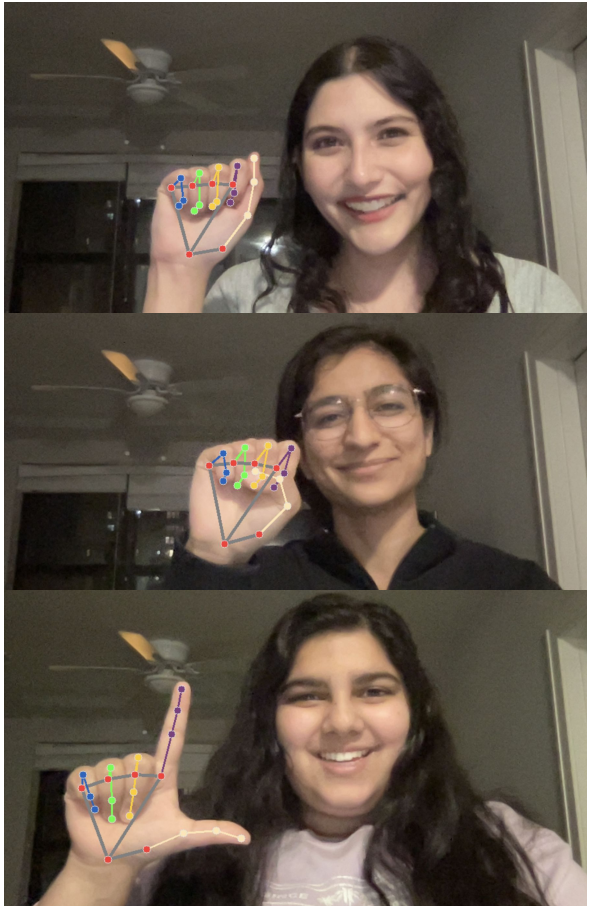
Team E5: ASLearn

Aishwarya Joshi, Hinna Hafiz, Valeria Salinas

Creating ASLearn: an ASL Learning Platform

- Facilitate ASL remote learning
- Users able to learn signs and test their knowledge
    - Signing into a camera
    - Getting immediate feedback on sign correctness
    - Flexibility in learning pace and schedule

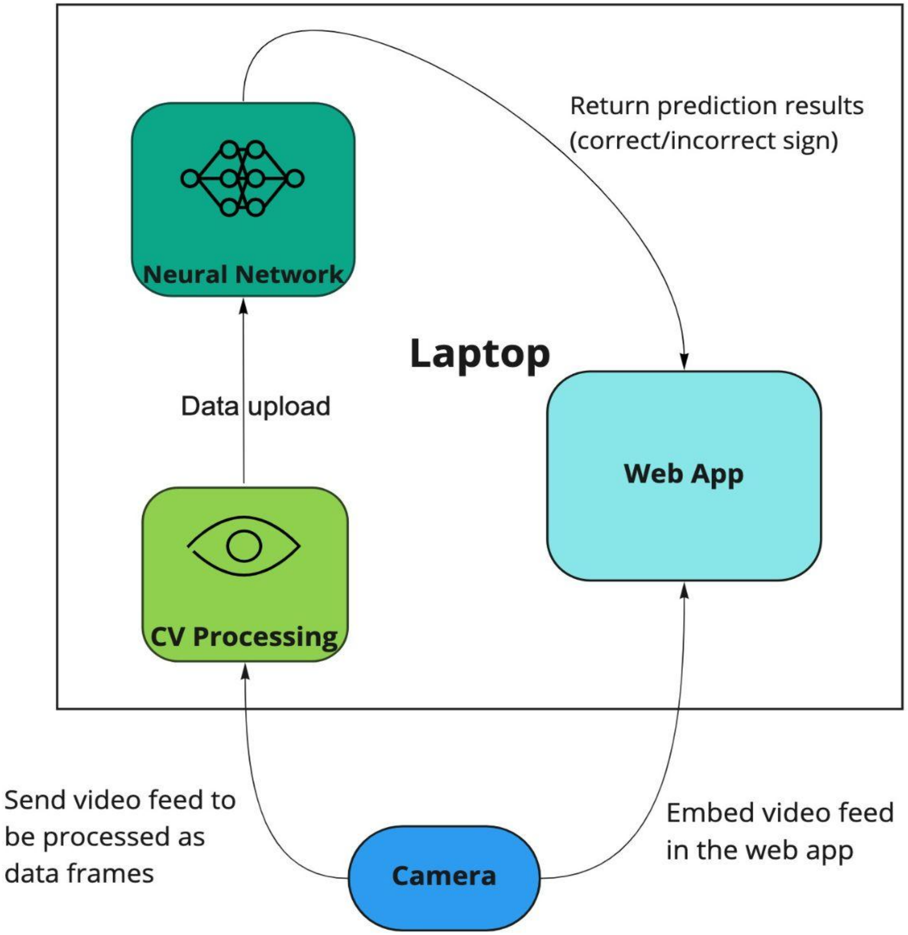


Use-Case Requirements

Requirement	**Metric**
Use computer vision to recognize when user is making ASL signs	**15** communicative signs + **26** alphabet letters + **10** numbers (0-9)
Model continues detecting user signs at a reasonable distance	Within **3 feet or less** facing camera head on
Correct detections occur in a timely manner	Within **2 seconds** of visual input being received
Accurate feedback on sign correctness	**90%** accuracy based on model evaluation with test data
Easy to navigate course page	**90%** user satisfaction based on user survey
Easy to understand feedback of signage	**90%** user satisfaction based on user survey

Solution Approach

- Video Input
- Computer Vision
- Machine Learning/Neural Network
- Web Application

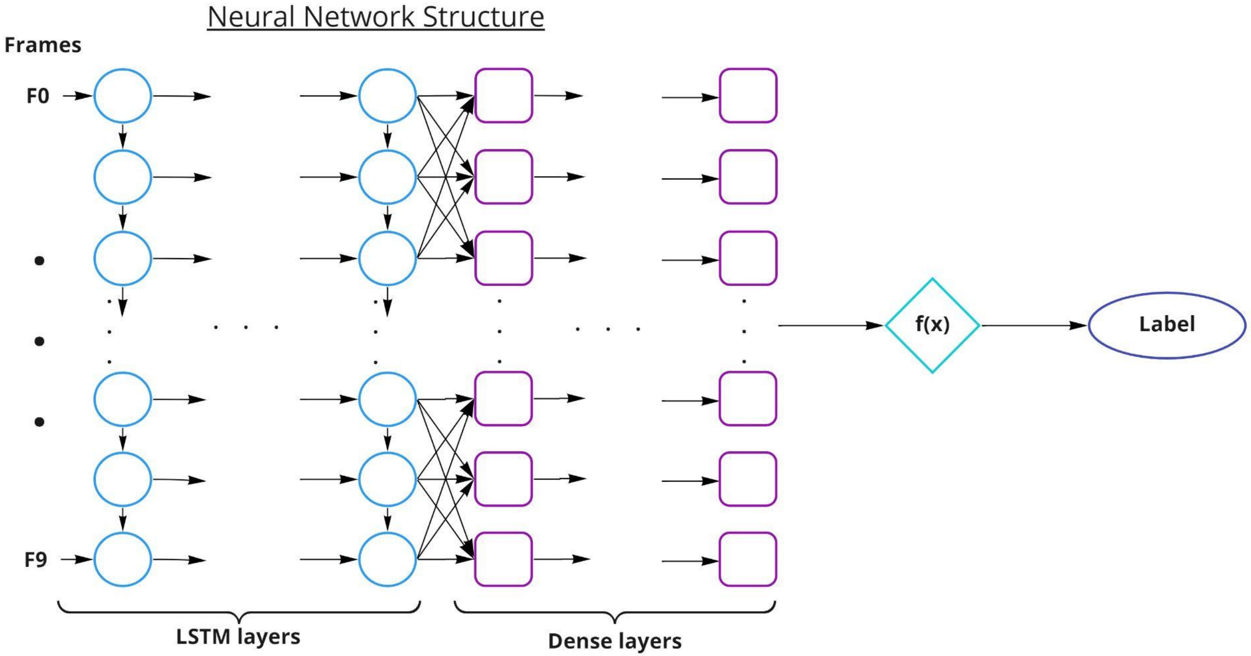


Return prediction results (correct/incorrect sign)
Neural Network
Laptop
Data upload
Web App
CV Processing
Send video feed to be processed as data frames
Camera
Embed video feed in the web app

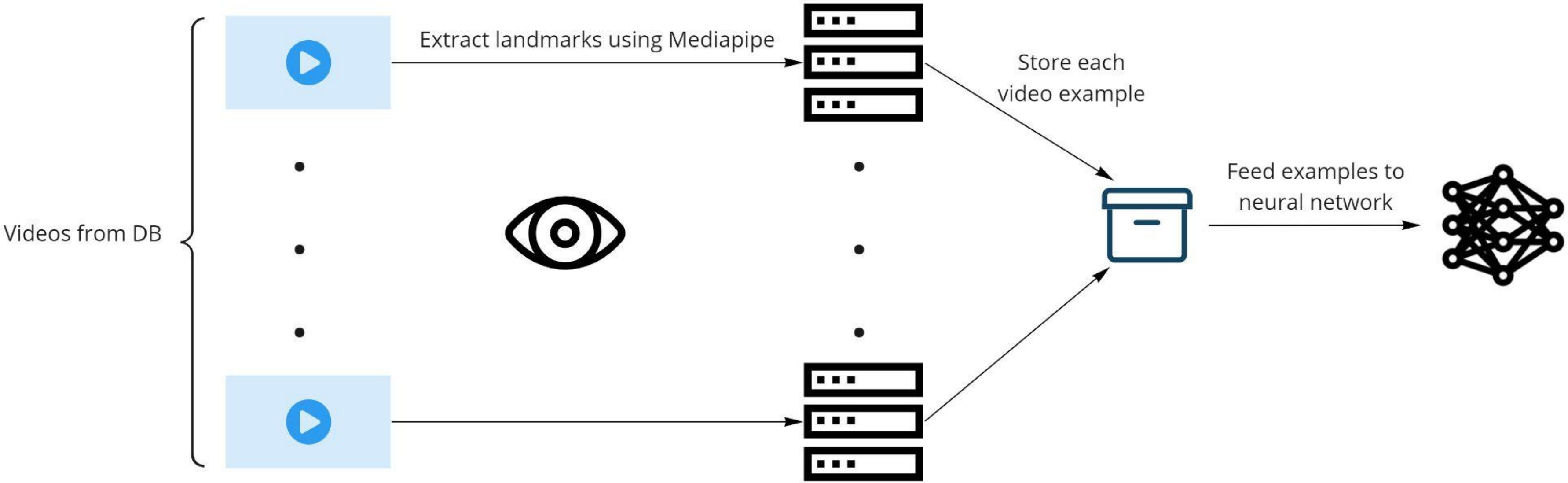
Neural Network

Final layer will output probabilities of each label being the detected gesture.

f(x) will apply argmax to compute the label with the highest probability



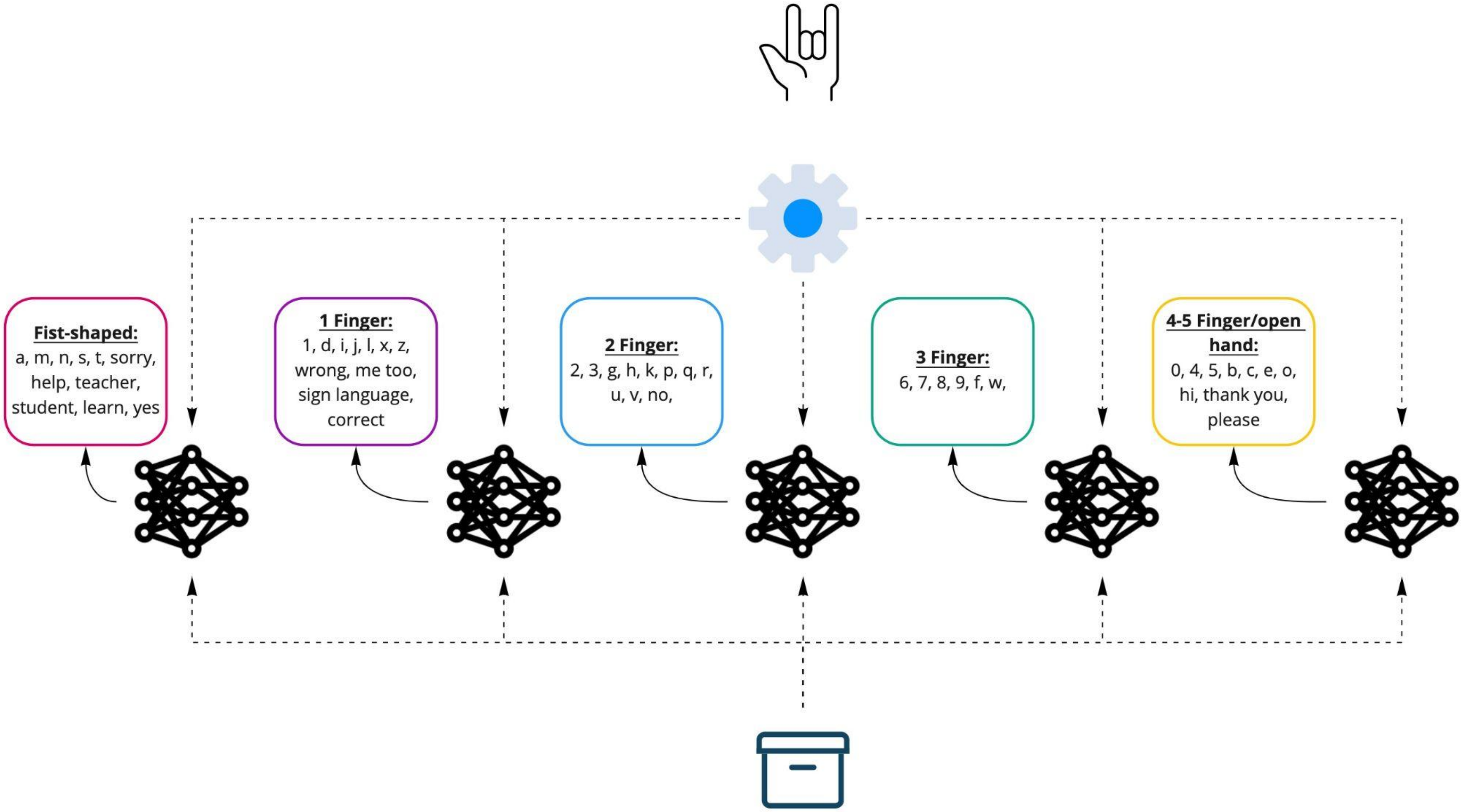
Model Training Pipeline


Extract landmarks using Mediapipe
Store each video example
Feed examples to neural network
Videos from DB

Training data sources:

[Mendeley](#) [WASL](#) [Letters j,z](#)
[Alphabet](#) [Numbers](#) [ASLLVD](#)

Neural Network as Microservice Architecture



Fist-shaped:
a, m, n, s, t, sorry,
help, teacher,
student, learn, yes

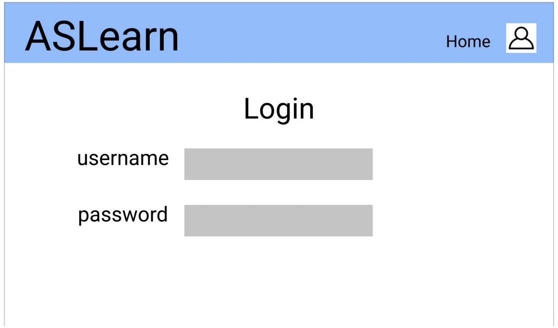
1 Finger:
1, d, i, j, l, x, z,
wrong, me too,
sign language,
correct

2 Finger:
2, 3, g, h, k, p, q, r,
u, v, no,

3 Finger:
6, 7, 8, 9, f, w,

4-5 Finger/open hand:
0, 4, 5, b, c, e, o,
hi, thank you,
please

Webapp UI Design

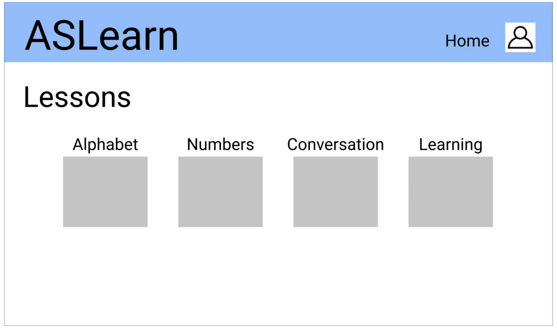


ASLearn Home

Login

username

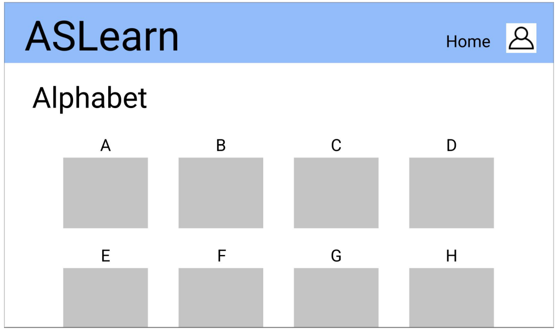
password



ASLearn Home

Lessons

Alphabet Numbers Conversation Learning

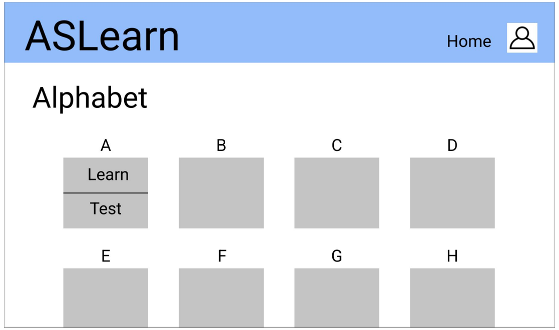


ASLearn Home

Alphabet

A B C D

E F G H



ASLearn Home

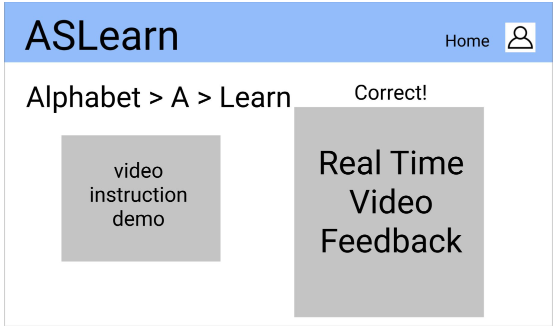
Alphabet

A B C D

Learn

Test

E F G H



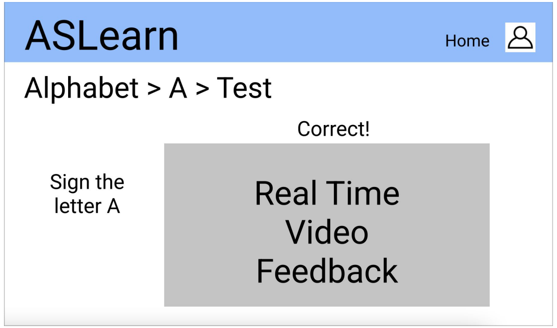
ASLearn Home

Alphabet > A > Learn

Correct!

video instruction demo

Real Time Video Feedback



ASLearn Home

Alphabet > A > Test

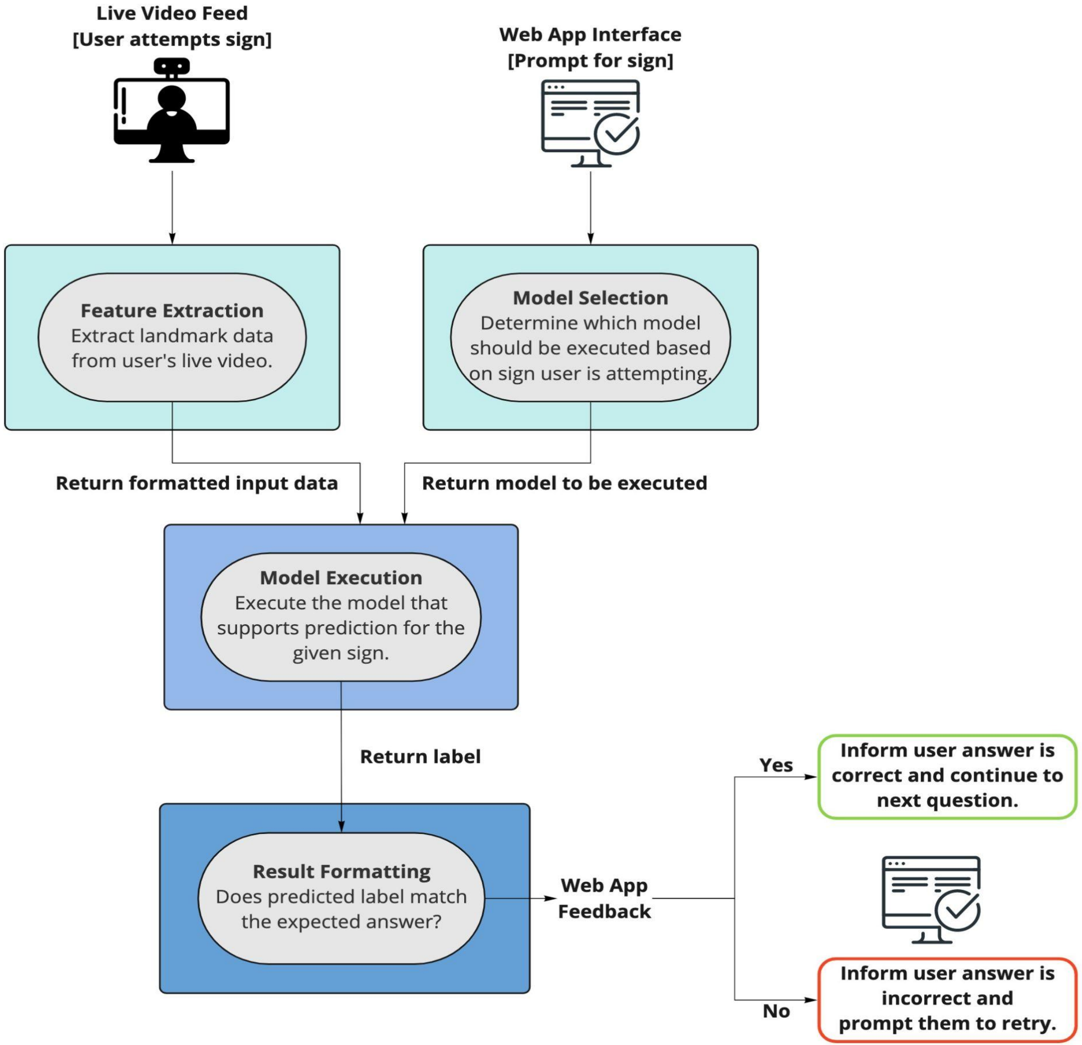
Correct!

Sign the letter A

Real Time Video Feedback

Responding to User Signs

This sequence of steps will determine correctness of user's answer and what feedback to display in the web app.


Live Video Feed
[User attempts sign]
Web App Interface
[Prompt for sign]
Feature Extraction
Extract landmark data from user's live video.
Model Selection
Determine which model should be executed based on sign user is attempting.
Return formatted input data
Return model to be executed
Model Execution
Execute the model that supports prediction for the given sign.
Return label
Result Formatting
Does predicted label match the expected answer?
Web App Feedback
Yes
Inform user answer is correct and continue to next question.
No
Inform user answer is incorrect and prompt them to retry.

Testing, Verification, Metrics

Requirement	**How We Will test**	**Passing Metrics**	**Risk Mitigation**
Distance from camera	Do signs at varying distances from camera (up to 3ft)	Sign recognition should be 90% accurate at **<3 feet**	Reduce distance requirement, use bounding box.
Platform latency	Have users sign in platform and record feedback time	Feedback will be given to user within **2 seconds**	Use alternative prediction generation method
Webapp User Interface	Conduct user testing and surveys (with at least 5 users)	**90%** overall user satisfaction based on survey	Modify UI according to user feedback

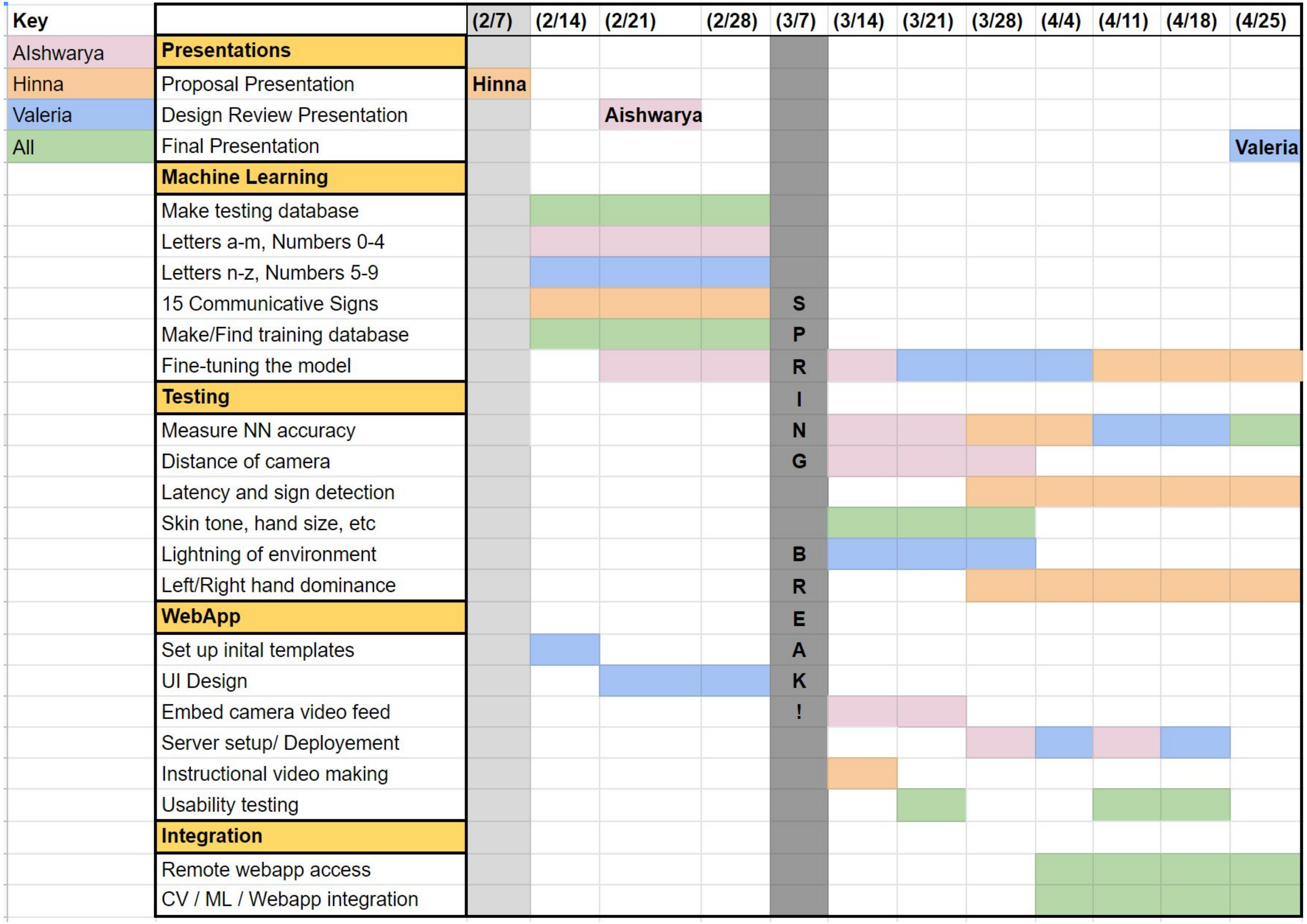
Testing, Verification, Metrics (cont)

Requirement	**How We Will test**	**Passing Metrics**	**Risk Mitigation**
Accuracy (i.e. jewelry)	Test with various user impediments (i.e users wearing bracelets, rings, etc)	Sign recognition should be **90% accurate** regardless of these impediments	Tell user not to wear jewelry, and remove other potential impediments
Left/Right hand dominance	Test platform with right and left handed users	Sign recognition should be **90% accurate** in both left / right handed cases	Tell user to make signs using only right hand (as that is our testing dominance currently)

Specifications

Initial Specifications (experimenting with toy data)	
Raw input video specs	5 seconds, 30 FPS
Sampling specs	10 frames, 2 FPS
Model execution time (1 LSTM Layer, 1 Dense Layer)	0.07-0.12 seconds
Tools	
MediaPipe (Hand, MediaSequence)	Extracting landmark data from video data
Tensorflow	Creating, training, and saving neural network
NumPy	Formatting data to feed into neural network
AWS	Training models and deploying web app within EC2 instances

Project Management: Schedule & Division of Labor



Key		(2/7)	(2/14)	(2/21)	(2/28)	(3/7)	(3/14)	(3/21)	(3/28)	(4/4)	(4/11)	(4/18)	(4/25)
🟪 AIshwarya	**Presentations**												
🟧 Hinna	Proposal Presentation	**Hinna**											
🟦 Valeria	Design Review Presentation			**Aishwarya**									
🟩 All	Final Presentation												**Valeria**
	Machine Learning												
	Make testing database		🟩	🟩	🟩								
	Letters a-m, Numbers 0-4		🟪	🟪	🟪								
	Letters n-z, Numbers 5-9		🟦	🟦	🟦								
	15 Communicative Signs		🟧	🟧	🟧	**S**							
	Make/Find training database		🟩	🟩	🟩	**P**							
	Fine-tuning the model			🟪	🟪	**R**	🟪	🟦	🟦	🟦	🟧	🟧	🟧
	Testing					**I**							
	Measure NN accuracy					**N**	🟪	🟪	🟧	🟧	🟦	🟦	🟩
	Distance of camera					**G**	🟪	🟪	🟪				
	Latency and sign detection								🟧	🟧	🟧	🟧	🟧
	Skin tone, hand size, etc						🟩	🟩	🟩				
	Lightning of environment					**B**	🟦	🟦	🟦				
	Left/Right hand dominance					**R**			🟧	🟧	🟧	🟧	🟧
	WebApp					**E**							
	Set up inital templates		🟦			**A**							
	UI Design			🟦	🟦	**K**							
	Embed camera video feed					**!**	🟪	🟪					
	Server setup/ Deployement								🟪	🟦	🟪	🟦	
	Instructional video making						🟧						
	Usability testing							🟩			🟩	🟩	
	Integration												
	Remote webapp access									🟩	🟩	🟩	🟩
	CV / ML / Webapp integration									🟩	🟩	🟩	🟩