

ML-Based ASL Alphabet Recognition

Ali Sayyed, Andrew Han, Marshall
Wurangian, Hao He, Matthew Ruffner,
Qinwen Zhou, Dian Zhao



Group Members



Marshall



Qinwen



Matthew



Andrew



Hao



Dian



Ali



Project Goals **01**

Approach **02**

TABLE OF CONTENTS

03 **Output**

04 **Future Steps**

05 **Conclusion**

01

Project goal





We Wanted to Create an ML-based Live ASL Recognition System



Why ASL? Background

- ~48 million Americans (15% of total population) have some degree of hearing loss
- Artificial Intelligence is increasingly becoming ubiquitous in all aspects of life
- Increase A.I. algorithm inclusiveness by having A.I. systems reflect the needs of the disabled
 - promote algorithms for people with disabilities, in particular people with hearing disabilities
- Our ML model
 - assist people with hearing loss
 - providing automated framework that transcribes ASL into written language
 - by recognizing alphabet letters from ASL hand signals



The background features a complex, abstract geometric pattern. It consists of numerous white lines of varying lengths and thicknesses, some forming solid polygons and others as loose, intersecting paths. Small white dots are scattered throughout, often serving as vertices for the lines. The overall effect is a sense of dynamic, interconnected structure.

02

Behind the Scenes

Dataset Overview

- The dataset consists of 87,000 images of American sign language
- 70% of the images are used for training and 30% for testing

Class: A



Class: B



Class: C



Class: D



Class: del



Class: E



Class: F



Class: G



Class: H



Class: I



Class: J



Class: K



Class: L



Class: M



Class: N

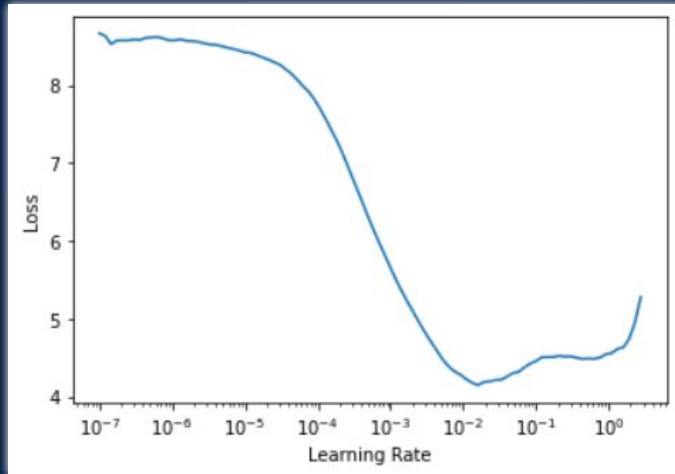


Class: nothing



The Process

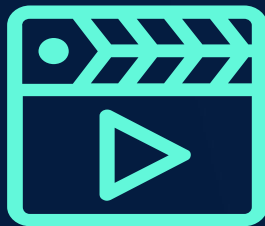
- We used a pretrained ResNet 50 model for training
- We then used the `learn.lr_find` function from the FastAI library to find the best learning rate
- Our final model achieves an accuracy rate of almost 100%



epoch	train_loss	valid_loss	error_rate	accuracy	time
0	0.077404	0.171081	0.046284	0.953716	2:39:22
1	0.001674	0.000877	0.000229	0.999771	13:49
2	0.000082	0.000242	0.000115	0.999885	13:49

Real-Time Prediction System

- OpenCV - a Python library aimed at real-time computer vision tasks
- Handles tasks including face recognition, object detection, etc.



Video



Frames



Prediction

03

Output



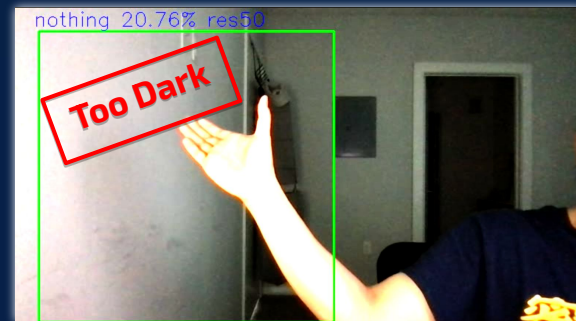
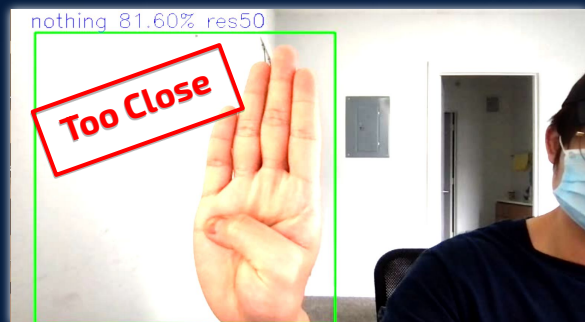


Demonstration

**INSERT DEMO
CLIPS HERE**

Model Shortcomings

- Distance
- Lighting





04

Future Steps




How Can We Make It Better?

Utilize a Larger Training Set

Create a more diverse training image set, allowing for more accurate outputs (mirrored images for left-handed people)

Evolving the Model to Recognize Words

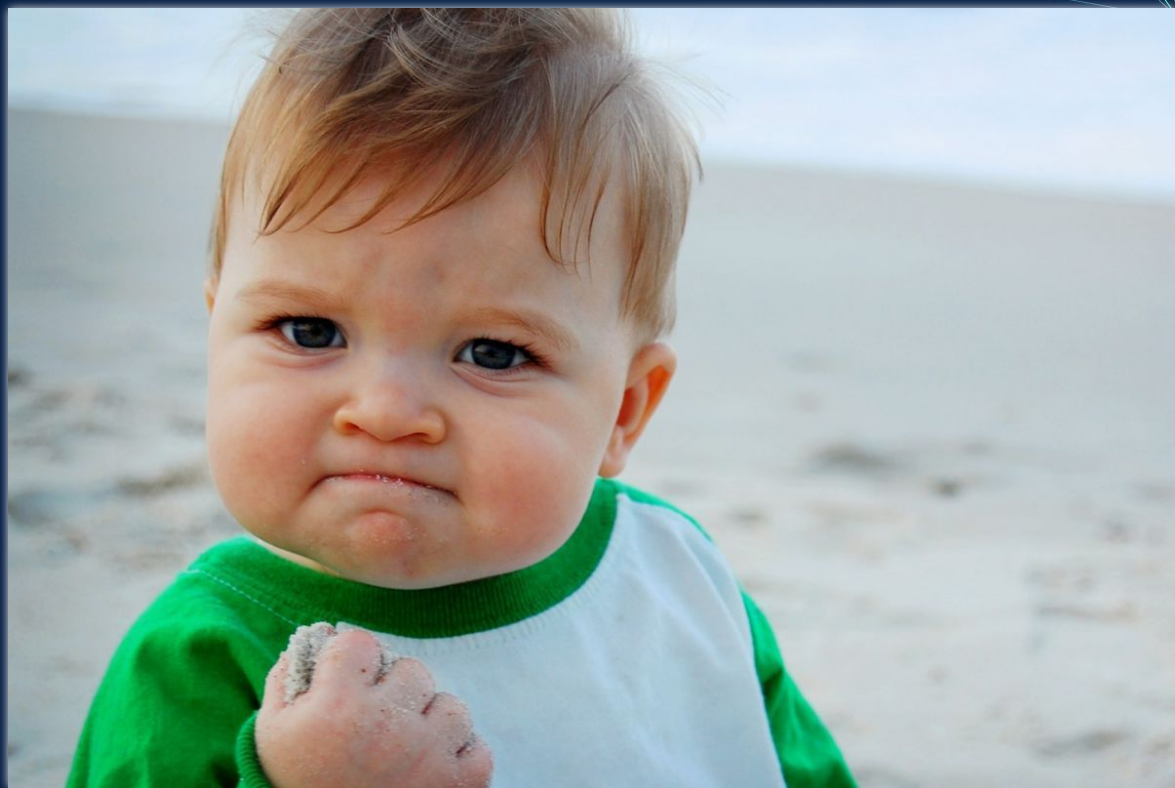
Recognize words and their associated sentiments, allowing us to blend both the ideas of CNN and NLP into the project.





05

Conclusion



Recap

01

Goal

02

Approach

03

Output

04

Future Steps

05

Conclusions



THANKS!

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.

Please keep this slide for attribution.