Hand Gesture
CLASSIFICATION

Ethan Gruening and Owen Harty

TABLE OF CONTENTS

01

BACKGROUND

Background

Our world revolves around constant communication and clarity.

Many translators exist for written and spoken communication.

Communication by signing and hand gestures is an emerging product in computer vision.

Data Availability

To classify a hand gesture from an image, a ML model will require a large dataset.

Gathering our sample and testing data requires extracting features from an image.

The HaGRID datasets provide classified images for testing, training, and validation.

Our Problem

Hand gesture classification is minimal for ASL translators.

We wanted to build a model to optimally classify a variety of hand gestures.

We will use different machine learning techniques to find the most optimal model.

02

DATASETS

HaGRIDv2 Dataset

- Large dataset of hand gestures
- Can extract 21 landmarks from each gesture as features.
- Output will be the class name
    - Ex. “fist” or “thumbs up”

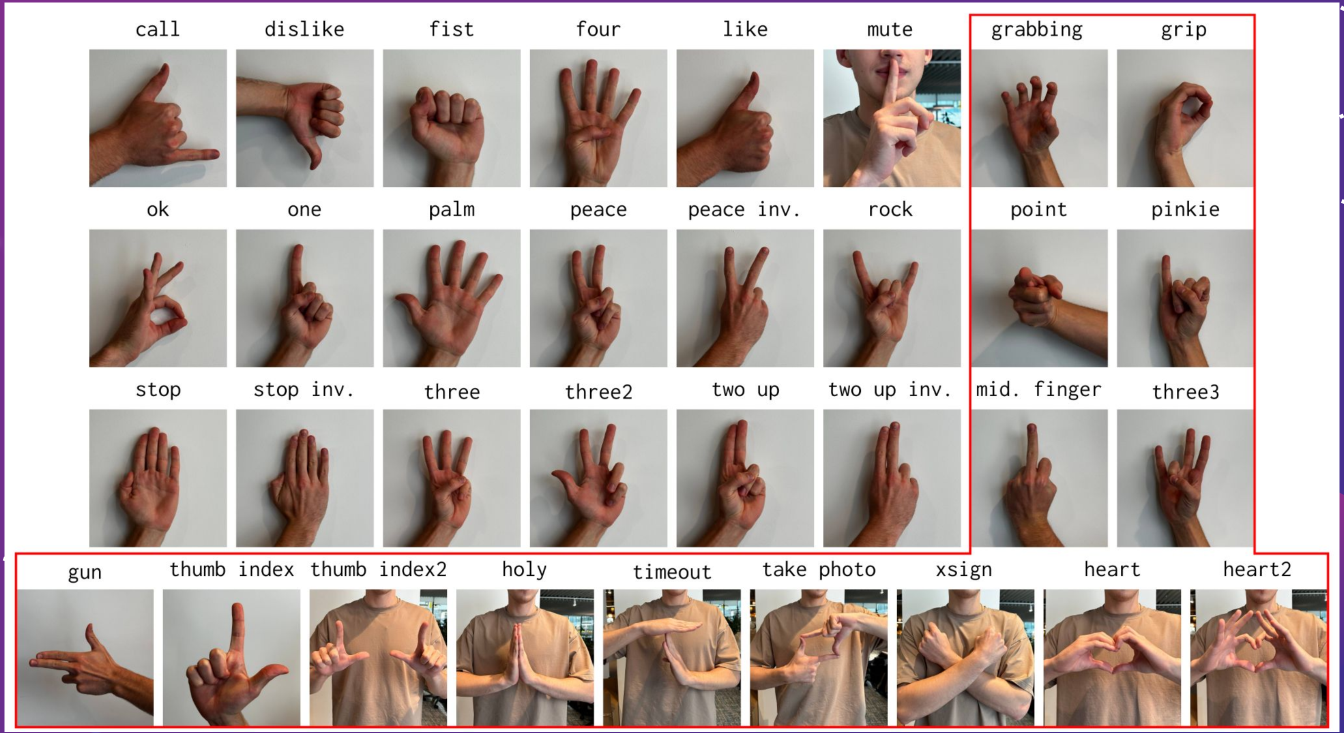
	Image Count	Percent
Training	410,800 Images	74%
Validation	54,000 Images	10%
Testing	90,000 Images	16%

37,583
Unique People (18-65 years old)

33
Unique Classes

723 GB
Image files

33 Gesture Classes



03 RELATED WORK

DeepASL

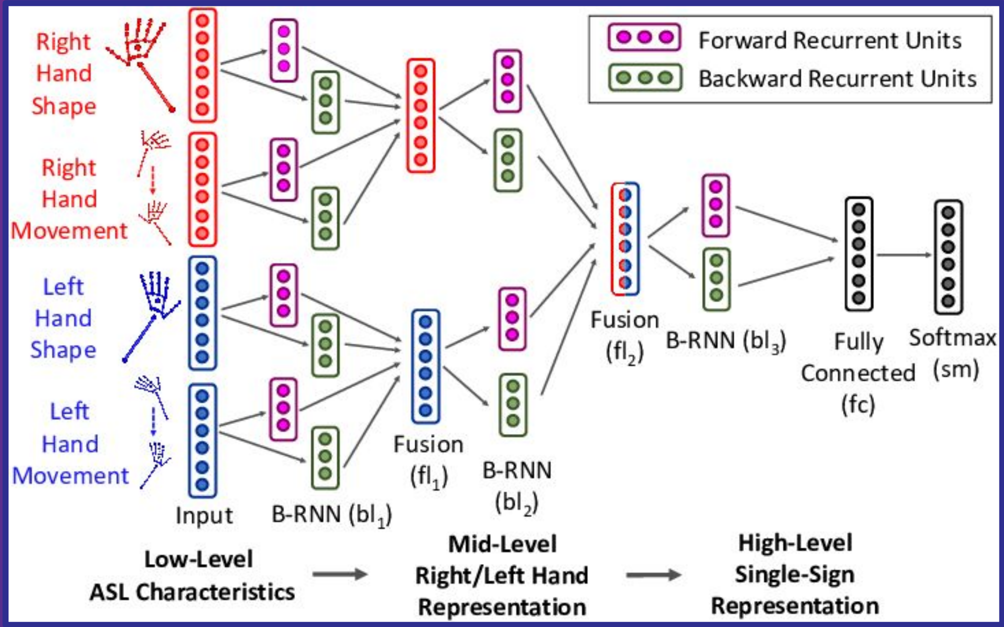
DeepASL, a system to translate ASL into text or speech, uses a novel hierarchical bidirectional deep recurrent neural network (HB-RNN) to classify and translate hand signs.

ASL signs depend on both past and future context. Bidirectional LSTMs ensure full contextual understanding.

How it Works:

1. Input Features
2. Bidirectional RNN (B-RNN) Processing
3. Feature Fusion
4. High-Level Single-Sign Representation

While CNN and SVMs are great for isolated gestures, HB-RNN can improve performance for dynamic sequences.



04

METHOD

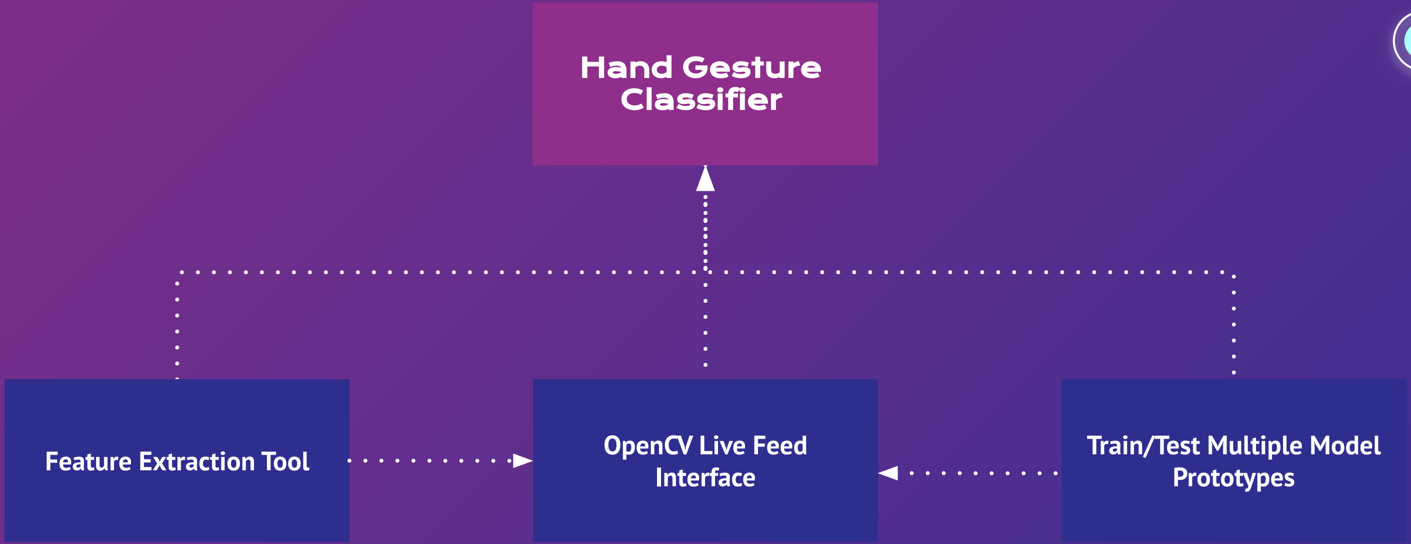
Intuition

Our system leverages MediaPipe's real-time hand tracking to capture 3D landmark coordinates, transforming raw gesture data into machine-interpretable features.

We recognized that different model architectures would excel at extracting different patterns:

- 2D CNNs
  - Spatial relations in 2D projections
- 1D CNNs
  - Temporal Sequences
- SVMs
  - Handcrafted features

Finding the optimal live classifier



Feature Extraction - Training

STEP 1

Parse image using
Google's Mediapipe

STEP 2

Map 21 hand landmarks
into (x,y) coordinates

STEP 3

Flatten coordinates into
a 42-feature vector

STEP 4

Input as training values

Live Predictions

STEP 5

Parse hand landmarks
from a live feed:
Steps 1, 2, 3

STEP 6

Enter as sample data:
predict(landmark_vector)

STEP 7

predict() outputs the
class and is displayed

Algorithm Details

- The live test subsystem demonstrates real-time capability, processing both hands independently at ~20 FPS while displaying classification results and confidence metrics.

- Our evaluation protocol uses strict train-test splits from HaGRID to ensure realistic performance measurements across all/six gesture classes.

- Additional innovations include dynamic architecture scaling; CNNs adjusting their size and structure automatically based on the task (simpler tasks can shift to fewer layers and more complex tasks can shift to extra layers).

05

EXPERIMENT

Model Prototypes

Model Type	Description
SVM	A scikit-learn Linear Support Vector Configuration (LinearSVC) - Default parameters | L2 Regularization | Linear Kernel
1D CNN	A 1D, sequential convolutional neural network - 10 epochs | ReLU Activation | .001 Learning Rate
2D CNN	An adaptive, 2D convolutional neural network - 10 epochs | ReLU Activation | Adam Optimizer
DenseNet	A 1D convolutional neural network with concatenated Dense Blocks. - 10 epochs | ReLU Activation | .001 Learning Rate
DenseNet2D	A 2D convolutional neural network with concatenated Dense Blocks. - 10 epochs | ReLU Activation | .001 Learning Rate
TabNet	A deep learning architecture to make sequential decisions and a feature transformer. - 100 max epochs | Batch Size 1024

Features to Use:

- Primary Features
  - 2D (x,y) coordinates from 21 hand landmarks (42 dimensions total)
- Normalization
  - Automatic coordinate normalization via MediaPipe (0 - 1)

Hyper-parameter Choices:

- number of filters
- kernel size
- dropout
- dynamic pooling

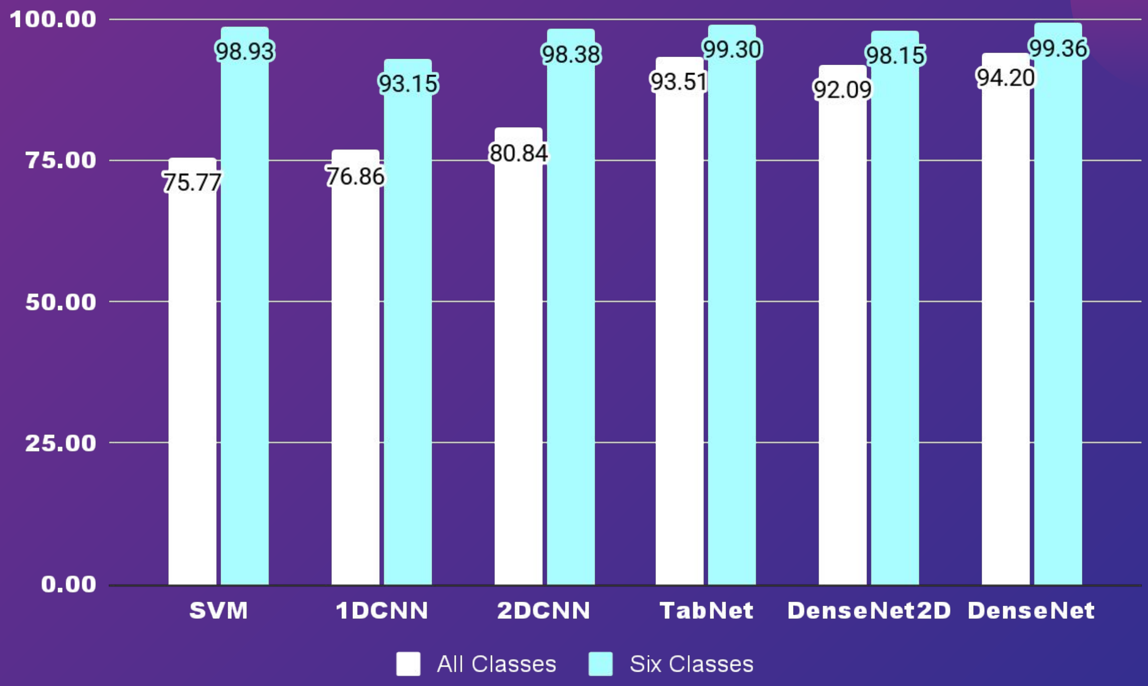
Accuracy Results

All Gestures

33 Classes

6 Gestures

6 Classes:
- OK
- Thumbs up
- Thumbs down
- Peace sign
- Fist
- Palm



Analysis

- For all classes, TabNet and DenseNet dominated with 93.51% and 94.20% accuracy respectively.
- For the six classes, all of the models were able to achieve >93% accuracy.
- Unexpectedly, SVM had a very strong 6-class performance, on par with DenseNet. TabNet also had surprising competitiveness with DenseNet.

Model Type	All-Class Challenges	Six-Class Strengths
SVM	Struggles with high dimensionality	Excellent linear separability
1D CNN	Limited temporal feature learning	Adequate for basic gestures
2D CNN	Better spatial generalization	Overengineered for simple cases
Dense Net	Optimal feature reuse	Slight overfitting tendency
TabNet	Effective feature selection	Maintains interpretability

06

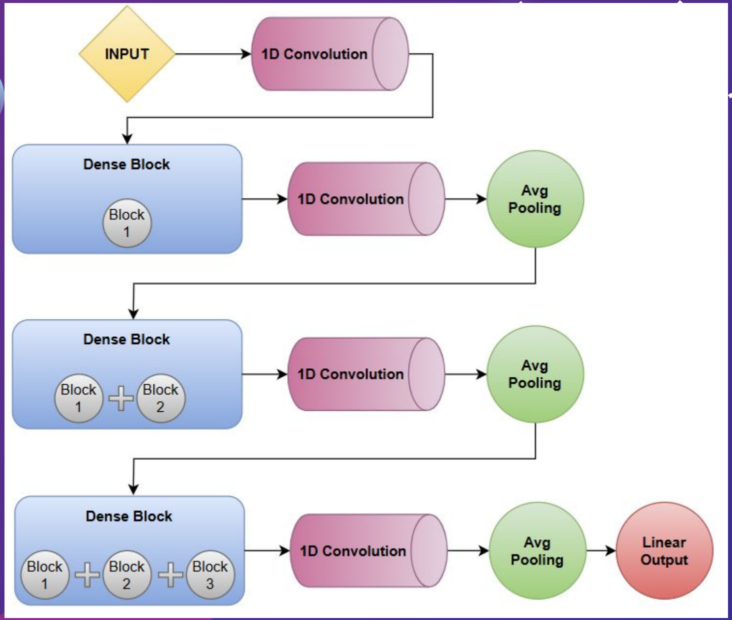
CONCLUSION

Our Best Model: DenseNet

- 94.2% accuracy on 33 gesture classes
- 99.36% accuracy on 6 gesture classes
- Ran with 100 epochs with 95.63% accuracy

Future Work:
- Develop an adaptive system that routes simple gestures to SVM and complex cases to DenseNet, optimizing both speed and performance.

- Integrate transformer layers into the 1DCNN to better capture long-range temporal dependencies in hand movements, improving sequential gesture recognition.



Demo Video Placed Here