We created a short online survey with 10 questions to find out more about user preferences of communications and their choices when it comes to ASL applications. We first sent the survey to the ASL student club at the University of California, Berkeley. We received 6 responses. Then, we iterated on the first survey and sent the second survey to the wider University of California community, including current students, staff, and alumni. We received 159 responses to the second survey, out of which 123 people responded that they know ASL in some capacity.

From the survey responses, we analyzed the quantitative and qualitative data to draw user insights. For example, we found that the majority of survey participants (around 88%) rate themselves as ASL beginners. Out of 10 questions, there were 7 quantitative questions with multiple-choice answers, and 3 questions were short-answer qualitative questions. The quantitative questions included questions such as: "In the past 90 days, how many times have you used American Sign Language (ASL)?". We found that less than a fifth of respondents use ASL on a regular basis (daily or weekly) and over 80% use it infrequently. Figure 1.3 shows a plot of the responses about frequency of ASL use in the past 90 days. Qualitative questions were of the form: "In the past 90 days, describe a situation in which you wished you had a technology that could help you communicate that doesn't exist yet.".

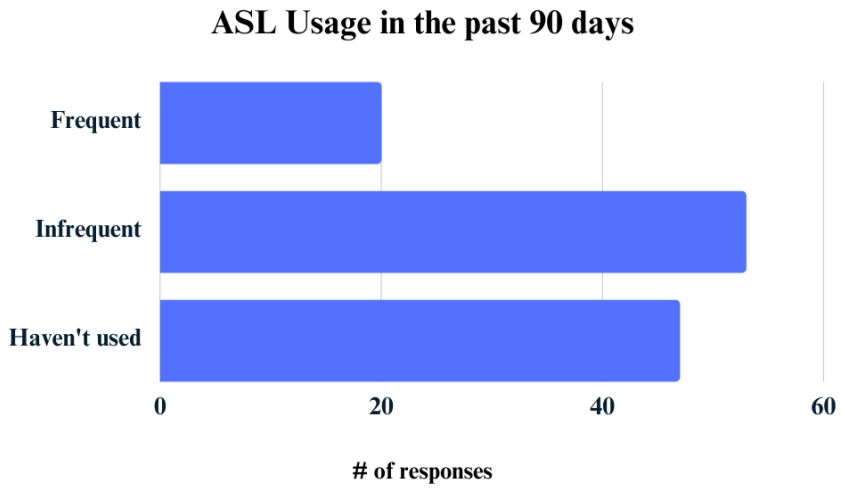

Figure 1.3: Comparing frequent and infrequent ASL use in the past 90 days from 120 survey responses

Based on the survey responses, most signers use ASL in face-to-face communications: around 65% use ASL in person and around 27% use ASL over video. A minority of survey participants currently use an application that relies on ASL (around 6%). Those who use an ASL-based app, use a dictionary app. Based on the responses from people who use an ASL-based app, we learned that users care about two parameters: the ease of use and speed. We asked if people would be interested in using an app that allows anyone to communicate in American Sign Language during

virtual meetings and the majority (around 84%) expressed interest. Therefore, we hypothesized that there's a need for a real-time ASL translator application and a similar app doesn't already exist on the market. Table 1.2 shows the summary of user survey results for quantitative questions. All questions were asked about ASL use in the past 90 days.

Question	Question topic	Result
1	ASL beginner level of knowledge	107 (88%)
2	Use ASL frequently	14 (17%)
3	Use ASL in face-to-face communications	61 (65%)
4	Have used an ASL app	4 (6%)
8	Interested in using ASL app for communications	97 (84%)
9	Interested in using ASL app for computer assistance	78 (70%)

Table 1.2: Summary of quantitative results (number of positive responses and percent) from the ASL user survey from 123 participants from the University of California student, staff, and alumni network

The qualitative survey responses revealed the main challenge of building an ASL-to-English translator: ASL is more than just signs. There are 5 main ASL components that play a crucial role in how the language works: gestures, body posture, facial expression, context, and repetition. All the components play a role and carry meaning. For instance, two identical gestures mean different things depending on context. Figure 1.4 shows two different signs made with the same gesture. Additionally, some signs vary only slightly in the hand orientation and can have different meanings, such as "dance" and "read" [10]. Moreover, ASL has dialects and slang, so sign meaning may differ depending on the person's location in the United States.

Figure 1.4: The ASL words "wish" (top) and "hungry" (bottom) are made with the same gesture. Image credit: [10], Figure 2 (a).

4.3 Creation of Persona and Customer Realization

After analyzing the user interviews and survey results, we developed a user persona based on the insights about our target audience. A user persona is an archetype of potential users [16], a tool used by user interface and product designers. We employed this tool to have a concrete representation of the target audience for whom we were designing. We combined the insights we drew from interviews and user surveys to explicitly describe our target user. This includes stating their motivations, worries, and personality characteristics. Developing a user persona is useful during the design and development phases. In order to decide whether to include a feature, we relied on the concrete user persona. If the feature is irrelevant to the target persona, it was not included in the application prototype. Appendix A contains the user persona document.

4.4 Storyboard for Detailed Application Scenario

In the final part of the design stage, we created a scenario in which the application would be used by the representative user persona. Based on our understanding of the target audience, we posited that the application we develop would be used in school, when a team works together on a project. Storyboards are used in interface design to create a narrative that gives application context. Originally a tool in the film industry, storyboarding is frequently used by application designers [17]. Storyboarding relies on simple sketches that show the main idea of the application use. We created the application storyboard that shows one potential use-case of the application. Figure 1.5 shows the application storyboard.

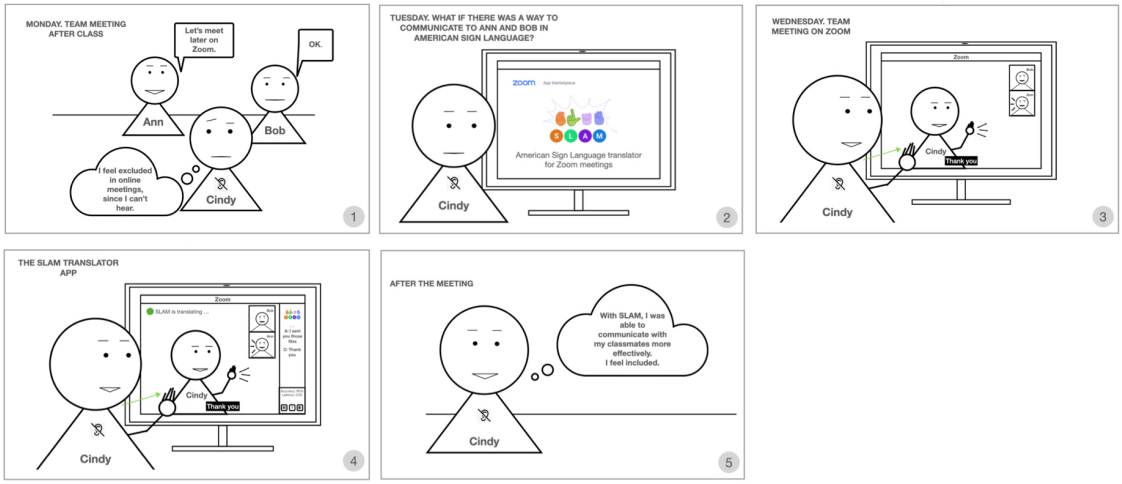

Figure 1.5: Application storyboard

Chapter 2

Designing SLAM: Sign Language Assistant for Meetings

5 SLAM Application Structure and Implementation of Machine Learning Models

5.1 SLAM Prototype as a Minimal Viable Product

The application we developed is called SLAM: Sign Language Assistant for Meetings. We chose a short and memorable name for the application that is descriptive of its function. The logo of the application spells SLAM using ASL finger spelling. Finger spelling is a way to represent distinct letters of the alphabet, based on the word written form. Finger spelling is used to spell proper names or words that have no official signs [18]. We chose purple as the main color of the application and its logo to be consistent with other popular communication applications such as Discord and Twitch. Figure 2.1 shows the SLAM application logo.

Figure 2.1: SLAM application logo and main color

The Minimal Viable Product (MVP) is a product with a small set of features that allows end-to-end testing with users [19]. The MVP we developed using the HCD approach is a web application, installed as an extension to the Zoom Video Communication client (Zoom) [20], [21]. The app prototype can be found in an online Zoom marketplace. We chose Zoom as the platform to deploy the prototype, due to Zoom popularity in the work and school settings [22]. Additionally, the Zoom Developer platform contains a robust set of Application Programming Interfaces (APIs), which is a standard way of developing a web application [23]. The Zoom Marketplace allows developers to deploy the application and test it with users, using their own machines.

5.2 SLAM App Infrastructure and Technology Stack

The SLAM application has three main components: the user interface (the frontend), the application logic (the backend), and the machine learning model. Figure 2.2 shows the main application components and a step-by-step application workflow that happens continuously as the application runs.

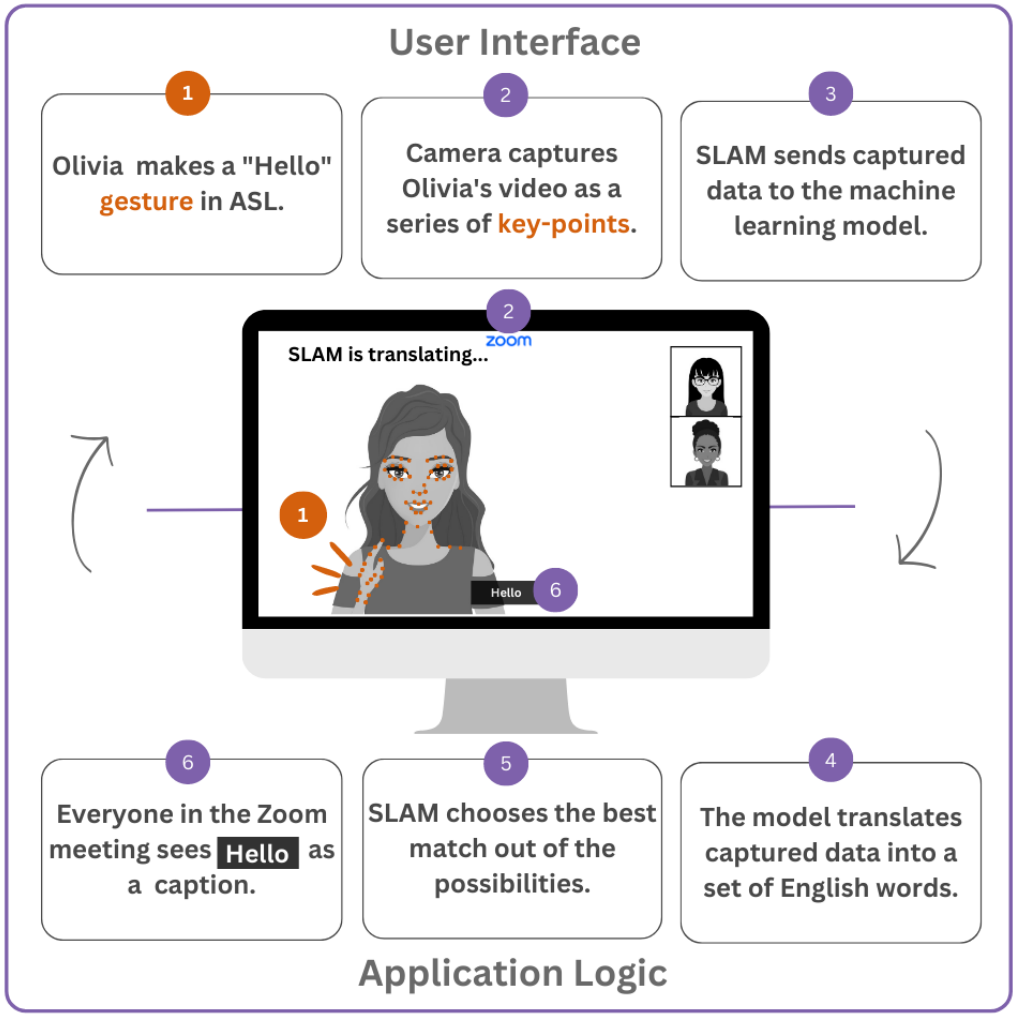

Figure 2.2: SLAM application workflow

We built the frontend with the JavaScript programming language using Node.js and Express.js frameworks. We used HTML and CSS with the Bootstrap framework to create a responsive user interface. Responsive user interfaces scale appropriately depending on the size of the screen. We chose responsive design because Zoom is used on both desktop and mobile devices with varying size of the screen.

The backend of the application is built using the Python programming language. The APIs transfer data between the frontend and the backend, using Representational State Transfer (REST) architectural style [24].

We used SPOTER [1] as our machine-learning model. The SPOTER model takes in a set of extracted key-points based on a collection of video frames and outputs a subset of English words that best correspond to the sign detected. The model is trained on the open-source data set WLASL2000, containing 2000 signs or glosses [10]. Each gloss is an English translation equivalent to one or two words.

5.3 SLAM User Interface Framework and Design

The SLAM user interface (frontend) is a component providing users with an intuitive and user-friendly interactions, as part of the Zoom meeting experience. The SLAM can be added as a Zoom application and the user can bring up the SLAM window from the Zoom menu bar when they join a meeting. The frontend's primary function is to call the backend of the application, when the user would like to start the translation of signs into text. The request to start translating is done via a GET API call to the SLAM backend. After the recognition process is complete in the backend, the frontend receives the translation data to display. ASL recognition results are displayed in the SLAM text box, shown in Figure 2.3. Users can edit the results if necessary, and when they are satisfied with the accuracy, they can press the send button to share the translation results with other meeting participants. The frontend also features a stop button, which users can use to terminate the recognition process. To help users navigate the app, there will be a help link located on the left top of the frontend that provides answers to Frequently Asked Questions.

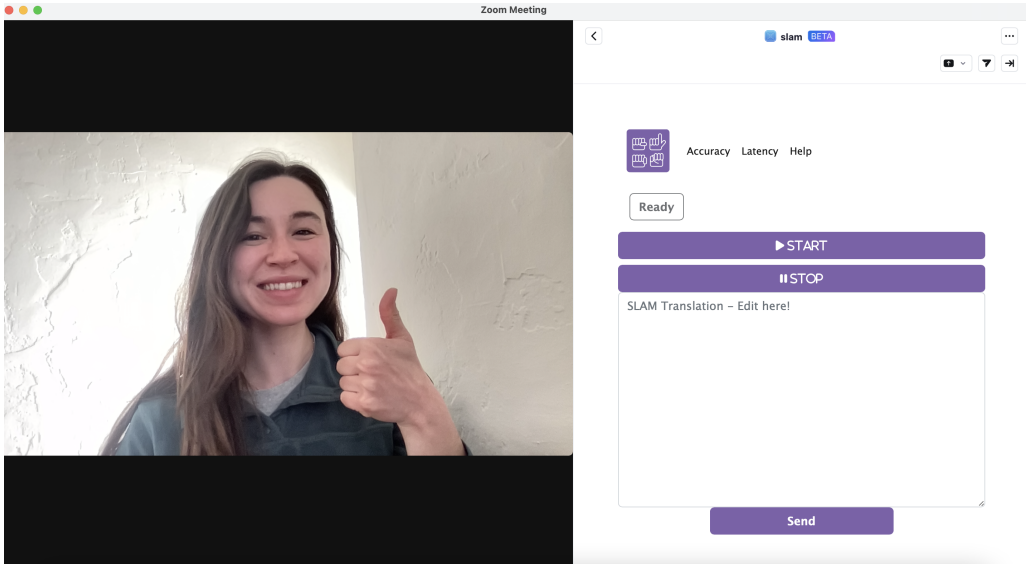

Figure 2.3: SLAM Application User Interface Design in the Zoom client

5.4 SLAM Backend and Application Programming Interface Design

The backend of the SLAM application exposes an Application Programming Interface (API) that gets called by the user interface, when the user presses the "start" button. When the backend receives the GET API call, the logic to capture video data and translate it begins. The backend uses the web camera to collect video input for a fixed amount of time, then passes it to the machine learning model, which returns top translations of a single sign. The length of the recording was based on the training set of over 20,000 videos, described in the next section. We chose 3 seconds as the fixed amount of time to record the video because over half of the training videos were between 2 and 3 seconds in length, with a mean of 2.4 seconds and a median of 2.3 seconds. Figure 2.4 shows the training length frequency distribution.

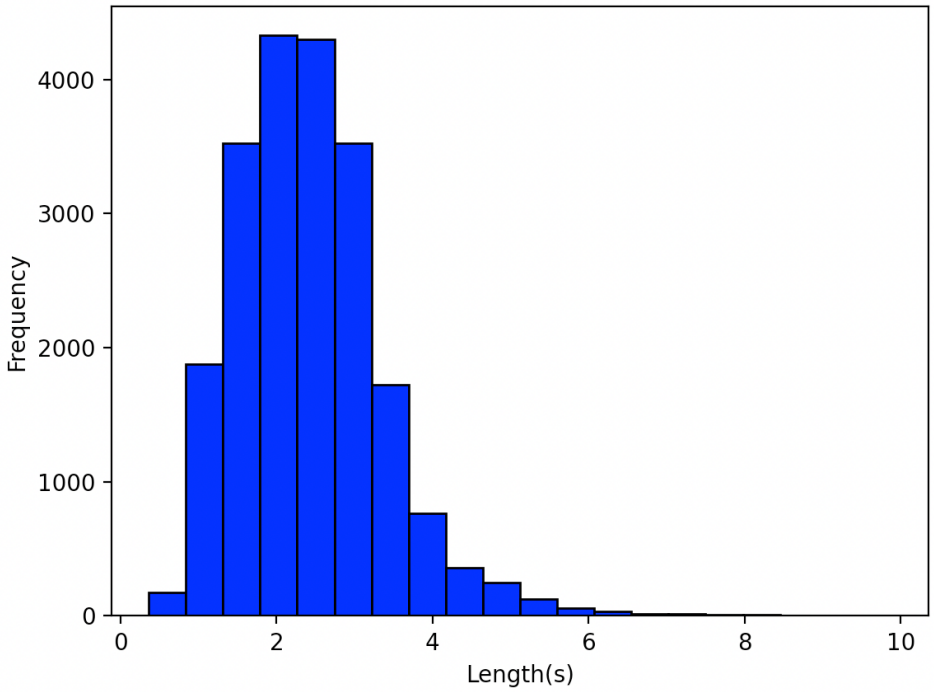

Figure 2.4: WLASL2000 video training data set length distribution (seconds)

The machine-learning model takes in extracted key point information from a set of video frames and generates the translation results as the top 5 most probable translations. This process enables the backend to handle the complex processing and manipulation of data, making it possible for the application to deliver near real-time translations of ASL to English text. The backend is built in the Python programming language, using OpenCV library to capture and prepare video data and Google MediaPipe library to extract the key points. Before sending the data to the machine learning model, backend standardizes the key point data to match the expected format. Each video frame is cropped such that the person's face is in the center and is scaled to 256 by 256 pixels before key points are located in each frame.

5.5 SPOTER: Real-Time Action Recognition ML Model

The SPOTER model is designed to recognize sign language gestures and translate them into English text at the word level. The model is based on a transformer architecture that uses the sign pose data as input to recognize signs. The sign poses are extracted using a 2D key point detection framework, which captures the movement of different body parts during a short video of a gesture in American Sign Language. We used Google MediaPipe library to extract key point data from the processed video frames. Each video frame records the location of body joints from the hands, face, and upper body, as a relative position in the video. Each frame contains 54 key points, with the majority of the key points located in the hands. There are 42 key points on both hands, 7 on the body, and 5 on the face. This means that the SPOTER captures detailed information about the location and shape of

each finger, while the body posture and facial expressions carry less information. Figure 2.5 shows the 21 key point locations on one hand.

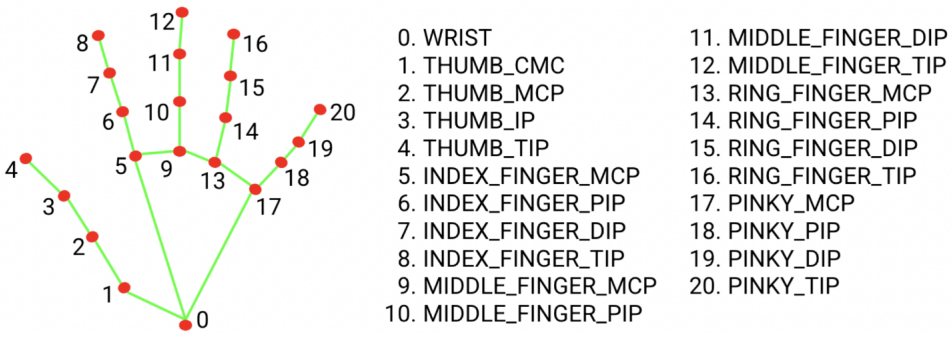

Figure 2.5: Extracted key point locations using Google MediaPipe library

Visualizing key points on top of a video frame looks similar to Figure 2.6. If there's no person in the video frame or a key point is hidden, we use the value of 0 to represent an absence of key points. The SPOTER model uses extracted joint coordinates as input to create a feature representation that it then processes in the transformer layers. The SPOTER model uses a self-attention mechanism, which allows it to focus on the most important parts of the input data while ignoring irrelevant information. The model also employs a bidirectional encoder that takes into account both the past and future sign poses to create a contextual understanding of the sign language sequence over time [1].

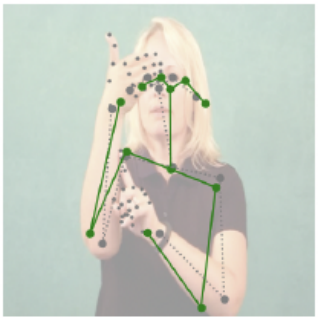

Figure 2.6: SPOTER model key point data extraction example (key points are shown in gray and green), from Figure 1 in [1]

We trained the SPOTER model on the WLASL2000 data set, which consists of 21,083 videos [10]. Each video represents 1 out of 2000 signs in ASL, performed by 119 distinct native signers or interpreters. Each sign is performed by at least 3 different signers. The data set was split into training, validation, and testing sets. The model was trained using a cross-entropy loss function and the Adam optimizer for 100 epochs. The highest validation accuracy the model achieved is 31.51%. Since not all signs were recognized with equal accuracy, we discarded the words that had 0% validation accuracy. We selected a subset of words with a positive recognition rate for user testing. The total number of signs with non-zero accuracy was 738. Out of 738 signs, 422 signs had validation accuracy of over 60% and 401 signs had 100% validation accuracy (around 54% signs with non-zero accuracy).

Figure 2.7 summarizes the SPOTER model validation accuracy results. Using these model training insights, we selected 10 words to run the first user application tests, described in the next section.

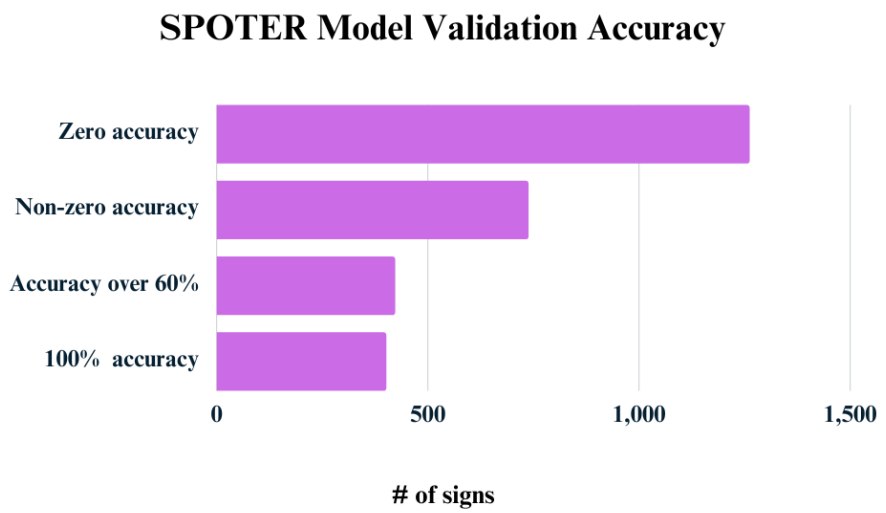

Figure 2.7: SPOTER model trained on WLASL2000 data set validation accuracy summary

6 ASL User Testing of SLAM in Real World Environments

The next step in the Human-centered Design approach is to test the prototype with users. We tested SLAM with a user who is fluent in ASL to receive feedback on the first iteration of the app. When completing the initial user research, the final question in the survey asked if the participants would like to test the product at a later time. We received interest in user testing from more than 30 people from the University of California, Berkeley, community and the ASL club at Berkeley. One user was able to test the SLAM application in person.

In order to simulate a real-world environment, the participant used SLAM in a Zoom meeting. SLAM was pre-installed into the Zoom client of the testing laptop and the user tested the application in a classroom setting. We provided the following list of 10 words for the participant to test: bird, onion, desk, presentation, lettuce, comfortable, late, report, computer, and mention. We then had the participant sign any word they would like. During user testing, the accuracy and latency of the machine learning model was calculated after each sign was translated. After the testing was finished, we asked the user for their feedback regarding the app.

6.1 SLAM Team Testing

Each member of the SLAM team tested the prototype in different locations, lighting, and Zoom virtual backgrounds. All of the locations were in a classroom setting and had a white background. The lighting varied from the user being back-lit

to uniformly lit. One team member tested SLAM with a slightly blurred virtual background. The accuracy of translated words from the set list ranged from 20 - 40% with an average accuracy of 30%. The average latency was 5 seconds after each sign was translated. The intended word for the user to sign, and the given translation from SLAM for each user is shown in the Table 2.1 below.

Word	User 1 Translation	User 2 Translation	User 3 Translation
bird	bird	Greece	bird
onion	hello	onion	onion
desk	desk	desk	desk
presentation	accent	lettuce	hello
lettuce	lettuce	stubborn	lettuce
comfortable	top	march	early
late	dive	engage	dive
report	star	go	go
computer	heaven	prevent	heaven
mention	read	hard	name

Table 2.1: A given word for the user to sign and the translation output from SLAM for each team member.

Each team member noticed the latency of 5 seconds. The speed of each sign being signed also varied across team members which influenced the results. The position, lighting, and the wearing of masks all influenced the accuracy of the translation as well. We found that the optimal setting for the user was to have bright lighting, a plain background, show half of their torso when signing, and wear no masks or face coverings.

6.2 ASL Survey Participant App Prototype Feedback

We repeated the same structure of testing for the user who is fluent in ASL. The average accuracy was 20% and the average latency was 5.4 seconds after each sign was translated. The intended word and given output of translation is shown as well as the latency of translation for each word in the Table 2.2 below.

Word	User Translation	Latency(s)
bird	priest	5.23
onion	crown	5.3
desk	desk	4.9
presentation	cabbage	5.27
lettuce	lettuce	5.17
comfortable	war	5.42
late	mature	6.76
report	peach	5.55
computer	paper	5.66
mention	pound	5.16

Table 2.2: A given word for the user who is fluent in ASL to sign, the translation output from SLAM, and the latency for each sign.

Initially, the user was too close to the computer, so we moved their location so their torso was shown on the screen. They also signed each sign very quickly as if they were having a conversation. This affected the accuracy of the translation, so we asked them to slow down and sign each sign multiple times during each translation. This allowed the model to recognize some words like "desk" and "lettuce".

6.3 SLAM App Redesign Based on Feedback

After the testing of specific words, we asked the user for any suggestions, improvements, or thoughts surrounding SLAM. Despite the average accuracy being 20% , the user expressed gratitude for the creation of this app. They also mentioned that ASL is difficult to discern in person, so they understood why the app would think "like" is "white". Inserting context into the machine-learning model would increase the accuracy of SLAM to translate many other similar looking words such as "Montana" and "museum".

Based on the user feedback, the following design changes would be beneficial in the next iteration of SLAM. The translation should be continuous so the user has to press the "start" button at the beginning of the meeting once. Additionally, the passing of data through API's causes a long latency for translation. To minimize the latency, there should be a restructuring of the API's connecting frontend to the backend. In order to combat the lack of context the model can discern, the new frontend can display the top two or three options for similar looking words. This allows for the user to choose which word they meant and have a more accurate translation.

7 Future Improvements to SLAM: Model and Functionalities

The SLAM app prototype is a successful first demonstration of automatic ASL to English translator using an existing virtual meetings platform. However, the technology has limitations. The biggest limitations are the available vocabulary and accuracy of translation. We envision that in the future, the app can be improved in a number of ways. The goal of the project was to build a prototype and as such, the SLAM app is complete. The shortcomings of the existing prototype come from both the technological aspect of the machine learning model and from limits imposed by the choice of distribution platform. In this study, we considered only people who would be using SLAM for translating American Sign Language into English. We envision that in the future, the app can be extended to other sign languages, given the requisite training data to develop an accurate machine-learning model.

7.1 Addressing Shortcomings of Vocabulary and Accuracy of the SPOTER Model

The core functionality of SLAM relies on the Sign Pose-based Transformer for word-level sign language recognition (SPOTER) machine learning model. The largest open American Sign Language data set, WLASL2000 [10], contains 2,000 signs and over 20,000 videos. However, having a vocabulary of 2000 is limiting in certain conversational contexts. For example, discussing a narrow topic that uses uncommon words would reduce the usefulness of SLAM. More training data is needed in order to extend the vocabulary of the app. Additionally, sign languages have dialects [25], so the translation learned by the model may not always correspond to what the signer intended because of the regional differences in the ASL they're using. Creating a larger vocabulary for SLAM exceeding 2,000 signs is essential for a product that can be used in schools and workplaces. For example, one use-case could be addressed by collecting and training a model to recognize signs based on a particular topic.

Moreover, the SPOTER model recognizes different signs with varying degree of accuracy. In order to hold a fluent conversation on any topic, more data is needed to train the model to be useful for signers in a general conversation. Additional complexities of recognition arise from language ambiguity, such as different words that share the same sign. Different regions of the United States have different dialects of American Sign Language. The accuracy of the translation also depends on environmental factors such as light settings, camera settings, how far away from the camera the signer is, and the setting the signer is in, among others. The speed with which the signer signs is a factor as well. Finally, our prototype sends captured sign data to the machine learning model at regular intervals. In practice, sign duration varies considerably from under a second to over 5 seconds. Improving the mechanism which sends the data for translation would improve the translation accuracy.

7.2 Extending App Functionalities and Broader Use Cases

With the SLAM prototype complete, there are a number of ways in which the application can be extended. First, other sign languages can be added such as the

British Sign Language, the German Sign Language, or any other sign language. The current challenge is collecting a large data set that enables a degree of translation accurate enough to hold a conversation. Another idea for future work is to add user-specific vocabulary, based on repeated use of signs, akin to auto-correct on phones. This idea can be explored by letting the user provide their data. Extending an app in this way would require modifying and training the machine learning model to recognize user-specific signs.

Additionally, extending the ASL data set can improve the SLAM functionality. For example, collecting data from a diverse group of people of various sex, age, ethnicity, using solid as well as more realistic backgrounds would help make a real-world application more accurate in any setting and for any user. The speed with which the signer signs also influences the translation and training. Collecting data from people who sign at different speeds would enhance the application functionality.

Finally, since Zoom is one of many virtual meeting platforms, extending the application to other platform is a possible future work. For example, the same app can function inside Google Meets, Microsoft Team, or Apple FaceTime. In the future, including other devices such as mobile phones and tablets would bring ASL to English translation to people on the go. The prototype of Zoom app makes it possible to think of other future uses of the real-time ASL translation, such as self-checkout cashier registers in grocery stores, airport check-in ticket booths, or drive-through food kiosks.

References

[1] M. Boháček and M. Hrúz, "Sign pose-based transformer for word-level sign language recognition," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 182–191.

[2] W. H. Organization. "Deafness and hearing loss." (), [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss. (accessed: 02.26.2023).

[3] C. Schoenborn and K. Heyman. "Health disparities among adults with hearing loss: United states, 2000-2006." (), [Online]. Available: https://www.cdc.gov/nchs/data/hestat/hearing00-06/hearing00-06.htm. (accessed: 02.26.2023).

[4] R. E. Mitchell, T. A. Young, B. Bachleda, and M. A. Karchmer, "How many people use asl in the united states? why estimates need updating," *Sign Language Studies*, vol. 6, pp. 306–335, 2006.

[5] J. P. Robinson, W. P. Rivers, and R. D. Brecht, "Speaking foreign languages in the united states: Correlates, trends, and possible consequences," *The Modern Language Journal*, vol. 90, pp. 457–472, 2006.

[6] R. Wolfe, J. C. McDonald, T. Hanke, *et al.*, "Sign language avatars: A question of representation," *Information*, vol. 13, no. 4, 2022, ISSN: 2078-2489. DOI: 10.3390/info13040206. [Online]. Available: https://www.mdpi.com/2078-2489/13/4/206.

[7] S. Saha, R. Lahiri, A. Konar, and A. K. Nagar, "A novel approach to american sign language recognition using madaline neural network," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016, pp. 1–6. DOI: 10.1109/SSCI.2016.7850121.

[8] C. Zhang, X. Yang, and Y. Tian, "Histogram of 3d facets: A characteristic descriptor for hand gesture recognition," in *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 2013, pp. 1–8. DOI: 10.1109/FG.2013.6553754.

[9] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4724–4733. DOI: 10.1109/CVPR.2017.502.

[10] D. Li, C. Rodriguez, X. Yu, and H. Li, "Word-level deep sign language recognition from video: A new large-scale dataset and methods comparison," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2020, pp. 1459–1469.

[11] A. Williams, "User-centered design, activity-centered design, and goal-directed design: A review of three methods for designing web applications," in *Proceedings of the 27th ACM international conference on Design of communication*, 2009, pp. 1–8.

[12] M. Melles, A. Albayrak, and R. Goossens, "Innovating health care: Key characteristics of human-centered design," *International Journal for Quality in Health Care*, vol. 33, no. Supplement_1, pp. 37–44, 2021.

[13] I. Göttgens and S. Oertelt-Prigione, "The application of human-centered design approaches in health research and innovation: A narrative review of current practices," *JMIR mHealth and uHealth*, vol. 9, no. 12, e28102, 2021.

[14] A. Van Pelt and J. Hey, "Using triz and human-centered design for consumer product development," *Procedia Engineering*, vol. 9, pp. 688–693, 2011.

[15] M. E. Raven and A. Flanders, "Using contextual inquiry to learn about your audiences," *ACM SIGDOC Asterisk Journal of Computer Documentation*, vol. 20, no. 1, pp. 1–13, 1996.

[16] S. Blomkvist, "Persona–an overview," *Retrieved November*, vol. 22, p. 2004, 2002.

[17] M. Haesen, J. Meskens, K. Luyten, and K. Coninx, "Draw me a storyboard: Incorporating principles & techniques of comics...," *Proceedings of HCI 2010 24*, pp. 133–142, 2010.

[18] D. Waters, R. Campbell, C. M. Capek, *et al.*, "Fingerspelling, signed language, text and picture processing in deaf native signers: The role of the mid-fusiform gyrus," *Neuroimage*, vol. 35, no. 3, pp. 1287–1302, 2007.

[19] D. R. Moogk, "Minimum viable product and the importance of experimentation in technology startups," *Technology Innovation Management Review*, vol. 2, no. 3, 2012.

[20] Zoom. "Zoom platform." (2023), [Online]. Available: `https://zoom.us/` (visited on 02/25/2023).

[21] S. S. Chaudhary, D. Panwar, S. Gautam, and S. Sundriyal, "Zoom video communications inc: Sustaining competitive advantage by addressing and enhancing user privacy and security: A case study," *Academy of Marketing Studies Journal*, vol. 26, no. 3, 2022.

[22] A. Chawla, "Coronavirus (covid-19)–'zoom'application boon or bane," *Available at SSRN 3606716*, 2020.

[23] M. P. Robillard, "What makes apis hard to learn? answers from developers," *IEEE software*, vol. 26, no. 6, pp. 27–34, 2009.

[24] J. Kopeckỳ, K. Gomadam, and T. Vitvar, "Hrests: An html microformat for describing restful web services," in *2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, IEEE, vol. 1, 2008, pp. 619–625.

[25] R. Stamp, A. Schembri, B. G. Evans, and K. Cormier, "Regional Sign Language Varieties in Contact: Investigating Patterns of Accommodation," *The Journal of Deaf Studies and Deaf Education*, vol. 21, no. 1, pp. 70–82, Sep. 2015, ISSN: 1081-4159. DOI: 10.1093/deafed/env043. eprint: https://academic.oup.com/jdsde/article-pdf/21/1/70/7142260/env043.pdf. [Online]. Available: https://doi.org/10.1093/deafed/env043.

Appendix A

SLAM User Persona

Name: Olivia Adams
Tagline: ASL Educator
Demographics:
Age: 22
Gender: Female
Geography: small town in Idaho, USA, now lives in Berkeley

Short Bio:
Deaf, raised by Deaf parents, raised in an integrated school because there were not any deaf schools in her home town. She is a fluent / native ASL signer. She didn't have many deaf friends growing up, and she's making many friends both hearing and non-hearing at Berkeley.

Occupation: Student at UC Berkeley
Major: Special Education
Minor: Biology
Income: part-time dog walker ($500 a month)

Personal characteristics:
- Communicates with her mom several times per week in ASL using FaceTime
- Extraverted, bubbly, and likes to get outside
- Open-minded
- Community leader (works with kids in her neighborhood, teaching ASL classes)

Key Cares & Concerns – Motivations:
- They are Deaf so they would like to support their community
- Part of many social clubs
- They have deaf and non-deaf friends
- Society and culture shift towards online meetings and they are concerned about inclusiveness on those platforms

Existing solutions they already use:
- Face-to-face and FaceTime, relying on translators and often choose to text instead

Challenges & Pain Points:
- Can't join Zoom meetings unless they hire an in-person translator, which is expensive and time consuming
- Feel excluded in day to day life
- Wish more people learned the basics of ASL for simple conversations (classmates, cashiers)
- Typing out what they want to say is annoying, slow, not fluid