ASLearn

Team E5: Hinna Hafiz, Aishwarya Joshi, & Valeria Salinas
18-500 Capstone Design, Spring 2022
Electrical and Computer Engineering Department
Carnegie Mellon University

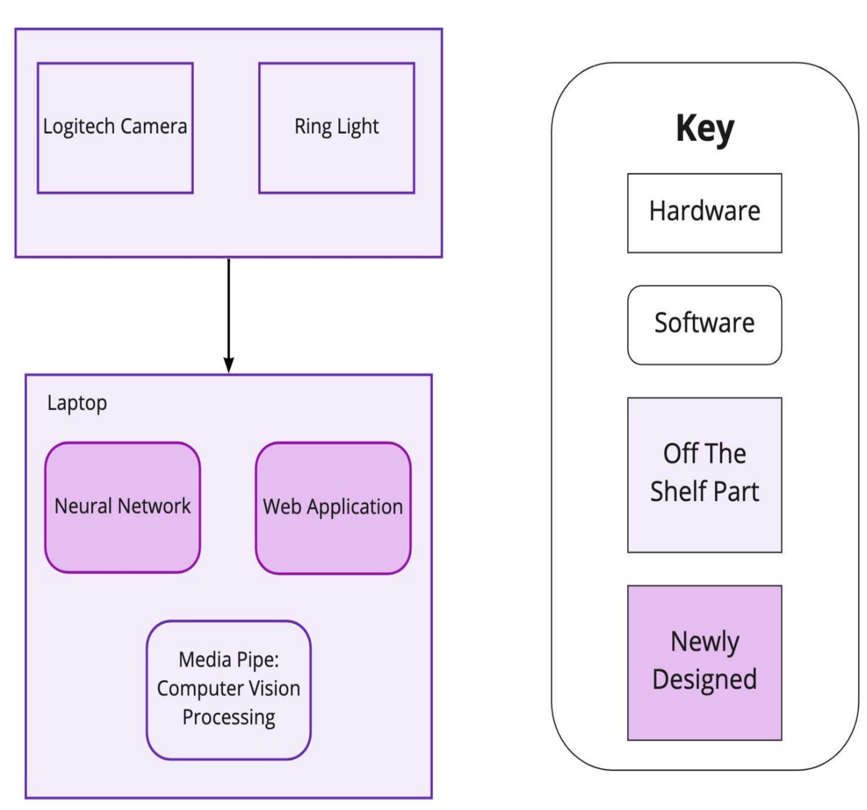
Product Pitch

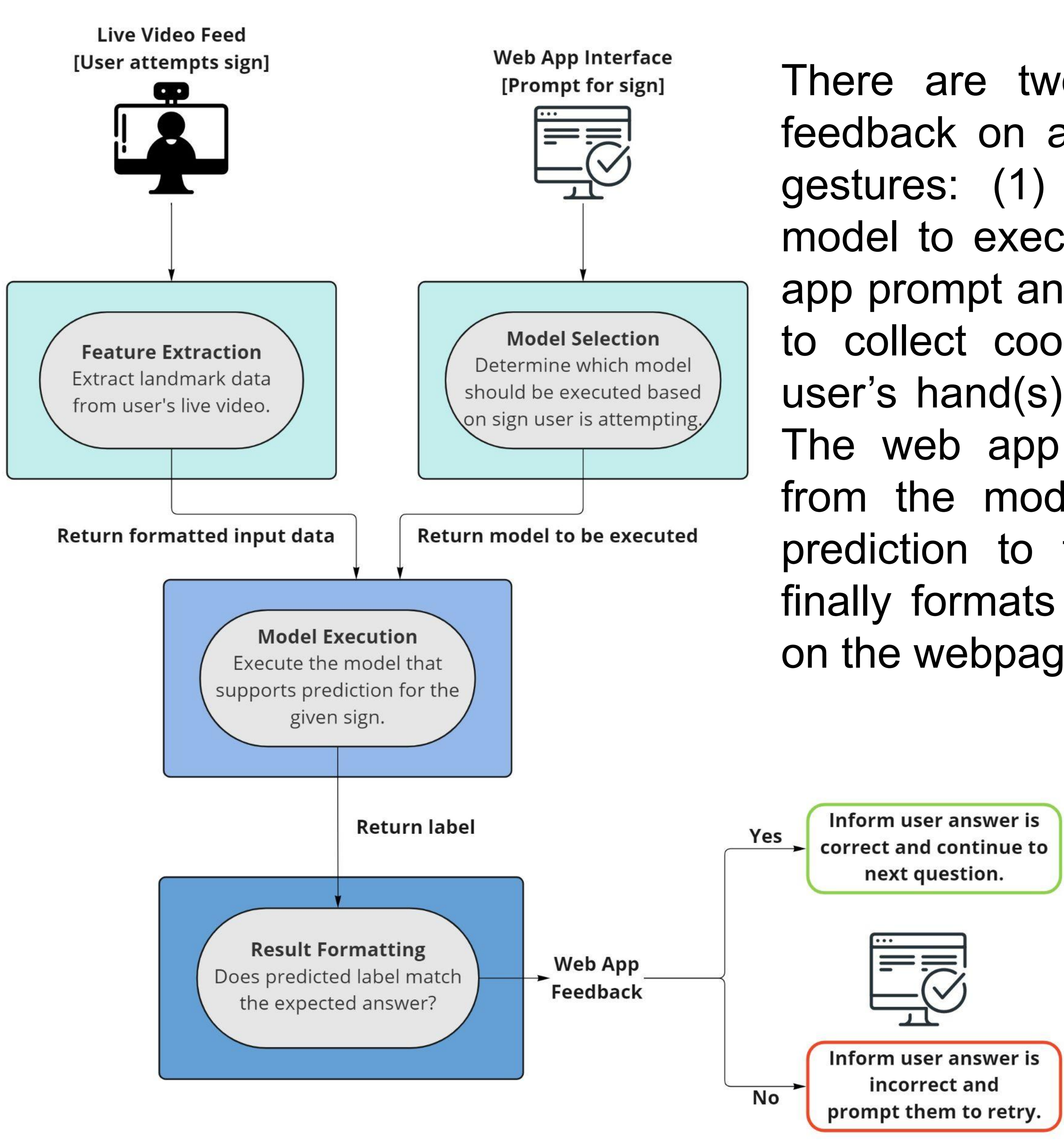
ASLearn, our learning platform for American Sign Language, aims to combine the flexibility of remote learning with the interactivity of live feedback to give users an effective, engaging experience in learning ASL. ASLearn supports learning, testing, and quizzing for over 50 signs with instructional videos and text. Additionally, unlike other ASL learning platforms, we provide a live webcam feed embedded into the platform so users can watch themselves do the signs side-by-side with our team of expert instructors. Furthermore, we use computer vision and machine learning to detect if the users are signing correctly, where they can get feedback in as little as 0.5 seconds.

With ASLearn, we can help the hearing community meet the hard-of-hearing community halfway in communication.

System Architecture

The user video feed from a camera will be embedded in our web app interface. This video is also sent to a computer vision processing component that will obtain meaningful data from it. This data is sent to the machine learning component of the system as inputs to a neural network model, which is selected from among 7 neural networks that each support prediction generation for a subset of sign language. The neural network makes a prediction for what sign the user is making, and this result is returned to the web application component.



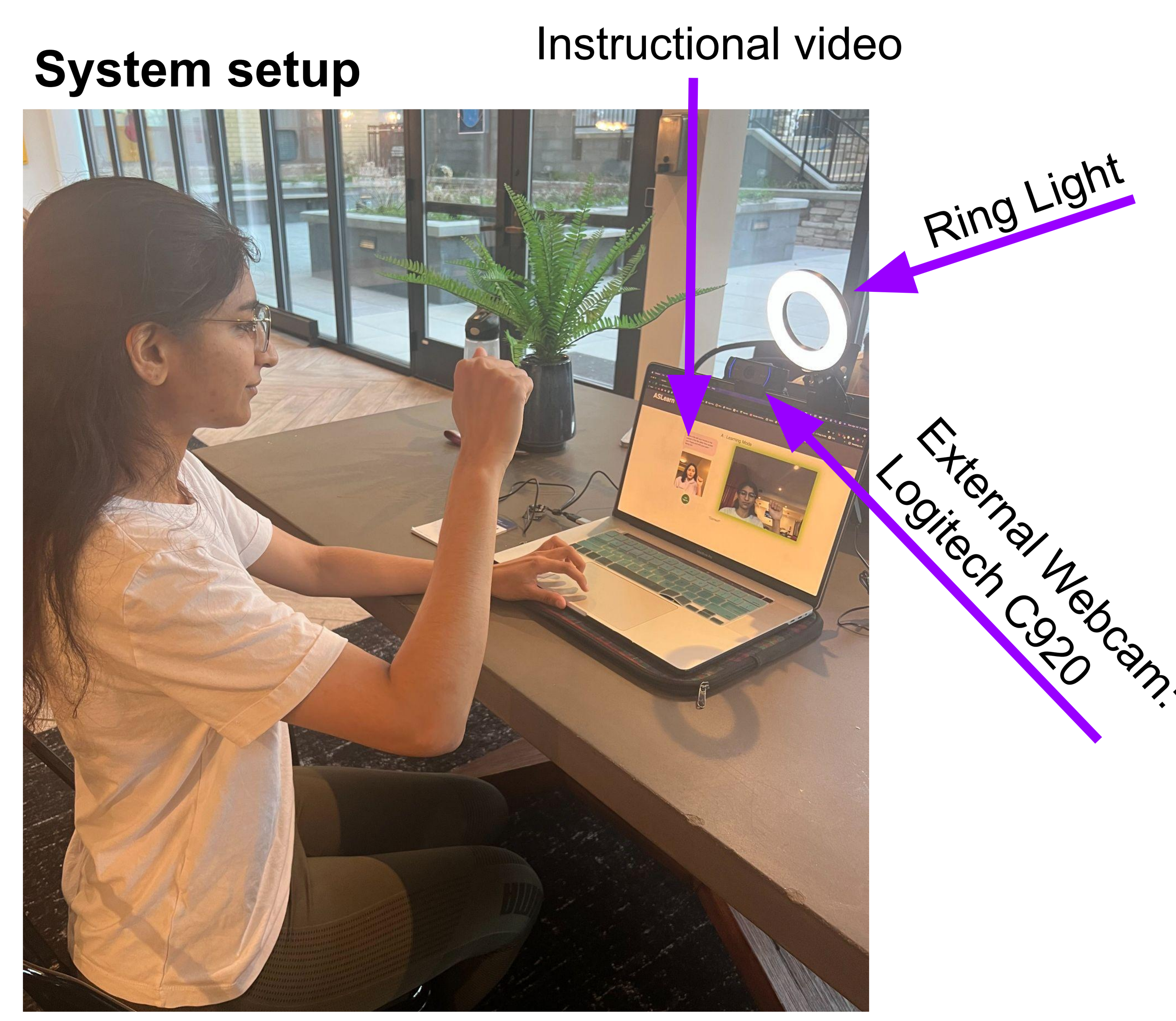


There are two steps for providing feedback on a user's sign language gestures: (1) selecting the correct model to execute based on the web app prompt and (2) feature extraction to collect coordinate data from the user's hand(s) to pass to the model. The web app retrieves a response from the model and compares the prediction to the prompted sign. It finally formats a response to display on the webpage.

System Description

We convert the user video input through MediaPipe to get landmarks of the hands present in the video. We are also using machine learning models that we trained with sign language data from outside sources for static signs and data we created ourselves for static and dynamic signs. These machine learning models can detect static and dynamic signs and give the user feedback on whether their sign was correct. As for our web application, we have a fully functional, locally hosted web application that contains lesson plans for four topics using AJAX to help with usability. Below is a picture showing how this setup looks.

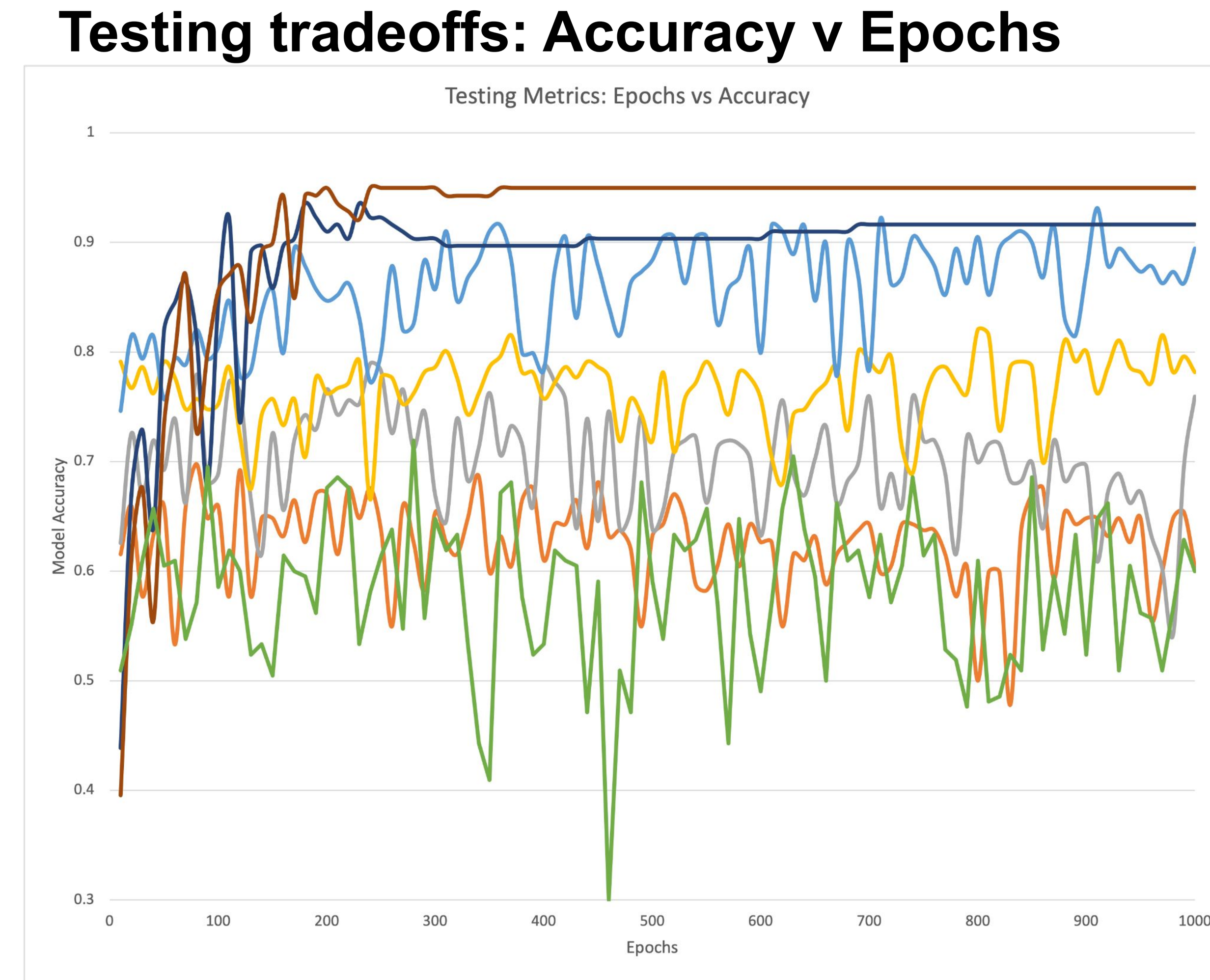
System setup



System Evaluation

For tradeoffs, we decided to test if our accuracy changes depending on the number of epochs that we run our model through. For the testing data, we can see that the accuracy is all over the place for all the models depending on the epochs.

Testing tradeoffs: Accuracy v Epochs



Conclusions & Additional Information

As a team, the lessons we have learned are that data sets ideally should be diverse and comprehensive, vetting our data sets to make sure its up to standard before training, and consider the latency affect from communication between subsystems. For the future work, we hope more dynamic signs could be added into the neural network. Additionally, other possibilities are to deploy web application to the cloud, checking if the dominant hand is doing the primary motion for two-handed dynamic signs, and adding facial recognition so that users emote when they sign.

http://course.ece.cmu.edu/~ece500/projects/s22-teame5/

Check this QR code to see our blog and learn more about our project!

Electrical & Computer ENGINEERING
Carnegie Mellon