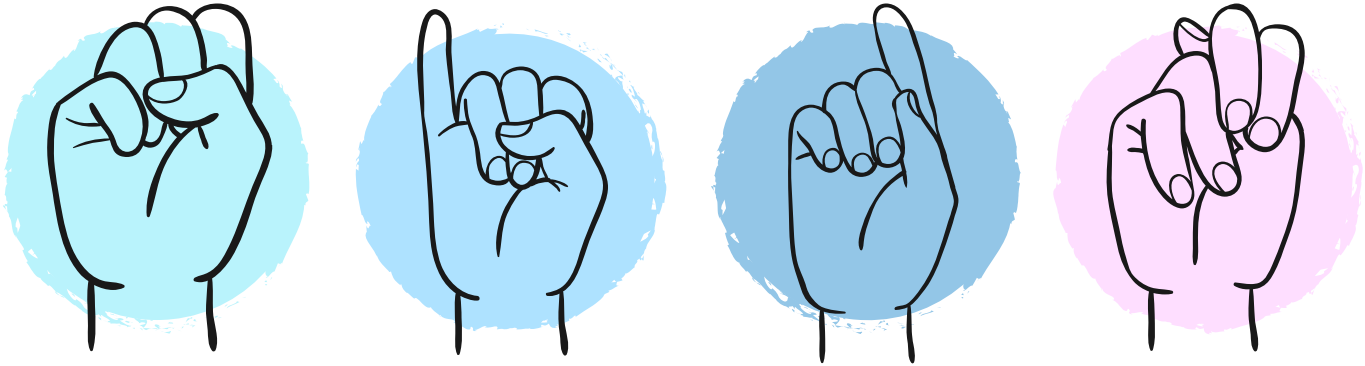
Sign Language Recognition

Skylar, Rainie, Mukhil, Edward, Divik, Derek,
Ainsley, and Aashna



Presentation Overview

01 DataSet BackgrouND

02 Data preprocessing

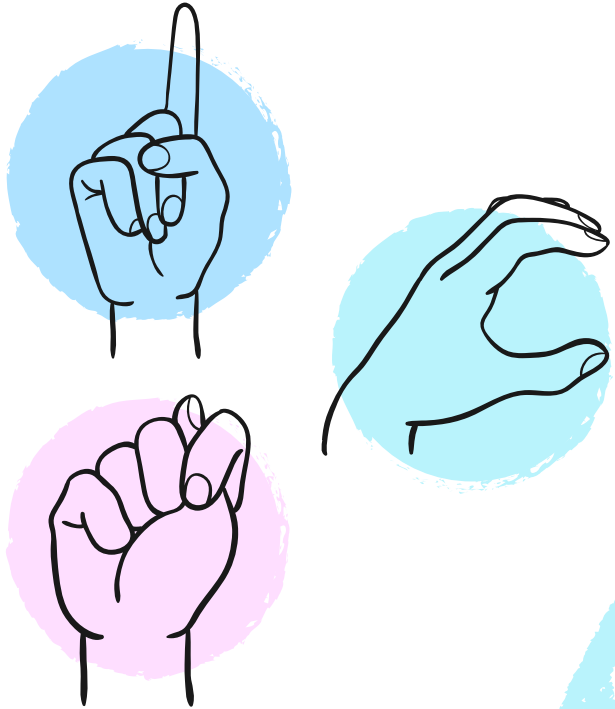
03 MODEL BUILDING

04 MODEL EVALUATION

05 FUTURE IDEAS

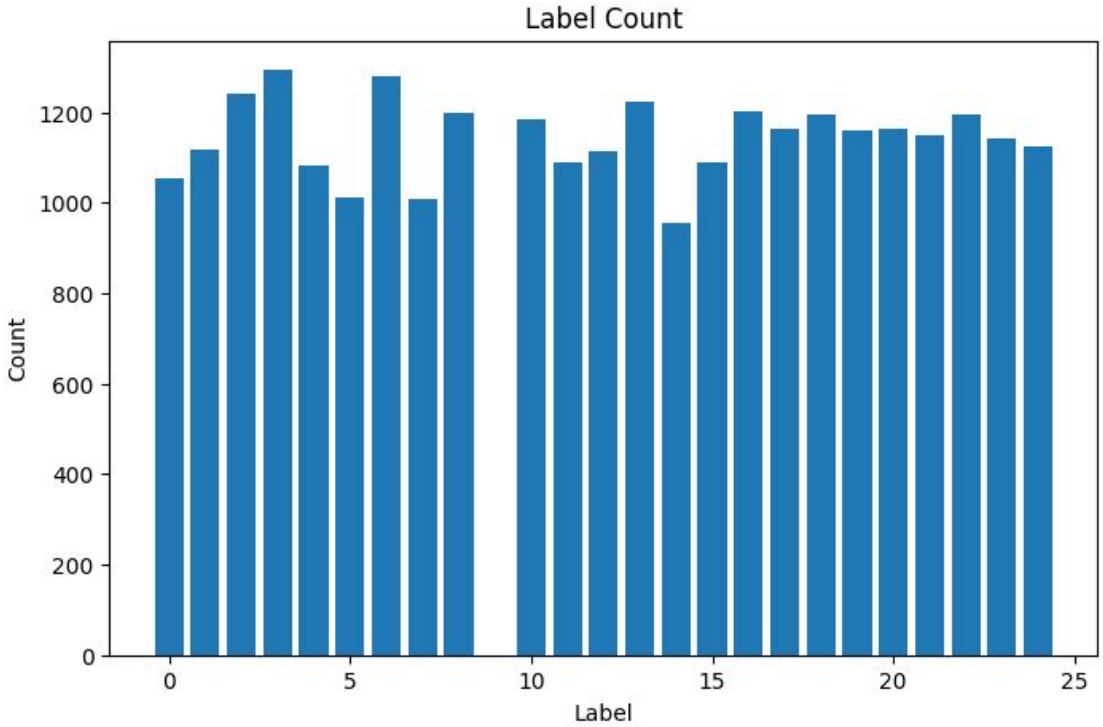
DATASET BACKGROUND

- How was the dataset presented?
  - Training (~28k) & Testing (~7k)
  - Pixels 1 to 784 correspond to 28x28
  - Grayscale = 0-255
- How was the dataset created?
  - Original Pictures (1704)
  - Image pipeline (ImageMagick)



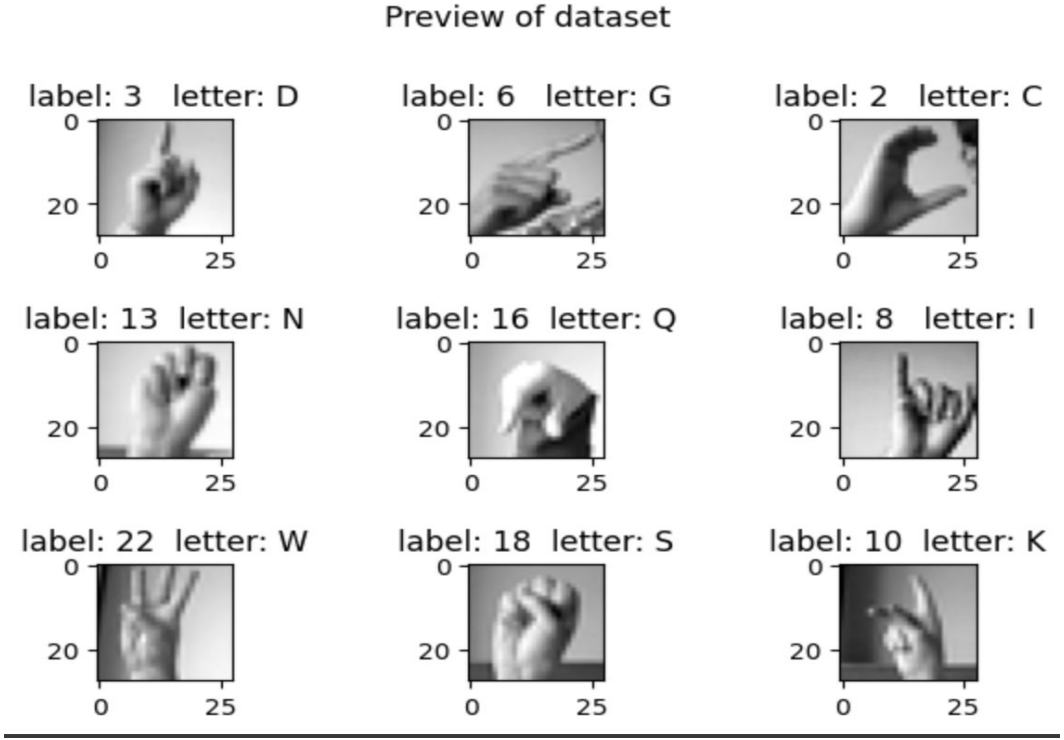
Dataset Source

Data Visualization



Note: Missing J and Z

Data Visualization



GOALS

1. Build models that classify images of sign language letters to their letter labels based off of pixel grayscale values.

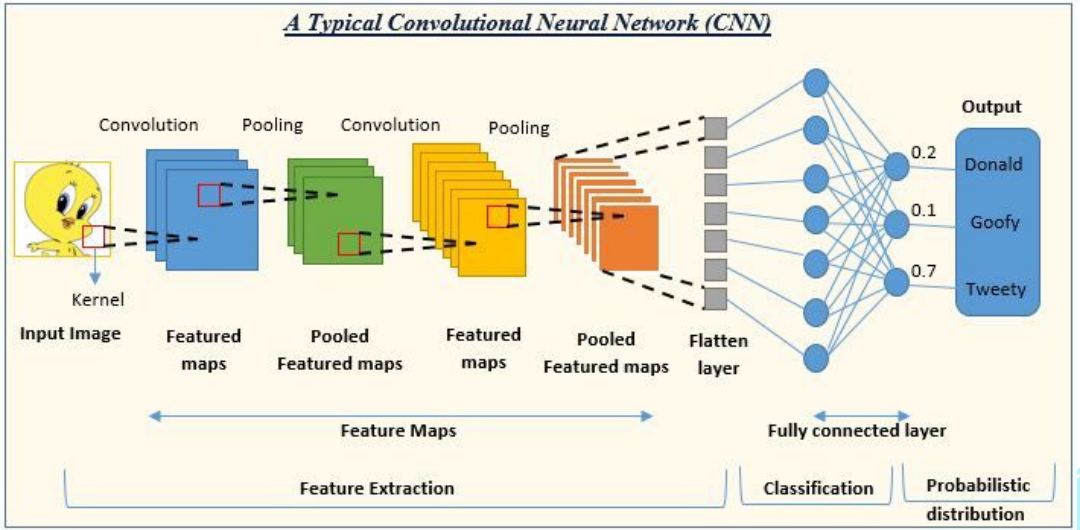
2. Find a model that has the best performance on unseen testing data.

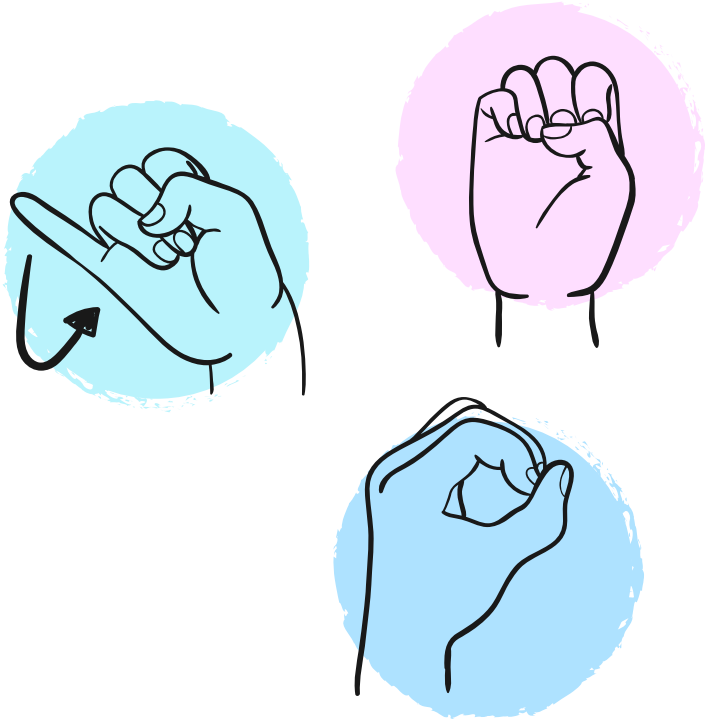
DATASET PREPROCESSING

- Label Binarizer
- Normalizing & Reshaping the Data
- Image Data Generator & Data Augmentation
- Split train and test into x and y

What are CNNs?

- Deep learning for image processing
- Main components:
  - Convolution layers
  - ReLu Function
  - Pooling
  - Classification
- Modern uses
  - Facial recognition
  - Autonomous vehicles





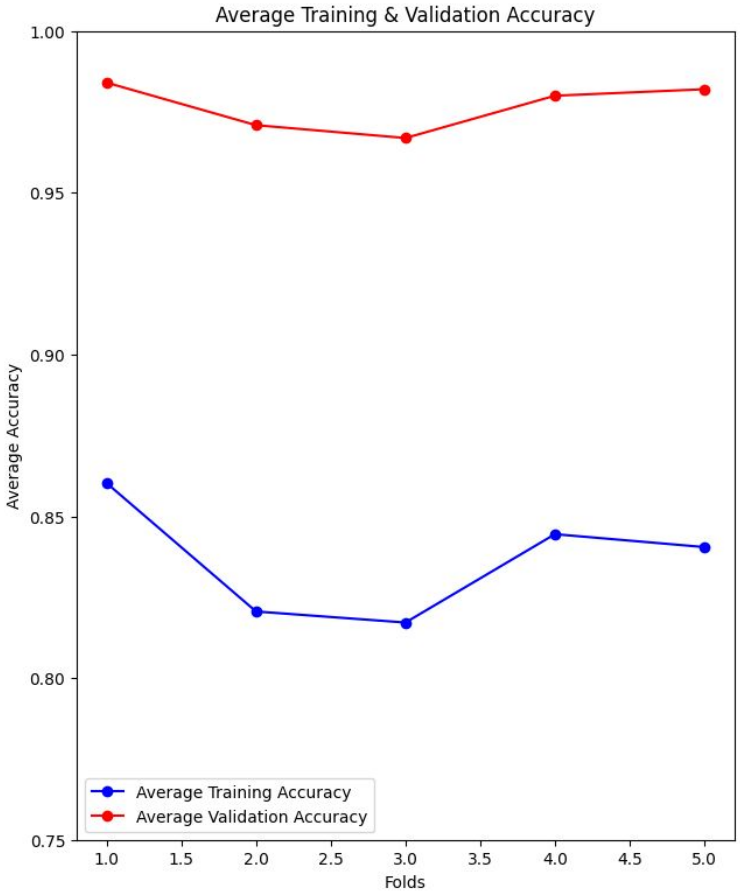
MODEL 1

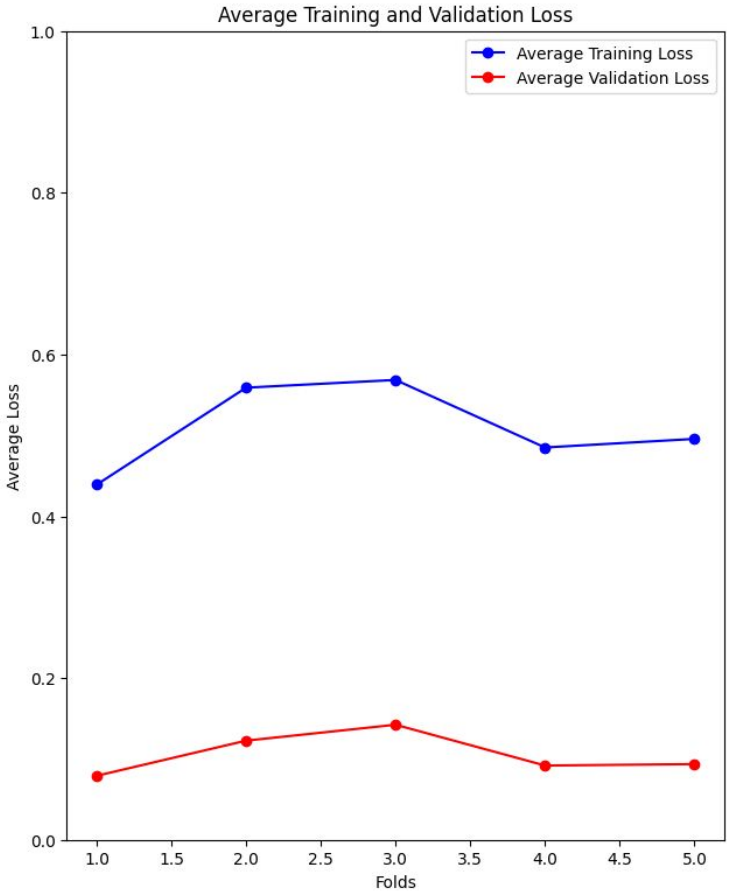
CNN (Tensorflow) + K-Fold Cross Validation

MODEL 1: CNN + K-FOLD CROSS VALIDATION

- Split training data into K subsets, using one subset as validation data for each fold
- New CNN model initialized per fold
- Compute mean validation accuracy of the folds
- Extract the best-performing model, using it to make predictions on testing set
- Evaluate model performance

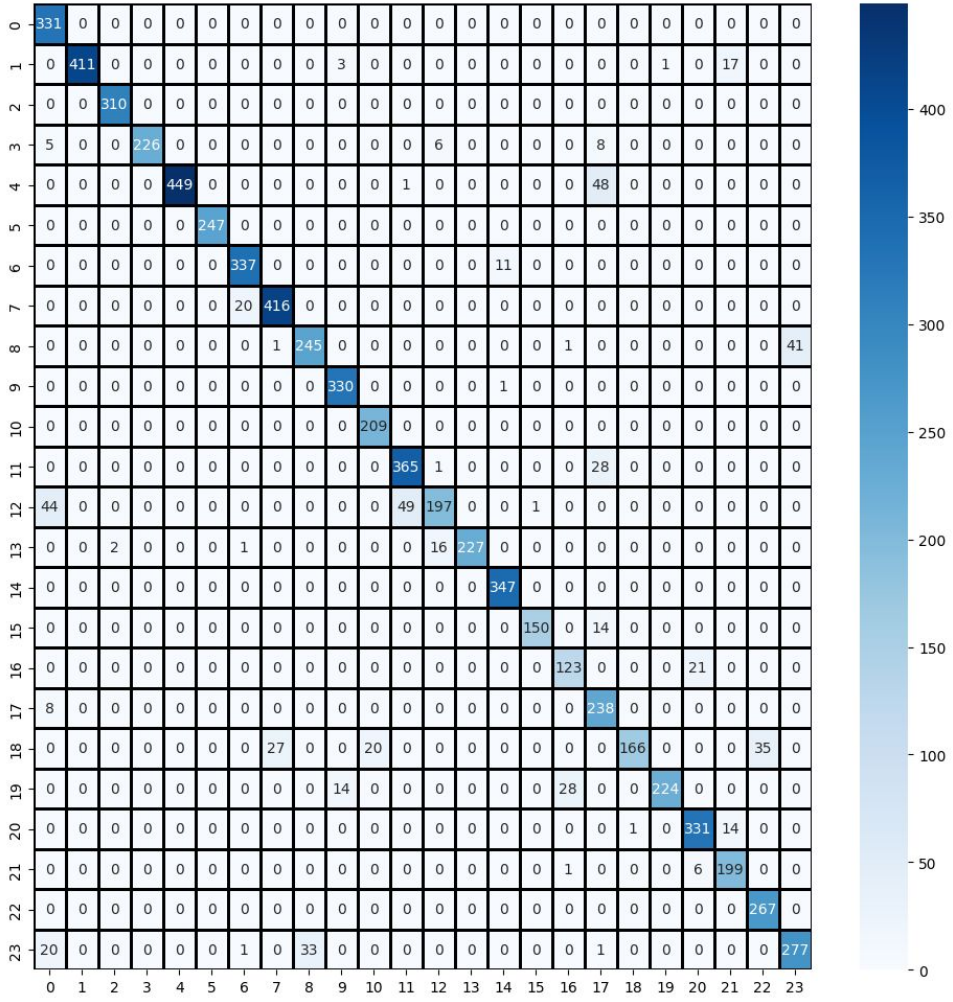
5 folds with 5 epochs per fold

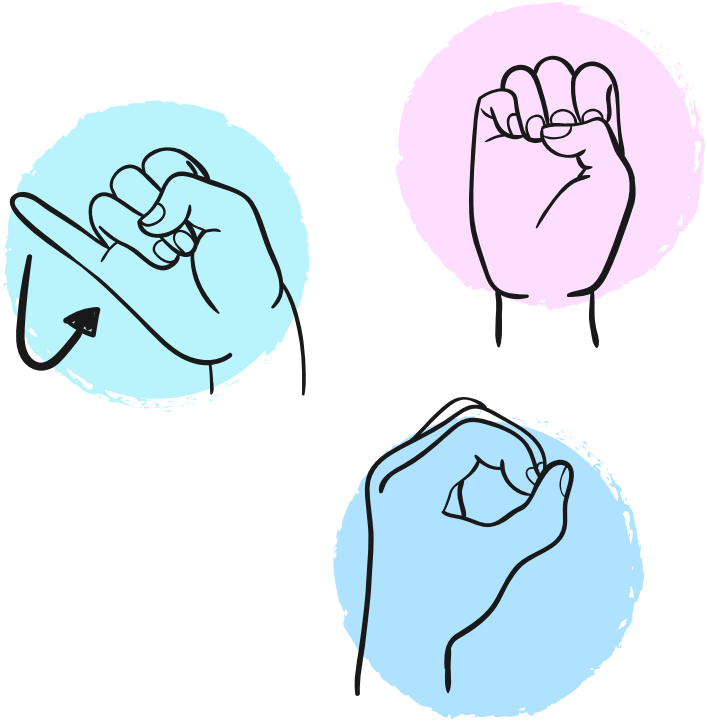




Mean Validation Accuracy: 99.93%

Test Accuracy: 92.21%



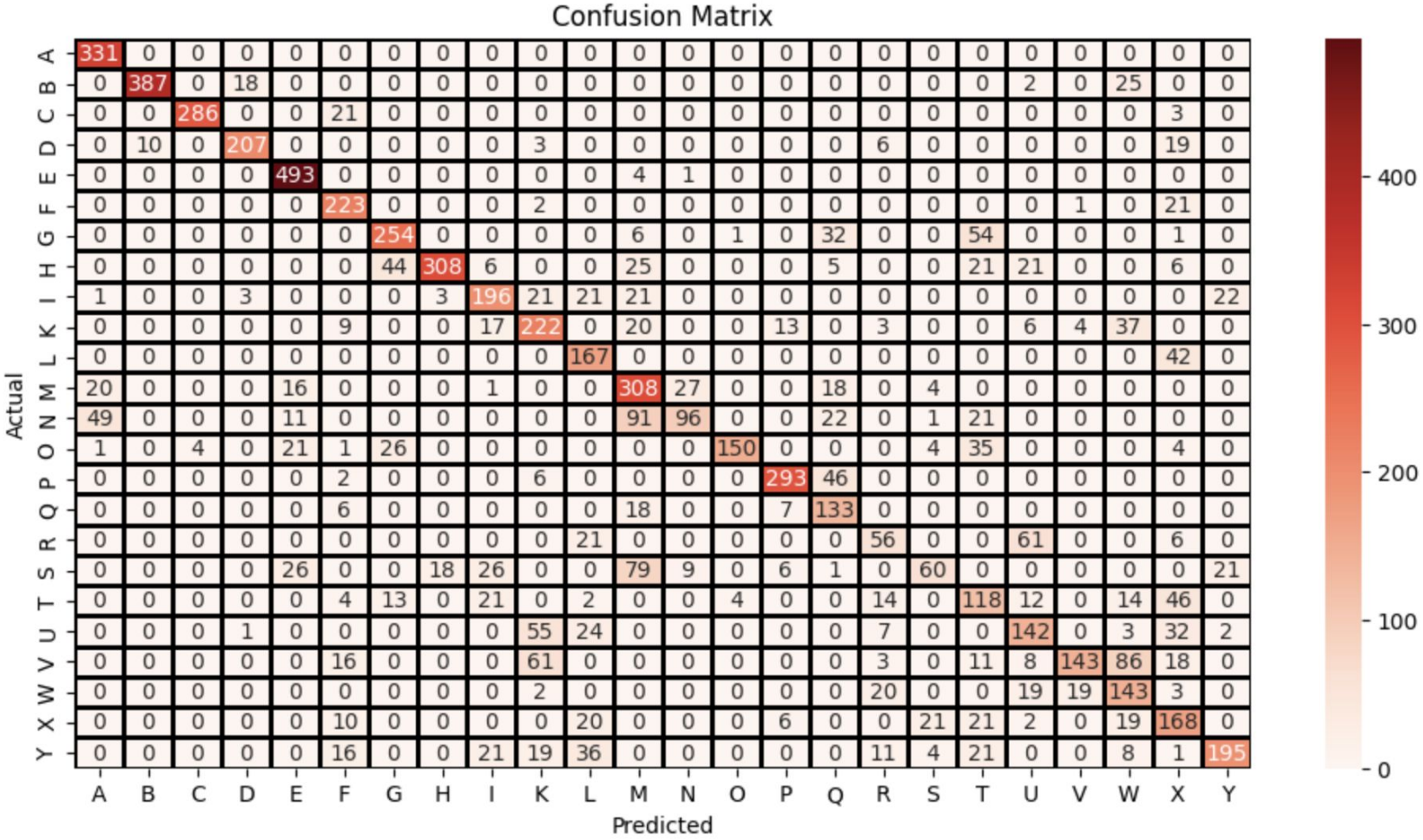


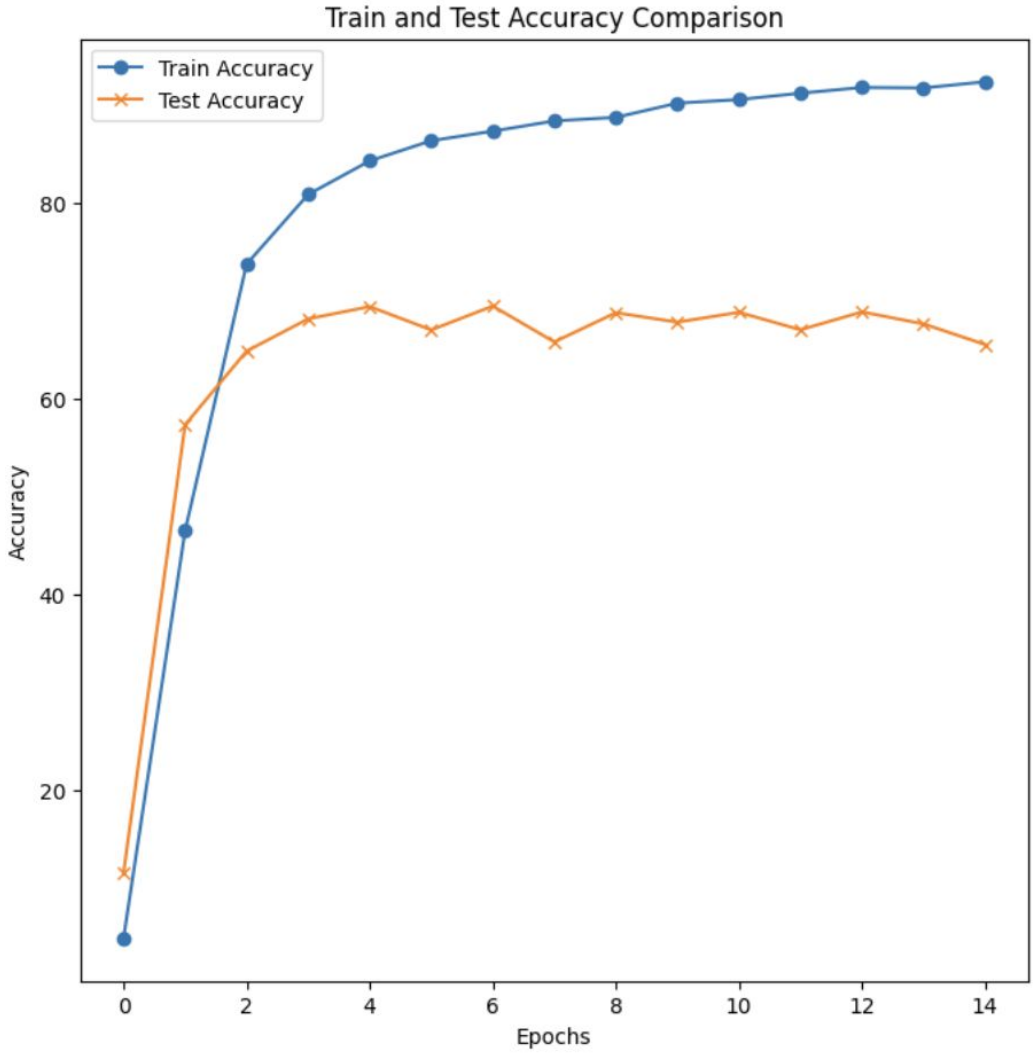
MODEL 2

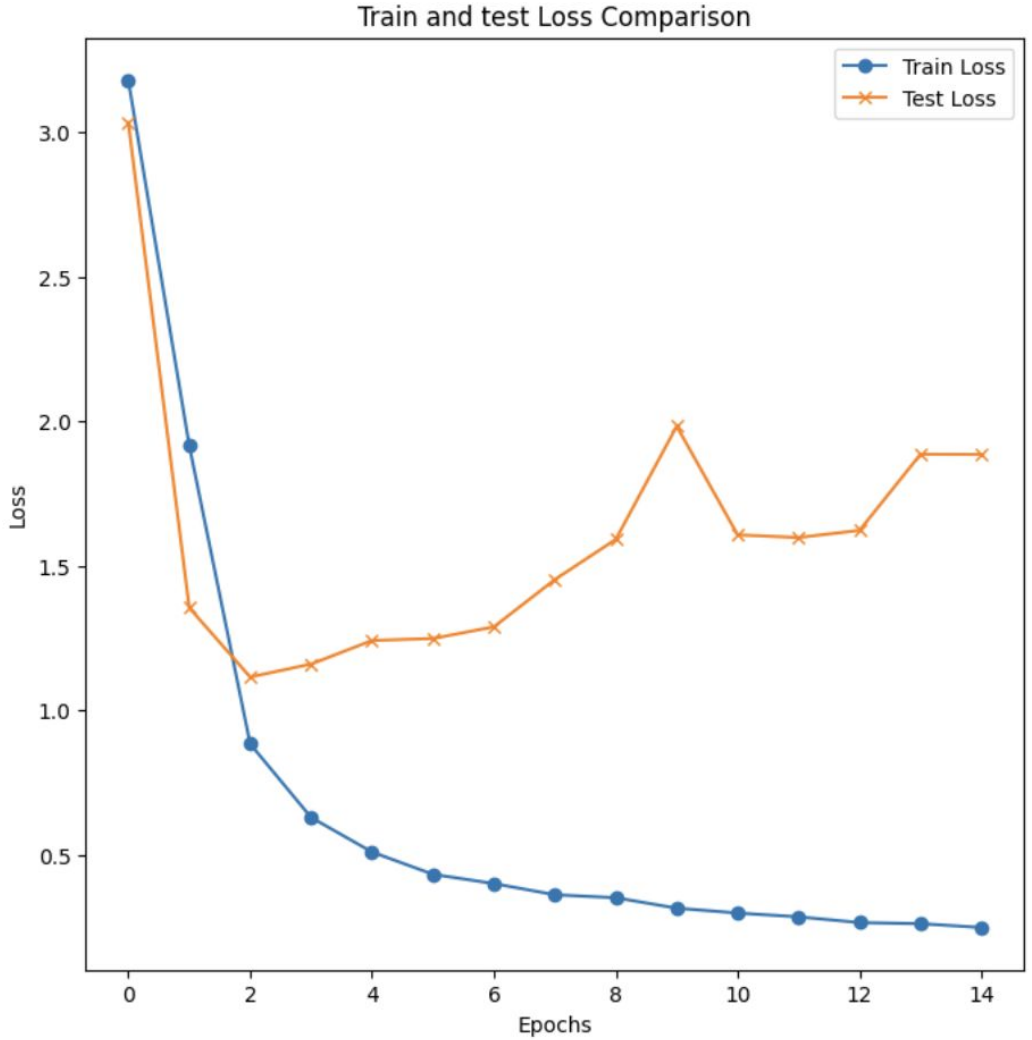
CNN - PyTorch

MODEL 2: CNN WITH PYTORCH

- Preprocessed the data into multiple batches of 512 images to train the model
- Every batch is used per one epoch to try to minimize the loss
- Keep track of the accuracies and losses during both training and testing to check for improvement
- Evaluate model performance with a confusion matrix and plots for the accuracy and loss


Confusion Matrix
Actual
Predicted
A B C D E F G H I K L M N O P Q R S T U V W X Y
A 331 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
B 0 387 0 18 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 25 0 0
C 0 0 286 0 0 21 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0
D 0 10 0 207 0 0 0 0 0 3 0 0 0 0 0 0 6 0 0 0 0 0 19 0
E 0 0 0 0 493 0 0 0 0 0 0 4 1 0 0 0 0 0 0 0 0 0 0 0
F 0 0 0 0 0 223 0 0 0 2 0 0 0 0 0 0 0 0 0 0 1 0 21 0
G 0 0 0 0 0 0 254 0 0 0 0 6 0 1 0 32 0 0 54 0 0 0 1 0
H 0 0 0 0 0 0 44 308 6 0 0 25 0 0 0 5 0 0 21 21 0 0 6 0
I 1 0 0 3 0 0 0 3 196 21 21 21 0 0 0 0 0 0 0 0 0 0 0 22
K 0 0 0 0 0 9 0 0 17 222 0 20 0 0 13 0 3 0 0 6 4 37 0 0
L 0 0 0 0 0 0 0 0 0 0 167 0 0 0 0 0 0 0 0 0 0 0 42 0
M 20 0 0 0 16 0 0 0 1 0 0 308 27 0 0 18 0 4 0 0 0 0 0 0
N 49 0 0 0 11 0 0 0 0 0 0 91 96 0 0 22 0 1 21 0 0 0 0 0
O 1 0 4 0 21 1 26 0 0 0 0 0 0 150 0 0 0 4 35 0 0 0 4 0
P 0 0 0 0 0 2 0 0 0 6 0 0 0 0 293 46 0 0 0 0 0 0 0 0
Q 0 0 0 0 0 6 0 0 0 0 0 18 0 0 7 133 0 0 0 0 0 0 0 0
R 0 0 0 0 0 0 0 0 0 0 21 0 0 0 0 0 56 0 0 61 0 0 6 0
S 0 0 0 0 26 0 0 18 26 0 0 79 9 0 6 1 0 60 0 0 0 0 0 21
T 0 0 0 0 0 4 13 0 21 0 2 0 0 4 0 0 14 0 118 12 0 14 46 0
U 0 0 0 1 0 0 0 0 0 55 24 0 0 0 0 0 7 0 0 142 0 3 32 2
V 0 0 0 0 0 16 0 0 0 61 0 0 0 0 0 0 3 0 11 8 143 86 18 0
W 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 20 0 0 19 19 143 3 0
X 0 0 0 0 0 10 0 0 0 0 20 0 0 0 6 0 0 21 21 2 0 19 168 0
Y 0 0 0 0 0 16 0 0 21 19 36 0 0 0 0 0 11 4 21 0 0 8 1 195
400
300
200
100
0


Train and Test Accuracy Comparison
Train Accuracy
Test Accuracy
Accuracy
Epochs
20
40
60
80
0
2
4
6
8
10
12
14


Train and test Loss Comparison
Train Loss
Test Loss
Loss
Epochs
3.0
2.5
2.0
1.5
1.0
0.5
0
2
4
6
8
10
12
14

Model Comparison

- Accuracy
  - Model 1: high (~90%), discrepancy between training and validation may suggest overfitting
  - Model 2: steep learning curve, accuracy still lower than Model 1 (~70%)
- Loss
  - Model 1: low and consistent
  - Model 2: sharply decreases, but loss still higher than Model 1
- Conclusion:
  - Although Model 2 was able to learn quickly, Model 1 was the better model in terms of accuracy and loss
  - Looking forward: How to improve Model 2

THANK YOU!

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