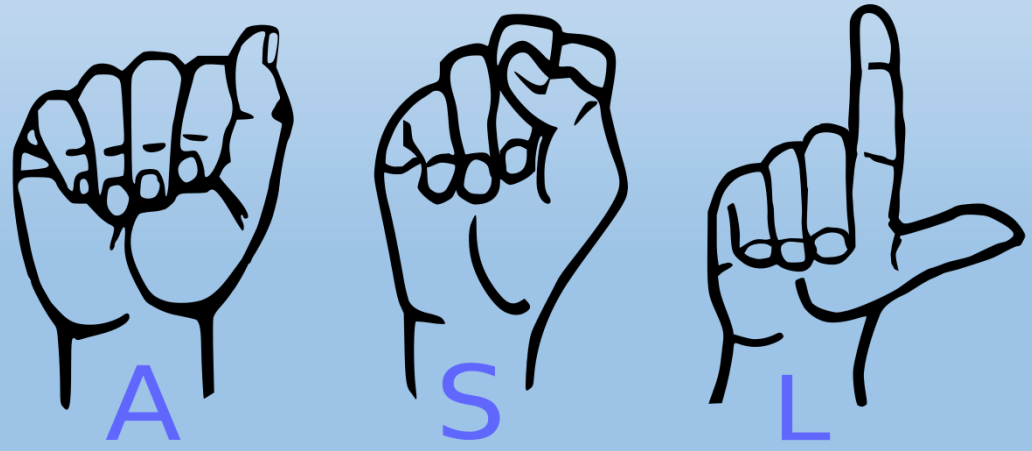
GNR-652 Course Project
American Sign Language (ASL) Detection

June 10, 2020

Submitted by:

Samyak Shah (18D070062)
Aseer Israr Ansari (18D070040)
Ritwik Khandelwal (18D070061)
Ayush Dahale (18D070041)



American Sign Language (ASL) Detection

Work Allotment and Division

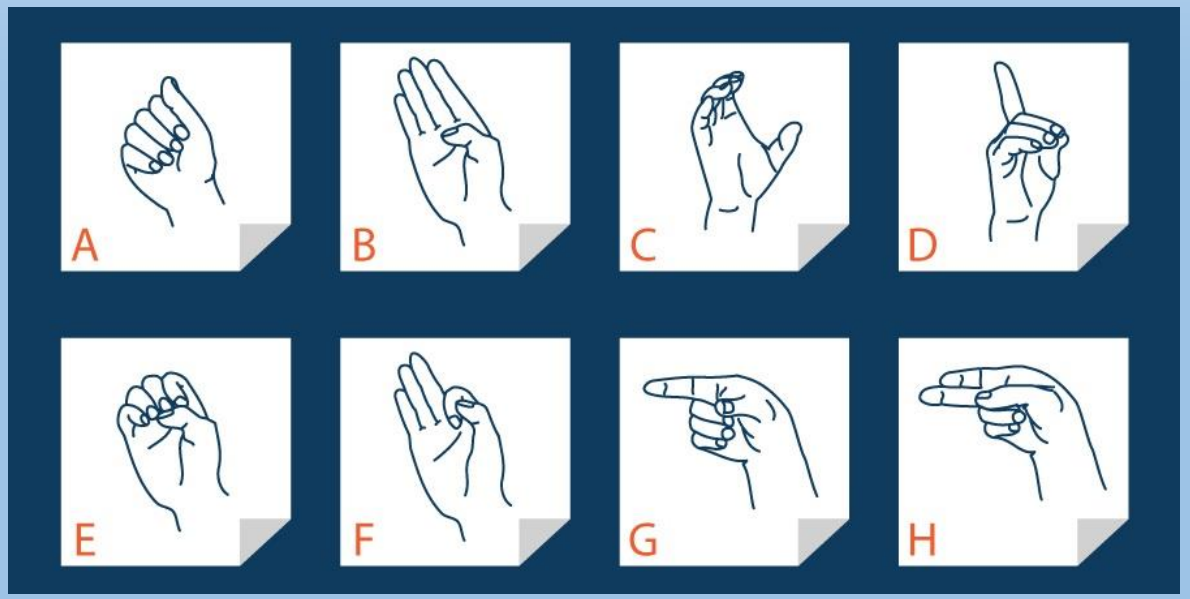
Every part of the project was first discussed in the group. Below is a rough break-up of major work done.

- Samyak Shah (18D070062): Architecture Design, Code Implementation, Data Visualization
- Aseer Israr Ansari (18D070040): Code Implementation, Optimization, Data Transformation.
- Ritwik Khandelwal (18D070061): Data Handling, Model Optimization, Documentation.
- Ayush Dahale (18D070041): Topic Ideation, Hyperparameter tuning, Documentation.

American Sign Language (ASL) Detection

Problem Statement and Motivation

This project aims to build a model to detect American Sign Language and output an English alphabet, that is robust enough to handle images of the ASL alphabet with different hands, sizes, backgrounds, lighting condition, skin colour of the hand etc.
Our project is an effort to make life easier for the destitute of vision and speech.



American Sign Language (ASL) Detection

The dataset

The dataset used in this project is ASL Alphabet dataset from Kaggle. The asl_alphabet_train subfolder contains 87,000 images. We used the images from this folder for training, cross validation and testing. The asl_alphabet_test subfolder contains only 28 images, so we will use this data for aiding the visualization of the model.

Note: The training folder contains 29 subfolders, 26 for each alphabet, one each for space, delete and blank images.

Images from the dataset.
They represent S, W and Y from left to right.

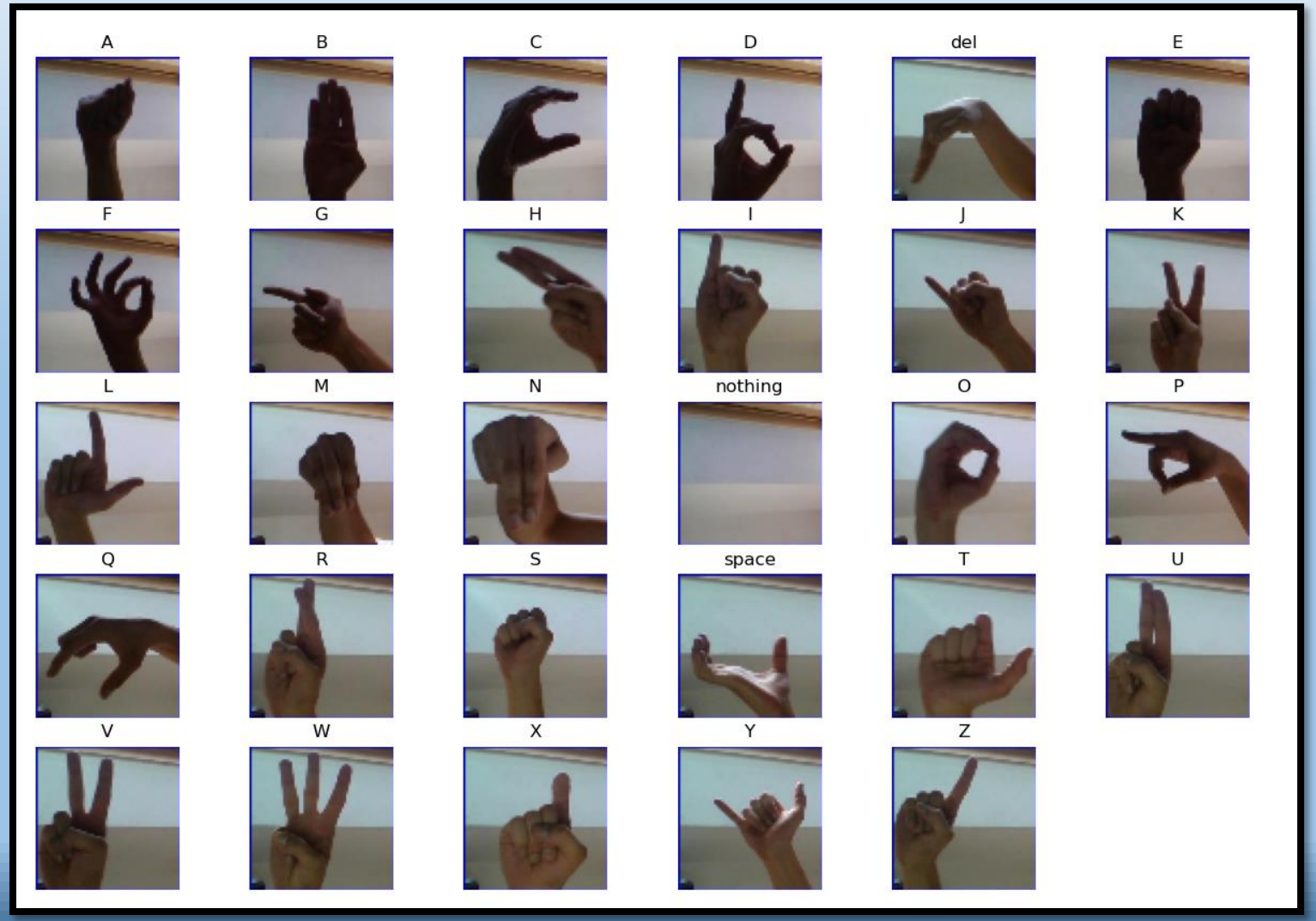
American Sign Language (ASL) Detection

Methods and Workflow

- We used a Deep Convolutional Neural Network for this image classification problem.
- Built using the Keras library from TensorFlow framework.
- Matplotlib.pyplot was used for plotting; cv2, sklearn were used during file input and data splitting respectively.
- All layers use ReLU activation function except the last dense layer which uses softmax activation.
- The layer sizes, no. of layers, no. of epochs and other hyperparameters were selected to get an optimal combination of test accuracy and algorithm speed.

American Sign Language (ASL) Detection

Inputting the images

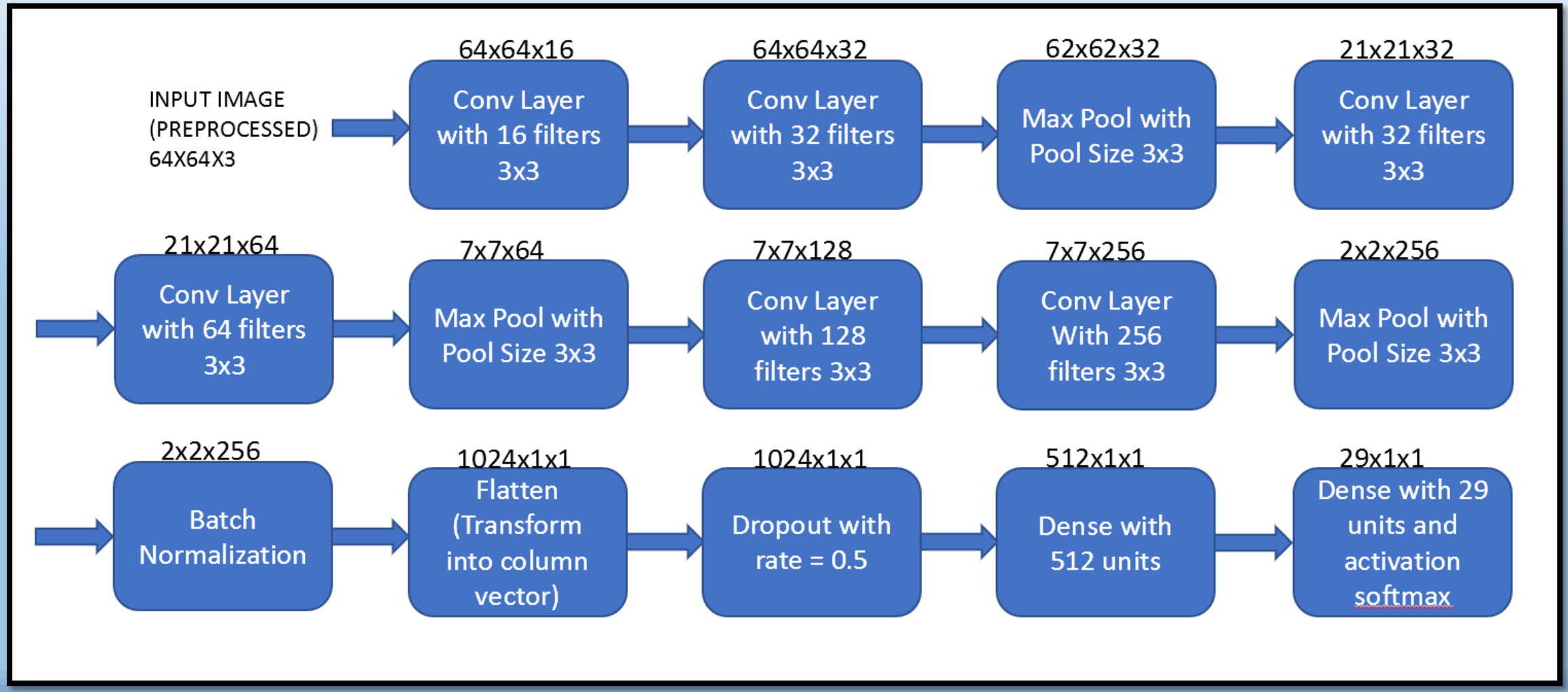


Printing one image of each kind (with the label) from the training data.

American Sign Language (ASL) Detection

The block diagram of the model

The model's architecture is stored in ***model_architecture.json*** (present in the project zip file)



American Sign Language (ASL) Detection

Training the model

Out of the 87000 images,

4350 were used in the test set.
74,385 were used in the training set.
8,265 were used for cross-validation

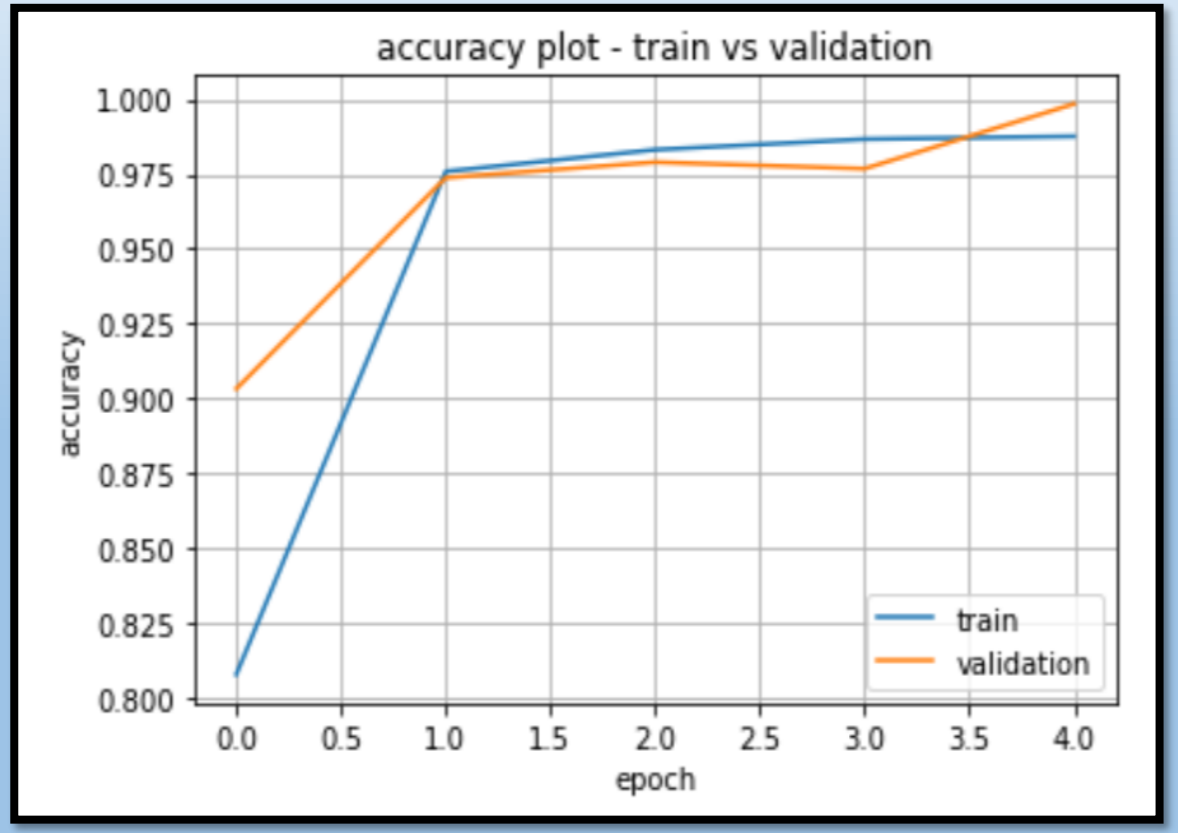
Train: Validation: Test :: 85.5: 9.5: 5

(5% of the total was used for testing, 10% of the remaining for cross validation and the rest was used for training)

```
MODEL CREATED
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_7 (Conv2D)            (None, 64, 64, 16)        448
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 64, 64, 32)        4640
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 21, 21, 32)        0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 21, 21, 32)        9248
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 21, 21, 64)        18496
_________________________________________________________________
max_pooling2d_5 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 7, 7, 128)         73856
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 7, 7, 256)         295168
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 2, 2, 256)         0
_________________________________________________________________
batch_normalization_2 (Batch (None, 2, 2, 256)         1024
_________________________________________________________________
flatten_2 (Flatten)          (None, 1024)              0
_________________________________________________________________
dropout_2 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_3 (Dense)              (None, 512)               524800
_________________________________________________________________
dense_4 (Dense)              (None, 29)                14877
=================================================================
Total params: 942,557
Trainable params: 942,045
Non-trainable params: 512
```

American Sign Language (ASL) Detection

Performance on training, validation set



No. of epochs= 5

Note: epoch no. in the graph starts from 0

American Sign Language (ASL) Detection

Model Performance

The model's trained weights are stored in ***model_weights.h5*** (present in the project zip file)

Training acc.	Validation acc.	Test acc.
98.89	98.25	98.18
98.69	99.99	99.95
98.11	98.46	98.89
98.91	98.88	98.79
99	99.91	99.91
98.16	98.81	98.92
99.04	99.67	99.9
98.53	99.03	99.49
99.09	99.89	99.95
98.79	99.84	99.86
98.721	**99.273**	**99.384**

Mean →

We did not wish to restrict the performance of the model by setting a seed in the notebook.

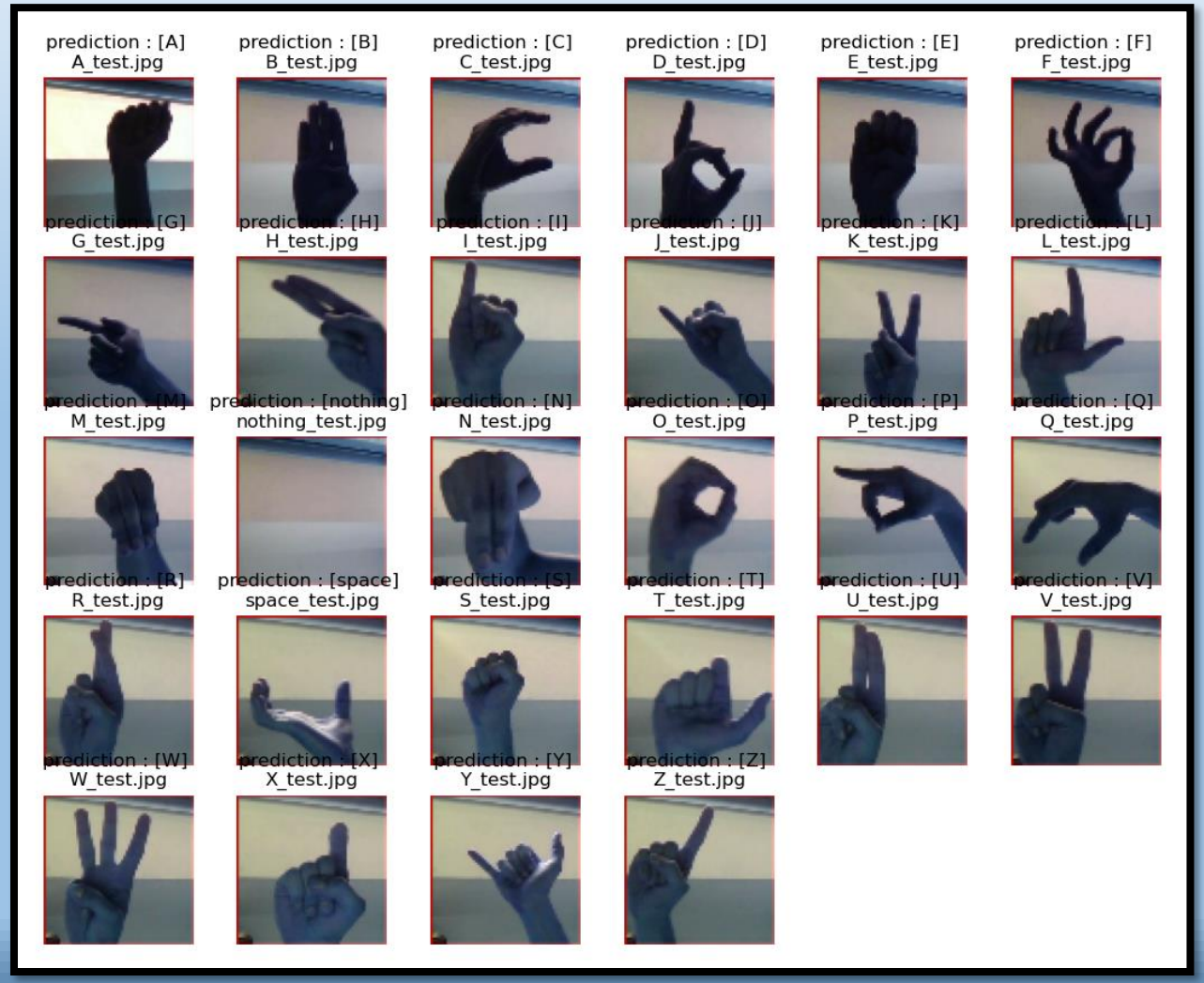
Since our train-validation-test split is random, we ran the code 10 times to calculate the average and maximum performance.

Average Test Accuracy: 99.38%
Maximum Test Accuracy: 99.95 %
Maximum Train Accuracy: 99.09%
Maximum Validation Accuracy: 99.99%

Mean performance over the 10 runs is shown in the green cells.

American Sign Language (ASL) Detection

Visualizing the predictions



We used the 28 images present in the asl_alphabet_test subfolder to print the images and the corresponding predictions.

We get pretty impressive results, with an average test accuracy of **99.38%**