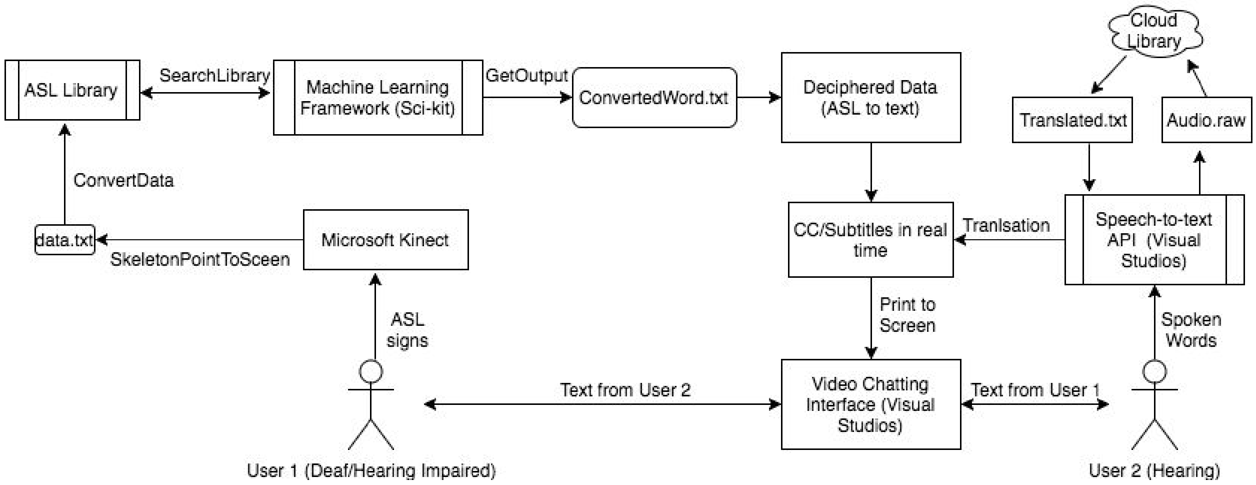
Overall project

The overall project involves creating a video chat interface that will translate American Sign Language (ASL) into text on one side, and it will convert speech to text on the other side. This will allow deaf and hearing-impaired individuals to communicate with hearing individuals that do not understand sign language. This project is broken down into three main modules: converting ASL to text, converting speech to text, and integrating the previous modules into a video chat interface. This entire project will serve for the single use case of a deaf or hearing impaired individual telecommunicating with a hearing user via video chat.

The figure below shows the overview of how this system works. User 1, which is the deaf/ hearing impaired individual, will perform sign language into the Microsoft Kinect Sensor. The Kinect sensor will send raw data into the machine learning framework. The Kinect sensor gets the data by its coordinates in the form of x, y and depth. The points are written to a text file and the ASL Library holds those files. SciKit learn is then able to read the text files and train the data to interpret the meaning of the word. This machine learning technique will use the ASL library, which will be generated by Elizabeth and Chelsea, to interpret what the user is trying to say. The results from the framework will be sent to a cs (C-sharp) file, where a script will read the words and show it on the screen for the other user. This will happen through the video chatting interface. Similarly, User 2, the hearing individual, will talk normally into the video interface. The speech to text API will convert the spoken word into text in real time, and save the text to the cs file for the script to read from. The speech to text API is also ran in Visual Studios and takes in the audio, converts it to a readable file, and finds the translated meaning from a cloud library that already exists. The translated words are correctly back to the user in the format of printed text.



Components:

This project is broken down into three main components, the speech-to-text, the machine learning and ASL library, and the video transmission software.

Speech To Text

For this module, the Google Speech API will convert a person's spoken word into text. The person will press record to start recording, and then they will save it to an audio.raw file. The raw file is then sent to Google's Cloud Speech API, translated into text, and sent to the display. The goal for this part is to have it completed by the end of January. As part of this API, a few methods that will be used are SpeechClientCreate(), Recognize(), and StreamingRecognitionConfig(), and SpeechClientClose().

SpeechClientCreate() - creates an instance of the SpeechClient object with default settings, set by Google.
Recognize() - analyzes the speech as it is being detected.
RecognitionConfig() - transforms the speech into text.
SpeechClientClose() - deletes the instance of the SpeechClient Object.

SciKit Learn SVM/ASL To Text/ASL Library API

EC-Chat will contain an ASL library that SciKit learn can use to classify the data. This library will be an open source library as it will need to continue growing as the application is more used. The beta version of EC-Chat will include an ASL library that contains thirty signs, each composed of ten different users. This library and software is built under the assumption of further growth. Each sign will be held in the format of a .txt file, which converts to a .csv file, which the SciKit learn machine learning library will be able to read, associate the English meaning to it.

When the user is signing their words, the software will interpret the sign by reading and plotting the users points in the format of x, y and depth, and save to a text file. The SciKit learn will see which existing word is the best fit for the incoming sign, fetch the translation and the output of the function will be the translated text, in the form of another text file. The string of word(s) will be sent to the video chatting interface, which will be able to relay them appropriately to the other user.

The EC-Chat team will maintain their application certified set of ASL signs for the library on their open source page. Any individual or group who wishes to contribute may add to the github repository, however only the EC-Chat team may modify the EC-Chat certified signs. This is an attempt to ensure that the content is accurate.

Users may add to this library by recording the signing of a certain vocabulary word. The video will have specifications on format and type accepted. The English translation must accompany the ASL sign. The EC-Chat software will have the ability to convert the video into the csv or text format required for the SciKit learn to read. As more data is added to the library, English translation of the word will have more signs to add to the correlating word. Due to SciKit learn's supervised learning method of using support vector machines, the more videos that are added, the more accurate the estimations will be for further use of the software.

There will be an implemented control to users uploading translations to the library to ensure there are not false or incorrect data that may influence the translations. This will be done by having auditors review and implemented training into our software. To adhere to our timeline, Elizabeth and Chelsea plan to have the simplest requirements completed by the end of February. This requires having a text file of data interpreted by SciKit learn with real data.

Video Transmission / Integration

This module will require the use an open source video chat interface. The text that is received from the text to speech module will come directly from the speech API referenced above. The text from the ASL component will be written to a text file, and the interface will read from that text file to output the signed words onto the screen. The expected completion time for this is around the middle of March.

Core Algorithm

This project does not have a single core algorithm. Each component has its own project that was individually built and integrated with the other components. The core pipeline is the integration of the three bases components working together to communicate seamlessly with the other portions.

Timeline

Each component is currently in development. The projected completion for the project as a whole is the end of March. By the end of March, EC-Chat should be ready to use for simple signs such as the alphabet. The longest module is the ASL to Text because there are many moving parts to it such as creating an ASL library and training SciKit to understand certain signs.