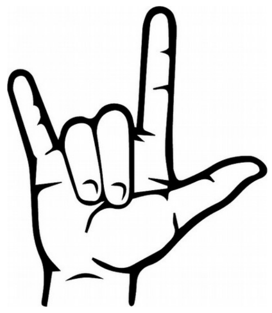
Software Requirements Specification

for

Sign Language Predictor

with Special Thanks to Professor Lubman, CSUF



Prepared by

Alexander Athas
Phuc Cong Le
Jeffrey Liv
Jeevitha Gudivada
Anushree Ankola
Vineeth Vasudevan
Elizabeth Tsan

Table of Contents

Table of Figures

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) details the descriptions of the requirements of our software application, known as Sign Language Predictor, in order to capture the needs and desires of the customer. This SRS establishes an agreed upon set of requirements for the development of Sign Language Predictor.

1.2 Intended Audience

The intended audience for this document is Dr. Lidia Morrison, the members of the design team (Anushree Ankola, Alex Athas, Jeevitha Gudivada, Liam Le, Jeffrey Liv, Elizabeth Tsan, and Vineeth Vasudevan) and the customer. This document will provide a clear understanding of the application to the customer and Dr. Lidia Morrison while providing a guideline for the members of the design team to develop the application. To gain an overall understanding of the application, proceed to read from the beginning of the document to the end concentrating on section 2. The intended audience of the application are non-signing English speakers. The application is geared towards those speakers of English in the United States having interactions with Deaf and Hard-of-Hearing signers that use American Sign Language (ASL).

1.3 Product Scope

Sign Language Predictor is a Machine Learning model to assist in determining the sign associated with the English letter or number, effectively creating subtitles. This program is designed to allow a non-signing English speaker to understand a signer of ASL. As the population of the United States ages and diversifies, more and more people will need to learn alternate methods of communication be they Hard-of-Hearing (HOH), Deaf or non-signing English speakers.This program is designed to predict basic signs including the alphabet and numbers 1-10.

1.4 References

1. SRS Template www.csc.villanova.edu/~tway/courses/csc4700/s2008/handouts/srs_template.doc
2. CS462 Lecture Power Point presentation on Use Cases
   - Deriving_Use_Cases_from_Requirements_part_1.ppt McGraw-Hill Companies, Inc.
   - Deriving_Use_Cases_from_Requirements_part_2.ppt McGraw-Hill Companies, Inc.
3. Use Case Template stlouis.iiba.org/download/UseCase.dot
4. TensorFlow https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0
5. Google's Machine Learning Youtube Video Series https://www.youtube.com/watch?v=cKxRvEZd3Mw&list=PLT6elRN3Aer7ncFlaCz8Zz-4B5cnsrOMt
6. Digital Ocean https://developers.digitalocean.com/documentation/v2/
7. Django https://docs.djangoproject.com/en/2.0/
8. Github https://github.com/googlecodelabs/tensorflow-for-poets-2

2. Overall Description

2.1 User Objectives

The user is provided with an interactive web interface with icons and text detailing what options they have. The options are uploading a sign language sign, a video, train new data, and see what the result is of what a sign or gesture is.

2.2 Product Functions

The primary goal of the Sign Language Predictor is to aid in understanding what a hearing impaired person is trying to convey to someone who does not understand sign language. The web application is designed to be simple and easy to understand for anyone to use, and to understand the basic alphabet, the numbers 1 through 10.

The following is a list of functions of Sign Language Predictor:
- Sign Language Predictor allows new static images to be uploaded
- Sign Language Predictor will enable videos to be uploaded
- If there has been new data added the web application also has a train button for the new data to help improve the accuracy of the AI.
- The web application will also display what the letter or number is and the accuracy of the prediction on the results page.

2.3 Operating Environment

Sign Language Predictor is a web application that runs on Linux systems, and the computers or devices must have a rear-facing camera for video or static images to be taken for the prediction. It must be the current version of Linux Ubuntu which is 16.04, and since Linux and Ubuntu are open sources anyone can download, so there is greater access to this application.

2.4 User Characteristics.

Anyone who has a computer with Linux can have access to the web app, there are no limitations to what someone is allowed to use on the web application.

2.5 Design and Implementation Constraints

The system is limited to the Linux operating system. The system can support running on a personal computer running linux.

The interface of the system will be limited to keyboard and mouse. The user will interact with the system solely through the keyboard and mouse on the device.

The system will only be able to handle uploads of static images and short video clips.

2.6 Assumptions and Dependencies

This SRS assumes that the reader has a basic understanding of the Linux operating system and the ability to navigate and operate a Linux system. This document will not discuss the requirements of the Linux system or its usability and navigability. This document will focus on the details of the web application (Sign Language Predictor) which will run on the Linux operating system.

3. Functional Requirements:

3.1 Use Cases

System shall display the main page/home page

Description-
The system displays a main home screen with the title of the project “Sign Language Predictor” and other options for Sign Language Prediction.
Technical conditions-
Precondition- the software is fired up on a Linux environment by connecting to the server keying a set of python commands on the terminal.
Post condition- The main screen of the software is displayed after which the user can select the required options for exploring the functionality of the software.
Risks-
Unsuccessful connection to the server results in improper execution of the software or the main screen page.
Dependencies-
None

System shall allow the user to upload a sign

Description-
System allows the user to upload a sign (alphabet or letter) to its respective folder.
Technical conditions-
Precondition- the home page is displayed, and the upload option is selected where the user selects desired images (signs).
Post condition- The selected images are uploaded to the desired folders on a local drive.
Risks-
Careless upload may result in destroying the result set or output of the software.
Dependencies-
None

System shall allow the user to perform live prediction

Description-
System allows the user to connect to the webcam of the system for live sign language prediction.
Technical conditions-
Precondition- the home page is displayed, and the video option is selected which triggers the system webcam.
Post condition- The results (signs) are predicted and displayed on the screen

with its corresponding accuracy with other similar alphabets or letters.
Risks-
Too many hand gestures produce bad results or predictions.
Dependencies-
None

System shall allow the user to train or upload new data
Description-
System allows the user to connect to the webcam of the system for capturing a series of images to be uploaded to their respective folders indeed helps in training the model.
Technical conditions-
Precondition- the home page is displayed, and the insert/train data option is selected which lets the user to capture images in burst mode.
Post condition- The desired images are selected and uploaded to their respective folders.
Risks-
Careless upload may result in destroying the result set or output of the software.
Dependencies-
None

3.2 Use Case Diagram

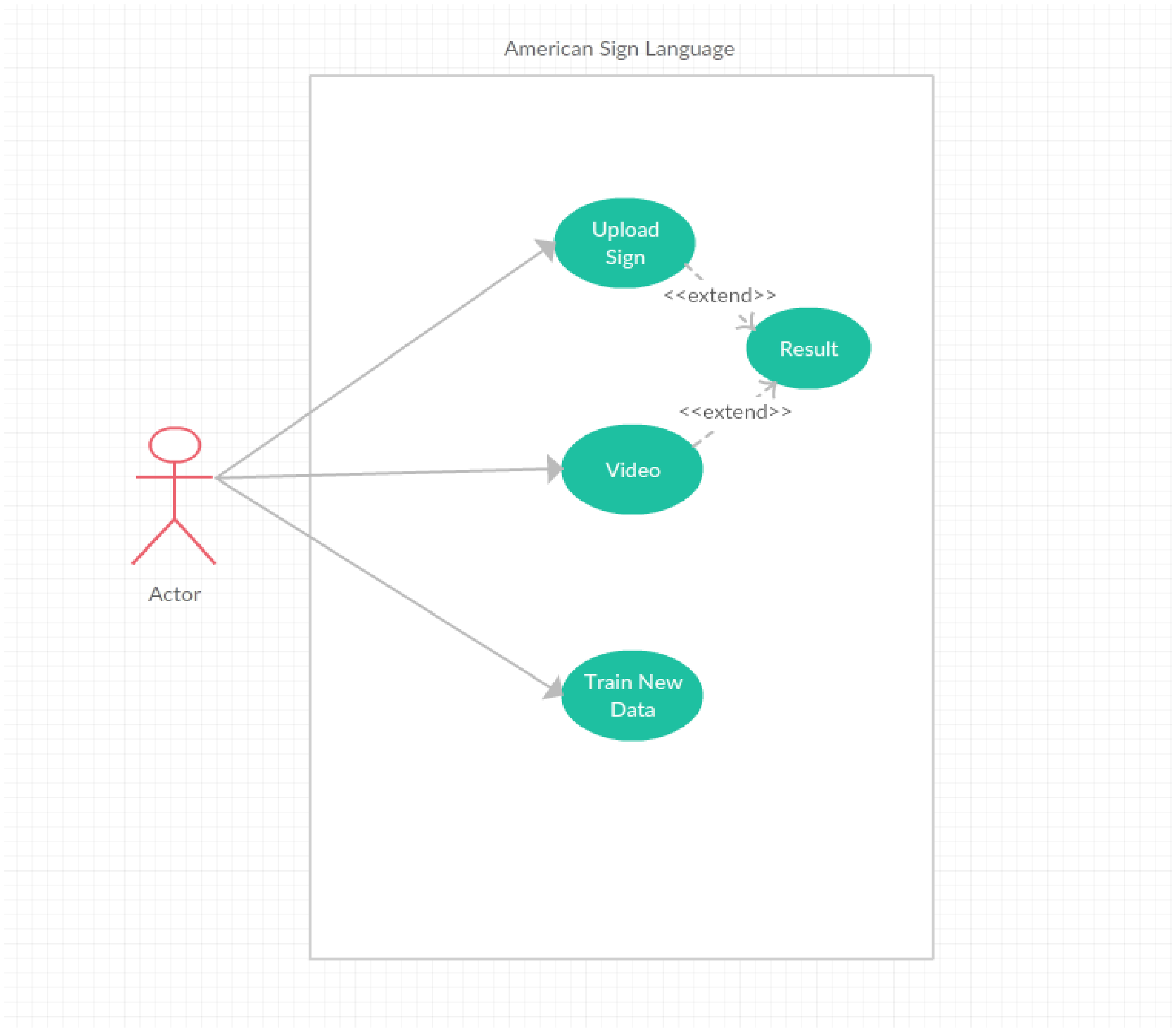


Figure 3.2-1 Use Case Diagram for American Sign Language

4. External Interface Requirements

4.1 User Interfaces

The user interface shall consist of buttons, words and pictures. The application shall utilize the menus to get input from the user and provide text feedback to the user. The GUI shall consist of a main menu and submenus. All menus shall consist of buttons which allow the user to navigate the system.

4.2 Hardware Interfaces

The users of the system shall have a device (laptop or desktop) which runs the Linux OS and meets the minimum specification requirements. The device shall have a functioning camera.

The device shall have the following minimum specifications:

- *1 GHz CPU*
- *512 MB RAM*
- *1 GB Free Internal Storage*

4.3 Software Interfaces

The users of the system shall have the stated Linux OS version.

5. Non-Functional Requirements:

Security:
The application requires the user to upload signs basically images of good clarity and to their respective folders for accurate predictions.
All the data stored on the hard disk are susceptible to modifications and thereby rendering bad result.

Usability:
After starting the application (i.e. The software being invoked) they take less than few seconds to load the UI and other components.

6. Diagrams

6.1 State Diagram

The following state diagrams will visually explain the states that our system will be in during the usage of our system. Ovals will represent actions that will be taken by our system. Arrows dictate the direction in which the system will proceed. Diamonds represent decision nodes and will have either two choices, yes or no. The starting state is represented by a single bold dot. The end state is represented as a single bold dot with a circle surrounding it. The state diagram will begin at the starting state and will end and the end state.

6.1.1 Translating ASL in Still Images

This state diagram shows the states and the flow of events that occur while a user wants to translate ASL in still images.

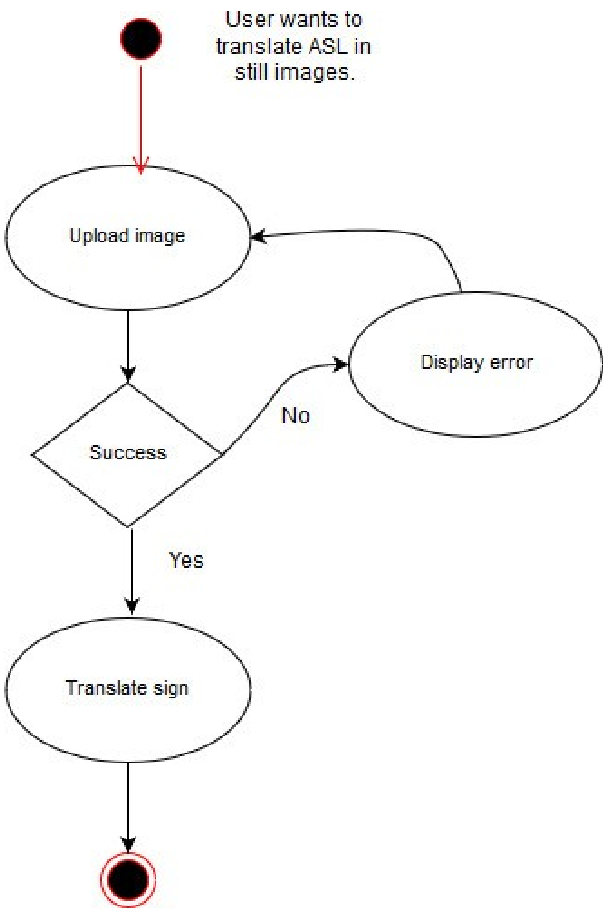


Figure 6.1-1 Translating ASL in Still Images State Diagram

6.1.2. Translating ASL Using Camera Feed

This state diagram will show the states involved when a user wants to translate ASL using a camera feed.

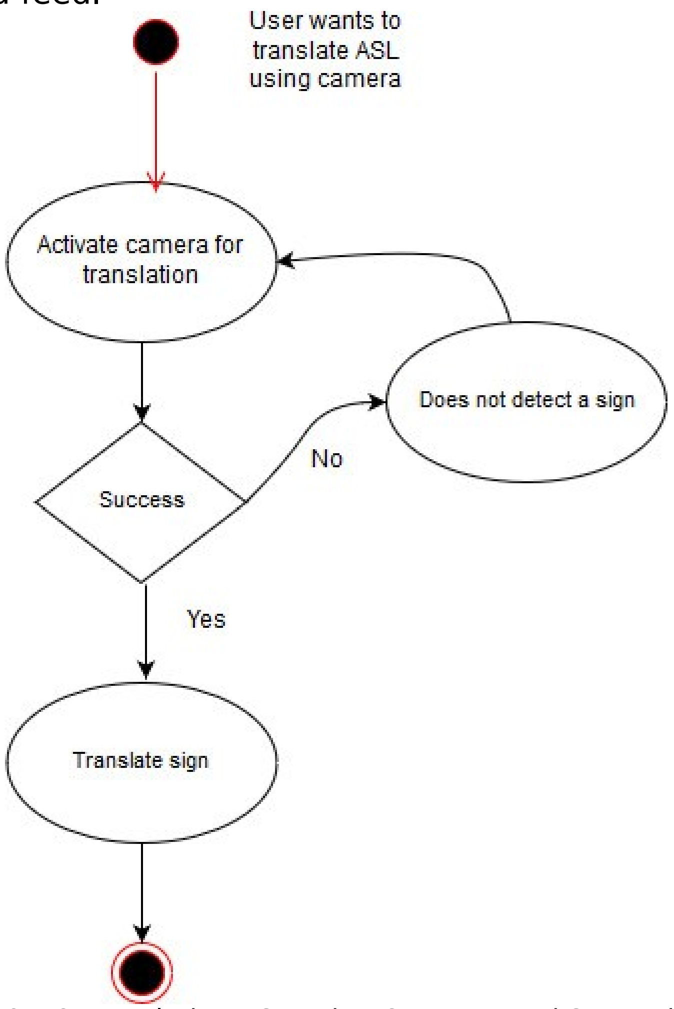

User wants to translate ASL using camera
Activate camera for translation
Does not detect a sign
No
Success
Yes
Translate sign

Figure 6.1-2 Translating ASL Using Camera Feed State Diagram

6.1.3 Training Data

This state diagram will show the states involved when a user would like to train data.

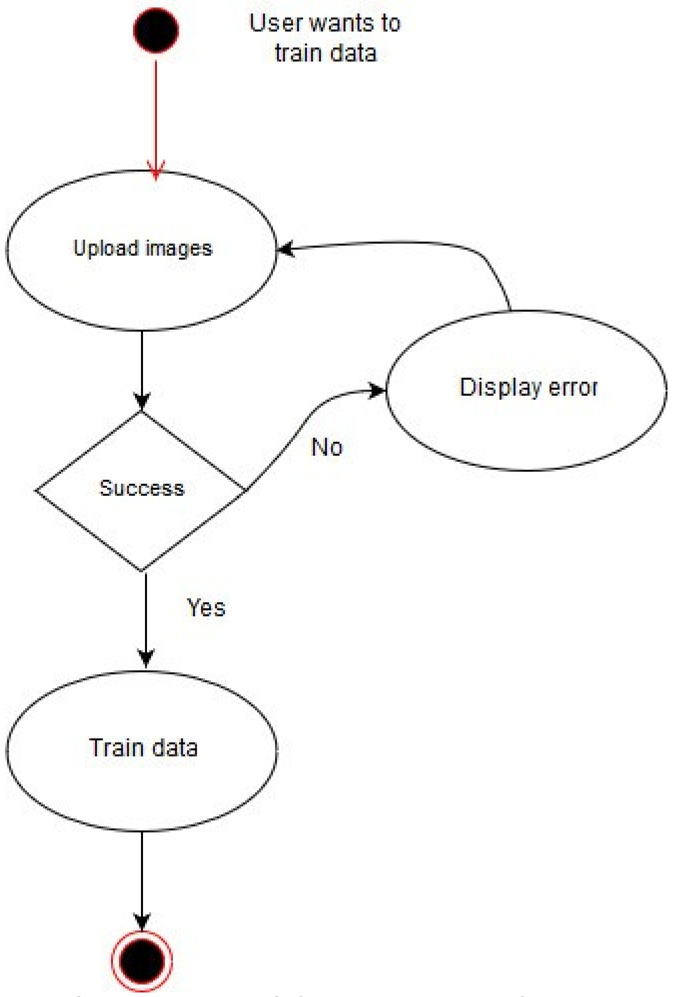


Figure 6.1-3 Training Data State Diagram

7. Operational Scenario

7.1 Successful Scenario

In a successful scenario, the user of the system will access the website's homepage, choose whether to upload pictures of sign language, upload video of sign language, or train new data. The upload feature allows users to upload still pictures of American Sign Language (ASL). Upon a successful upload, the predicted translation of the sign and its prediction accuracy will be displayed on the screen. The screen will also display suggested letters that the translation may be along with the corresponding accuracies. The video feature will allow users to use video feed to sign ASL. Upon a successfully read sign, the predicted translation of the sign and its prediction accuracy will be displayed on the screen. Once again, the screen will also display suggested letters that the translation may be along with their respective accuracies.

7.2 User Manual

1. To begin, **Select** an option: upload sign, video, train new data.

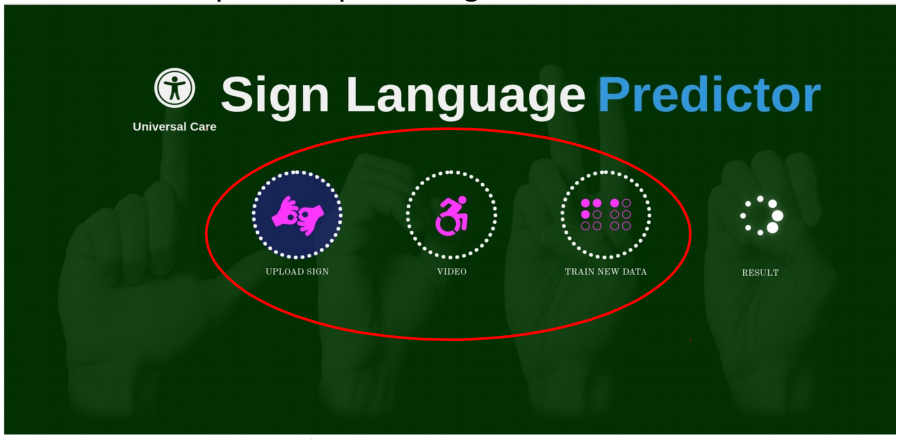


Figure 7.2-1 Homepage

2. To navigate through the website, we will be demonstrating the features of our Sign Language Predictor in a left-to-right fashion. We will first **Select** the "upload sign" button. Once we are on the upload page, we **Select** the upload button and upload a still image containing an ASL sign.

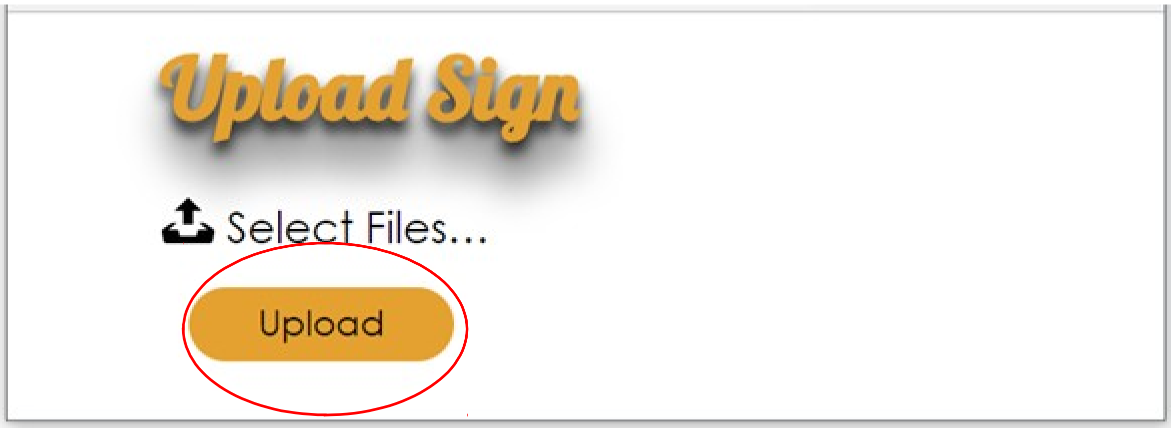
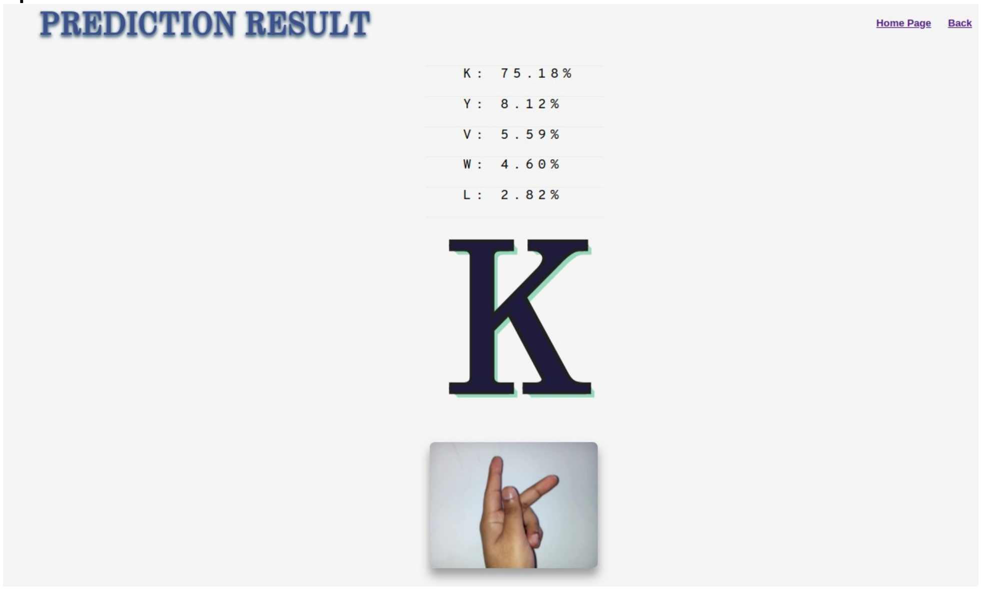


Figure 7.2-2 Upload Page

3. Once the still image has been uploaded, the resulting prediction will be displayed on the screen along with the accuracy of the prediction. There will also be other suggestions of letters that the sign may be along with the respective accuracies.



```
K :  75.18%
Y :  8.12%
V :  5.59%
W :  4.60%
L :  2.82%
```

Figure 7.2-3 Prediction Result

4. The next feature, video, can be accessed by selecting the video button on the homepage. Once on the video page, we will be able to use our device’s camera feed to capture ASL signs. Once a sign has been detected on-camera, the resulting prediction will be displayed similar to the results of the still image prediction.

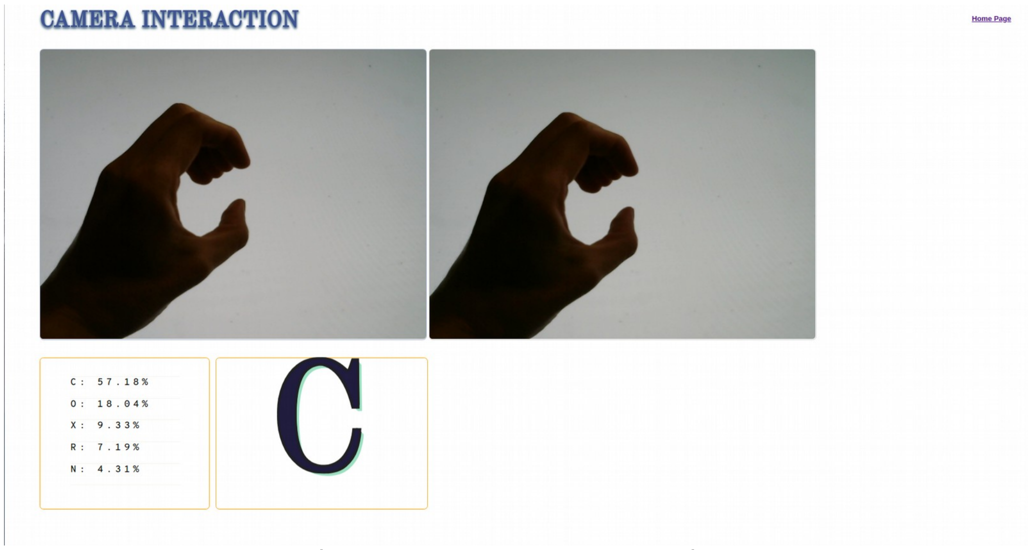


Figure 7.2-4 Camera Interaction

5. The next feature we will go over is the “training new data” feature. Once on the training page, we will be able to upload a dataset of images that include ASL images for training. To upload a dataset, **Select** the upload button provided on the page. Once all images have been uploaded, the newly added images will be a part of the dataset for training.

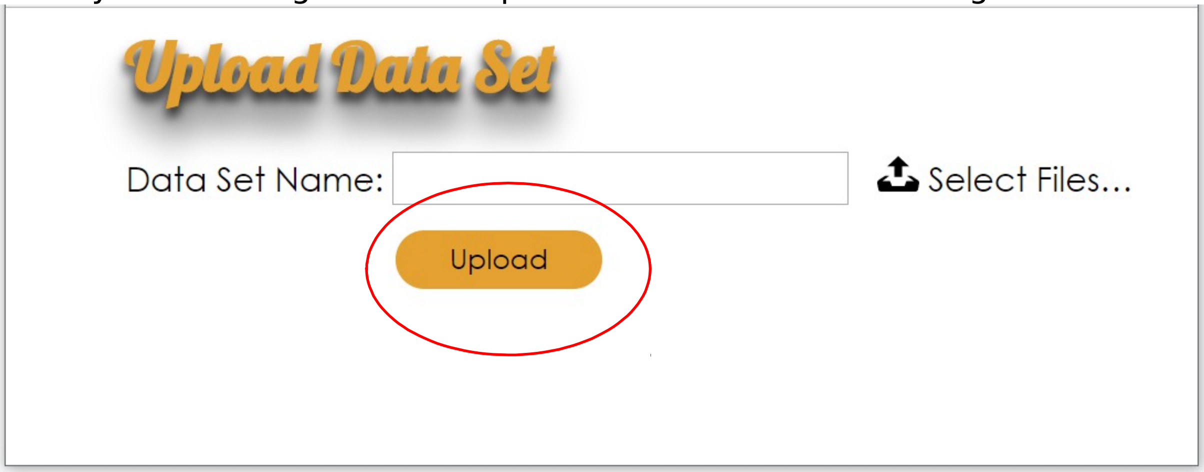


Figure 7.2-5 Training New Data

6. You now have the power to use our Sign Language Predictor!

Appendix A: Glossary

Acronyms	Definitions
ASL	American Sign Language
CPU	Central Processing Unit
GUI	Graphical User Interface
HOH	Hard of Hearing
OS	Operating System
RAM	Random Access Memory
SRS	Software Requirements Specification
UI	User Interface

Appendix B: Analysis Models

None