Sign Language Detection and Classification

Varun Mohan | University of Chicago | MPCS

Most of us communicate verbally every single day without truly thinking about it. But what if that wasn’t an option?

This is the reality for the deaf community.

People with hearing impairments have no choice but to defer to their other senses, and as a result, there is a gap in communication with those who can hear normally.

10,000,000
People in the U.S. are hard of hearing,

1,000,000
Are functionally deaf,

500,000
Americans use ASL alone.

1

Bridging the Gap

What can we do help communication with those who rely on ASL?

What is ASL?

- American Sign Language (ASL) is a natural language expressed through hand gestures, allowing communication to be entirely visual rather than audial
- It is the predominant sign language used by the U.S.'s deaf community

Can We Use Machine Learning to Classify Sign Language?

- What kind of data do we need, and how much of it can we get?
- What is a reasonable scope for a project to answer this question?
- Assuming a successful model, what would future steps look like?

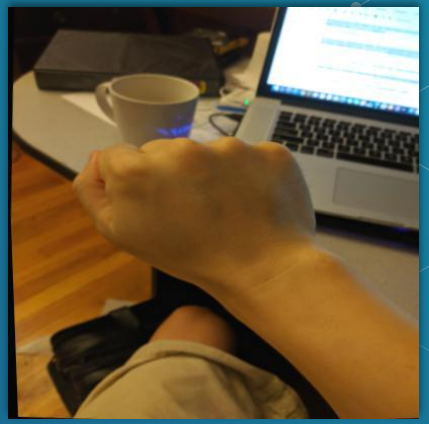
2

Data

We need image data to train a model

Roboflow: ASL Object Detection Dataset

- **1728** images in **26** classes for each ASL letter
- **1512** training set, **144** validation set, and **72** testing
- Multiple images per class, minor augmentations to brightness and horizontal flip



3

Modeling

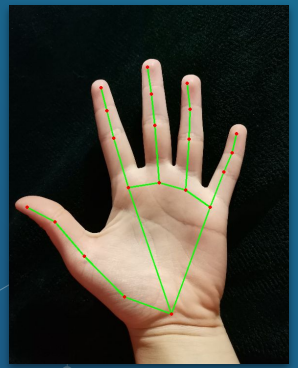
Object Detection and CNN Classification

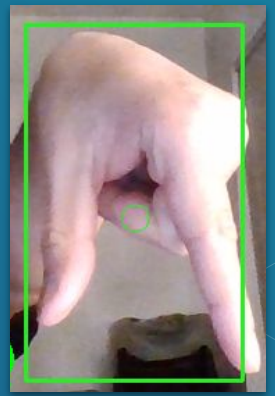
A Two-Step Process

Hand Detector + **Neural Network Classifier**

Hand Detection: MediaPipe

- **MediaPipe** is a suite of ML models by Google, one of which specializes in hand landmark detection
- Trained on over 30,000 data points

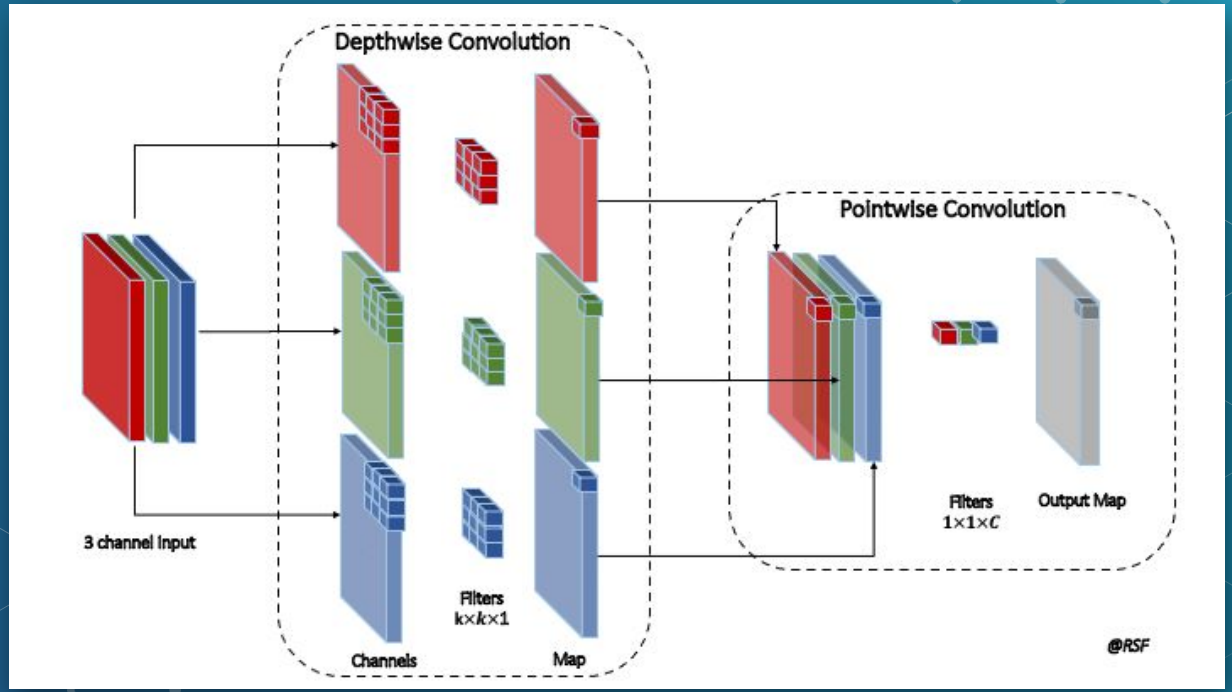




Classification: MobileNetV2

- **MobileNet** is a Convolutional Neural Network that specialized for being **lightweight** while maintaining strong predictive accuracy
- It accomplishes this using **depthwise separable convolutions**

MobileNetV2 Architecture



4 Results

How did we do?

Accuracy and Precision

	Accuracy	Precision
Train	0.93	0.94
Validation	0.83	0.88
Test	0.75	0.75

Demo

Let’s see how it works in real-time!

5

Conclusions

Key takeaways and future work

Key Takeaways

Hand Positions Matter

Letters with similar hand positions (i.e. close- fisted) were misclassified more often than more distinctive letters

More Data, Varied Data

Performance was respectably strong on a sparse dataset, but more data + image variation = a better model

What's Next?

Words and Phrases

The future lies in extending models like this into a complete language model for ASL to facilitate true translation.

Real Applications

In any capacity, applications like this can help non-ASL speakers communicate with ASL-speakers, and can be used as learning tools.

THANKS!

ANY QUESTIONS?

5 Appendix

Custom Head Model

```
mobilenet = MobileNetV2(input_shape=IMG_SHAPE, include_top=False, weights='imagenet')

handModel = mobilenet.output
handModel = GlobalAveragePooling2D()(handModel)

handModel = Flatten(name="flatten")(handModel)

handModel = Dense(128, activation="relu")(handModel)
handModel = Dense(256, activation = 'relu')(handModel)
handModel = Dense(512, activation = 'relu')(handModel)

handModel = Dropout(0.5)(handModel)

handModel = Dense(26, activation="softmax")(handModel)
```

Optimization

```
# compile our model
opt = Adam(learning_rate=0.001, weight_decay=0.1)

model.compile(loss="categorical_crossentropy", optimizer=opt,
    metrics=["accuracy", Precision()])

# train the head of the network
history = model.fit(
   aug.flow(train_data, train_labels, batch_size=12), epochs=30,
    validation_data=(val_data, val_labels),
    validation_steps=144/12,
)
```