Sign Language Learner

Group 8
Brennen Hill (Undergraduate)
Joseph Mostika (Undergraduate)
Mehul Maheshwari (Undergraduate)

Problem description

Our project seeks to raise current benchmarks of ASL image classification using concepts of transfer learning and pre-trained models touched upon in class.

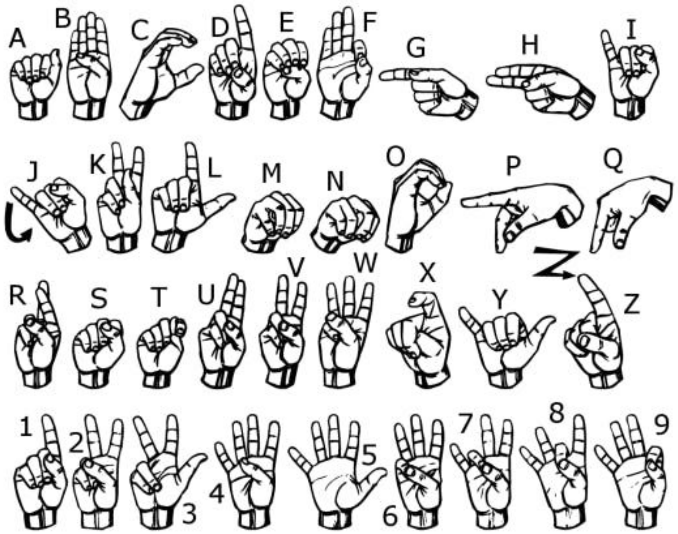
Current benchmarks sit at around 95-97% accuracy for image recognition - our goal was to improve these results while maintaining optimal efficiency

Tasks

1. Find a dataset that provides sufficient sample of standardized ASL images
2. Produce new model(s) that are able to achieve *at least* the benchmark
3. Produce an enhanced model that achieves 98%+ accuracy, landing on a finalized architecture that surpasses current sequential models

Related work: what has been done, state of the art/baseline result

- Communication is essential for people around the world to work and share ideas with one another.
- The methods used largely depend on the way the data is collected
- Most existing models take in preprocessed images that show only a hand representing some sign
  - The majority use convolutional neural networks
- To a lesser extent, work is also being done to convert videos to text (Li)
- One of the models we analyzed was especially robust (sachinpatil1280)
  - It reached 97% accuracy on a database
  - We decided to work towards improving the existing model to exceed that accuracy

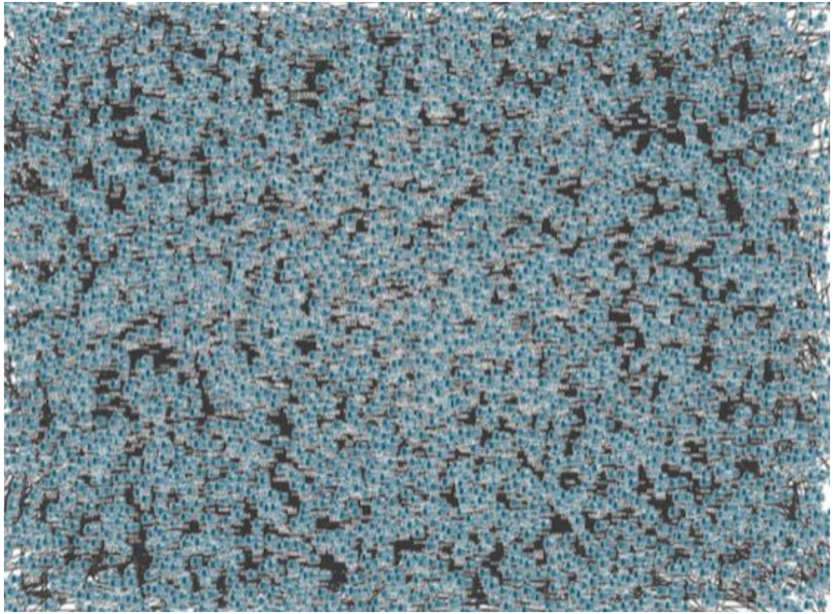


Dataset: Source, size, shape, partitioning, pre-processing performed, example of data sample

- The source of our database was a Sign Language MNIST on Kaggle (Tecperson)
- The data set is a collection of images of alphabets from the American Sign Language, separated into 24 classes.
- Training data (27,455 cases) and test data (7172 cases),
- Each image was formatted 300x300, with grayscale values between 0-255
- Images were taken of only the hand of someone forming a sign
- Data Preprocessing
    - We partitioned data into 80% training, 10% validation, and 10% testing
    - We applied data augmentation

Model Version A

- Implemented transfer learning with resnet
- Added pooling layers and large fully connected layers on top of resnet in an attempt to get a strong fit for the data
- There was an exceptionally large number of trainable parameters
- It was taking an unreasonable amount of time to train with available computing resources
- The architecture was scrapped and we moved to computationally cheaper options



Model Version B - ResNet50 w/ 30 epochs (98.5% final accuracy)

Unique aspect: First 10 epochs ran with **base layers frozen**, then another 20 ran with unfrozen base layers (for fine-tuning purposes)

Added an extra Dense layer with 512 neurons (relu activation) and softmax output

Ran with 2x augmentation parameters compared to benchmark model

Accuracy after 10 epochs with no fine-tuning: 87%

Accuracy after 30 total epochs with fine-tuning: 98.5%

a. Experiment with different rotation, scaling, zoom factors to leverage data augmentation capabilities
b. Reduce image size to increase efficiency while maintaining accuracy

Training time: 6-7 hours

Model C - Enhanced ResNet50, 10 epochs

Uses transfer learning with ResNet50 model

Added Dense Layer with softmax activation for output

```
# Add additional layers
x = Flatten()(base_model.output)
x = Dense(128, activation='relu')(x)
output = Dense(num_classes, activation='softmax')(x)
```

ResNet50 is fine-tuned while the output layer is trained from scratch

Ran with various different data augmentations

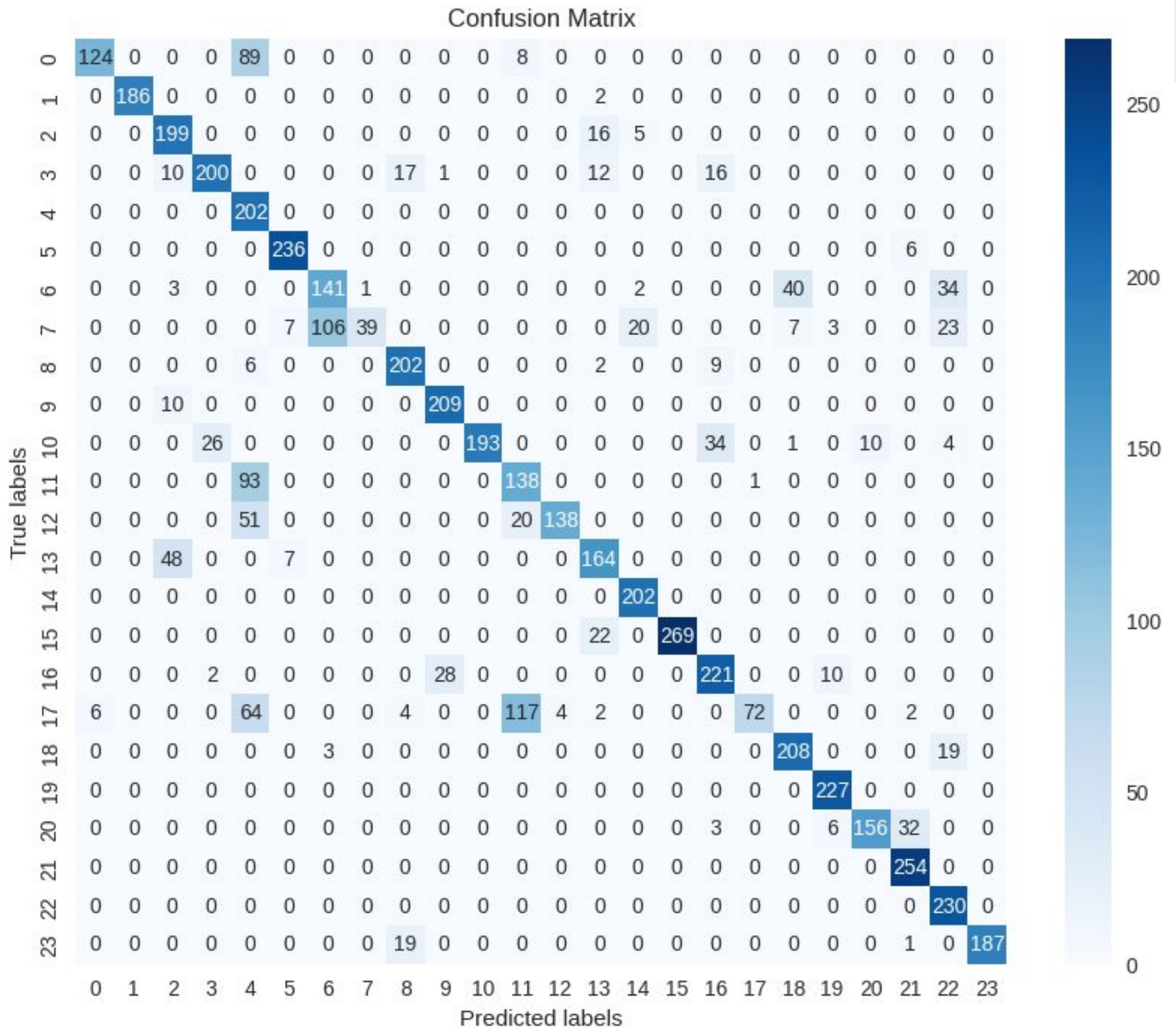
Use grayscale image set in the form of CSV to drastically improve efficiency and reduce learning time

Training time: 3 hours

f) Result / Analysis

Baseline Model: 97% Accuracy

Our Model: ~98.6% Accuracy



References

[1] Wolmark, M. (2023, September 2). 79 hearing loss statistics: How many deaf people in the U.S.?. In-Home & Center-Based ABA - Golden Steps ABATM. https://www.goldenstepsaba.com/resources/hearing-loss-statistics

[2] Li, D. (n.d.). Welcome to WLASL homepage. WLASL. https://dxli94.github.io/WLASL/

[3] CSC321 tutorial 6: Optimization and Convolutional Neural networks. tut06. (n.d.). https://www.cs.toronto.edu/~lczhang/321/tut/tut06.html

[4] Tecperson. (2017, October 20). Sign language mnist. Kaggle. https://www.kaggle.com/datasets/datamunge/sign-language-mnist

[5] sachinpatil1280. (2023, September 23). Hand-sign✊🖖: Multi-class classification CNN (97%). Kaggle. https://www.kaggle.com/code/sachinpatil1280/hand-sign-multi-class-classification-cnn-97