Analysis of Deep Learning Models on Sign Language Recognition

Chau Nguyen, Pavan Muthamsetty, Evan Boothby
Department of Computer Science and Engineering
University of South Florida, Tampa, Florida, USA

Abstract

Sign language is the form of communication used primarily by people with hearing impairment. Due to the lack of sign language knowledge by the rest of the population, they might face problems conveying ideas and thoughts to other people. Thus, it is important to implement systems with the ability to recognize, classify, and predict sign language to help bridge the communication gap and enhance the inclusivity and effectiveness of assistive technologies. This paper analyzes the performance of 3 convolutional networks, including a baseline CNN and a pre-trained ResNet50 on 2 sign language image datasets: the ASL alphabet and the AASL datasets. The goal of this paper is to provide a comparative study between the performance of each model on a small but challenging dataset and a larger dataset to assess the model's prediction ability and generalizability.

1 Introduction

We are training 2 convolutional networks, including a baseline Convolutional Neural Network (CNN) and a ResNet50V2 model pre-trained on the ImageNet dataset with millions of images. We will perform hyperparameter optimization on each model and compare the performance of these models on 2 RGB image datasets: the ASL alphabet dataset and the Arabic Sign Language dataset. The report contains the dataset statistics in section 2, the description of each model architecture in section 3, the hyperparameter optimization for each model on each dataset in section 4, the discussion and analysis of the results of all three models in section 5, and the conclusion in section 6.

2 Dataset Statistics

2.1 American Sign Language Alphabet Dataset

The American Sign Language Alphabet (ASL Alphabet) dataset is a collection of color images of the American Sign Language alphabet, which contains

87,000 training images that are 200x200 pixels (**?**). There are 29 classes, including 26 classes for the letters from A to Z and 3 classes for SPACE, DELETE, and NOTHING. Figure 1 shows a sample image for each class.

The original test dataset contains only 29 images to encourage the use of real-world test images. However, for the scope of this project, we will randomly split the original training dataset with the following ratios: 60% for training, 20% for validation, and 20% for testing.

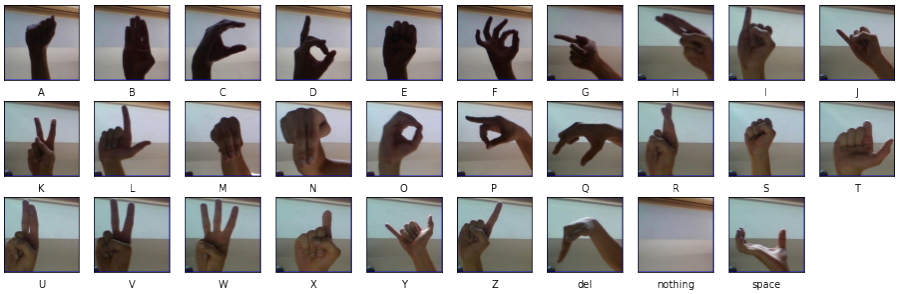


Figure 1: Sample images from the ASL Alphabet dataset

When training the baseline CNN model, we resized the images into shapes (32, 32, 3). However, due to the complexity of the ResNet50V2 model, we resized the images into shapes (64, 64, 3) to keep more information.

2.2 Arabic Alphabet Sign Language Dataset

The RGB Arabic Alphabet Sign Language (AASL) dataset consists of 7,857 raw and fully labeled RGB images of the Arabic sign language alphabets (**?**). It was collected from 200 signers with different settings, such as lighting, background, image orientation, image size, and resolution. Although this dataset is smaller than the ASL alphabet dataset, the variety in image settings makes it more difficult to train. Thus, the training results for this dataset provide more insight into each model's ability to generalize on unseen data.

When training the baseline CNN model, we resized the images into shapes (32, 32, 3). However, due to the complexity of the ResNet50V2 model, we resized the images into shapes (256, 256, 3) to retain more information.

3 Model Architectures

3.1 Baseline CNN

The baseline CNN model has 6 convolutional layers: the first 2 layers have 64 units, the next 2 layers have 128 units, and the last 2 layers have 256 units. A max-pooling layer of size 2x2 and batch normalization are added after every two convolutional layers.

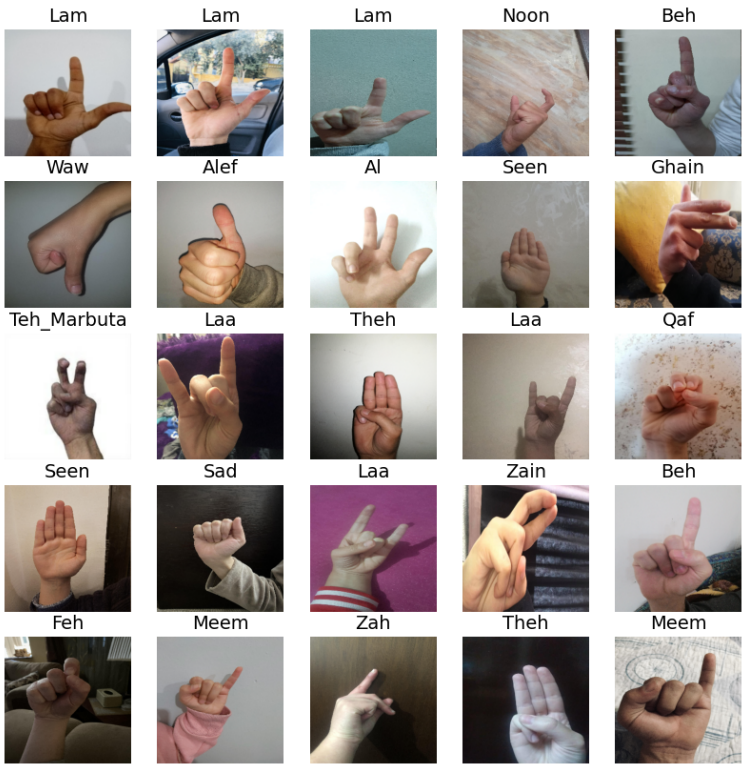
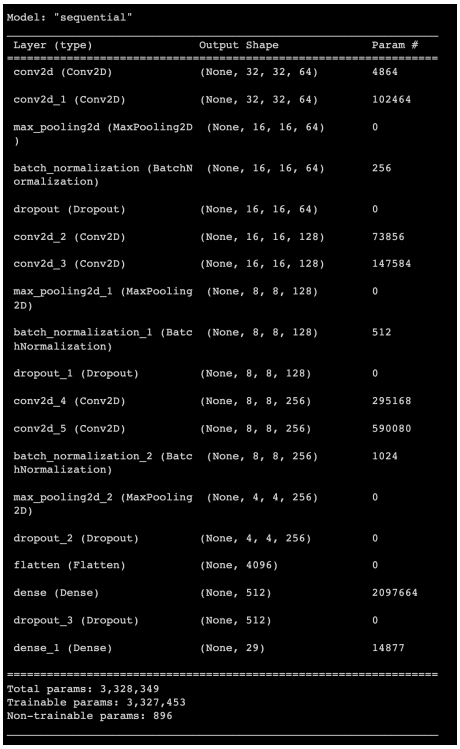


Figure 2: Sample images from the AASL dataset

Figure 3 shows the model summary for the ASL Alphabet dataset. We add 4 dropout layers in total with alpha values of 0.25 and 0.3.



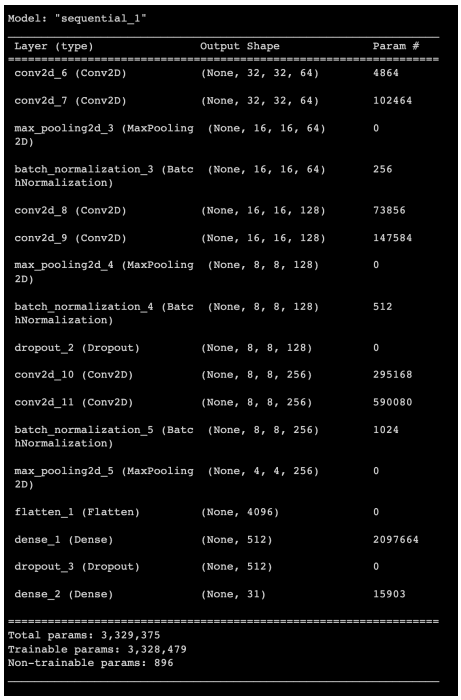
```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 32, 32, 64)        4864

conv2d_1 (Conv2D)            (None, 32, 32, 64)        102464

max_pooling2d (MaxPooling2D  (None, 16, 16, 64)        0
)

batch_normalization (BatchN  (None, 16, 16, 64)        256
ormalization)

dropout (Dropout)            (None, 16, 16, 64)        0

conv2d_2 (Conv2D)            (None, 16, 16, 128)       73856

conv2d_3 (Conv2D)            (None, 16, 16, 128)       147584

max_pooling2d_1 (MaxPooling  (None, 8, 8, 128)         0
2D)

batch_normalization_1 (Batc  (None, 8, 8, 128)         512
hNormalization)

dropout_1 (Dropout)          (None, 8, 8, 128)         0

conv2d_4 (Conv2D)            (None, 8, 8, 256)         295168

conv2d_5 (Conv2D)            (None, 8, 8, 256)         590080

batch_normalization_2 (Batc  (None, 8, 8, 256)         1024
hNormalization)

max_pooling2d_2 (MaxPooling  (None, 4, 4, 256)         0
2D)

dropout_2 (Dropout)          (None, 4, 4, 256)         0

flatten (Flatten)            (None, 4096)              0

dense (Dense)                (None, 512)               2097664

dropout_3 (Dropout)          (None, 512)               0

dense_1 (Dense)              (None, 29)                14877

=================================================================
Total params: 3,328,349
Trainable params: 3,327,453
Non-trainable params: 896
_________________________________________________________________
```

Figure 3: Baseline CNN model for ASL dataset (4 dropout layers)

Figure 4 shows the model summary for the AASL dataset with only 2 dropout layers with alpha values of 0.25 and 0.5, respectively.

3.2 Pre-trained ResNet50V2

For the third model, we will use a ResNet50V2 model that is pre-trained on the ImageNet dataset with millions of images. We will add a fully connected layer with 512 hidden units and a dropout layer before the output layer for classification. Figure 5 shows the summary of the pre-trained ResNet50V2 model.



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_6 (Conv2D)           (None, 32, 32, 64)        4864

 conv2d_7 (Conv2D)           (None, 32, 32, 64)        102464

 max_pooling2d_3 (MaxPooling  (None, 16, 16, 64)       0
 2D)

 batch_normalization_3 (Batc  (None, 16, 16, 64)       256
 hNormalization)

 conv2d_8 (Conv2D)           (None, 16, 16, 128)       73856

 conv2d_9 (Conv2D)           (None, 16, 16, 128)       147584

 max_pooling2d_4 (MaxPooling  (None, 8, 8, 128)        0
 2D)

 batch_normalization_4 (Batc  (None, 8, 8, 128)        512
 hNormalization)

 dropout_2 (Dropout)         (None, 8, 8, 128)         0

 conv2d_10 (Conv2D)          (None, 8, 8, 256)         295168

 conv2d_11 (Conv2D)          (None, 8, 8, 256)         590080

 batch_normalization_5 (Batc  (None, 8, 8, 256)        1024
 hNormalization)

 max_pooling2d_5 (MaxPooling  (None, 4, 4, 256)        0
 2D)

 flatten_1 (Flatten)         (None, 4096)              0

 dense_1 (Dense)             (None, 512)               2097664

 dropout_3 (Dropout)         (None, 512)               0

 dense_2 (Dense)             (None, 31)                15903

=================================================================
Total params: 3,329,375
Trainable params: 3,328,479
Non-trainable params: 896
_________________________________________________________________
```

Figure 4: Baseline CNN model for Arabic (AASL) dataset (2 dropout layers)

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 resnet50v2 (Functional)     (None, 2048)              23564800

 module_wrapper (ModuleWrapp  (None, 2048)             0
 er)

 module_wrapper_1 (ModuleWra  (None, 512)              1049088
 pper)

 module_wrapper_2 (ModuleWra  (None, 512)              0
 pper)

 module_wrapper_3 (ModuleWra  (None, 29)               14877
 pper)

=================================================================
Total params: 24,628,765
Trainable params: 1,063,965
Non-trainable params: 23,564,800
_________________________________________________________________
```

Figure 5: Pre-trained ResNet50V2 model with additional fully connected layer

4 Hyperparameter Optimization

This section provides the hyperparameter optimization and data augmentation for each model on both datasets. For all models and datasets, we use Categorical Crossentropy as our loss function.

4.1 Data Augmentation

4.1.1 AASL Dataset

For the AASL dataset, we use the following data augmentation techniques: shear range of 0.2, zooming range of 0.2, width and height shift range of 0.1, and random horizontal flip.

4.1.2 ASL Alphabet Dataset

For the ASL Alphabet dataset, we use the following data augmentation techniques: rotation range of 10 degrees, zooming range of 0.1, and width and height shift range of 0.1. For the ResNet50V2 model, we also add shear range of 0.2 to add more data.

4.2 Baseline CNN

This section shows the hyperparameter optimization for the CNN model on each dataset.

Since the AASL dataset is more challenging, we will try to optimize the CNN model on this dataset first before using the same model settings for the ASL Alphabet dataset.

Figure 6 shows the hyperparameter optimization for this model on the AASL dataset after training for 50 epochs. Originally, the CNN model had 3 pairs of convolutional layers with 32, 64, and 128 hidden units, respectively. Due to the limited number of filters, the original model setting did not perform well on the AASL dataset. Thus, we decided to increase the number of filters to 64, 128, and 256. From the third trial, having 4 dropout layers lowers the accuracy we could achieve in 50 epochs. Thus, we decide to reduce the number of dropout layers for this dataset to 2 dropout layers with alpha values of 0.25 and 0.5. Thus, the final hyperparameters for training the CNN model on the AASL dataset are: Adam optimizer with a learning rate of 1e-4, batch size of 32, hidden units of 64, 128, and 256, and 2 dropout layers with alpha values of 0.25 and 0.5.

Figure 7 shows the hyperparameter optimization for the CNN model on the ASL Alphabet dataset after 10 epochs. From the results in Figure 6, we decide to keep the same number of filters for the CNN model. Thus, the final hyperparameters for training the CNN model on this dataset are: Adam optimizer with learning rate of 1e-4, batch size of 128, hidden units of 64, 128, and 256, and 4 dropout layers with alpha values of 0.25 and 0.3.

Hyperparameters	Learning rate	Batch size	Train Accuracy	Val accuracy
Adam (32-64-128) dropout 0.2 with data aug	1e-3	32	51.96	51.97
Adam (32-64-128) dropout 0.3 with data aug	1e-4	32	86.38	63.93
Adam (64-128-256) dropout 0.25, 0.25, 0.25, 0.3 with data aug	1e-4	32	70.17	60.69
Adam (64-128-256) dropout 0.3, with data aug	1e-4	32	90.98	63.74
RMSprop (64-128-256) dropout 0.25, 0.5, with data aug	1e-4	32	68.47	65.46
Adam (64-128-256) dropout 0.25, 0.5 with data aug	**1e-4**	**32**	**69.62**	**72.58**

Figure 6: Hyperparameter optimization for the baseline CNN model on ASL dataset

Hyperparameters	Learning rate	Batch size	Train Accuracy	Val accuracy
Adam (64-128-256) dropout 0.25, 0.25, 0.25, 0.3 with data aug	**1e-4**	**128**	**96.72**	**97.65**
Adam (64-128-256) dropout 0.25, 0.25, 0.25, 0.3 with data aug	1e-4	512	91.20	96.47

Figure 7: Hyperparameter optimization for the baseline CNN model on Arabic (AASL) dataset

4.3 Pre-trained ResNet50V2

In this section, we will perform hyperparameter optimization on the pre-trained ResNet50V2 model on both datasets.

Figure 8 shows the hyperparameter optimization for the pre-trained model on the AASL dataset after training for 20 epochs. Increasing the image size from 64x64 to 256x256 pixels (using Adam optimizer) increases the validation accuracy from 38.42% to 62.72%. Trials 6 and 7 show that the Adam optimizer is a better choice for this model on the AASL dataset. Although not shown here, these optimizers perform better than the SGD with optimum on this model and dataset. With a dropout layer with alpha value of 0.5 and reduced batch size, the model is underfitting. Thus, after reducing alpha value of the dropout layer to 0.3, the model achieves 77.67% with only 20 epochs. Thus, the hyperparameters we will use to train the pre-trained ResNet50V2 model for the Arabic, or AASL, Alphabet dataset are: Adam optimizer with learning rate of 1e-4, batch size of 2, additional fully connected layer with 512 hidden units, and dropout layer of 0.3. Note that the input shape is changed to (256, 256, 3) since this improves the pre-trained model's performance by keeping useful photo information.

Hyperparameters	Learning rate	Batch size	Train Accuracy	Val accuracy
ResNet50V2, Adam, no data aug, input shape (64, 64, 3)	1e-3	32	64.25	38.42
ResNet50V2, Adam, additional layer (1024 units), with data aug, input shape (64, 64, 3)	1e-4	32	44.66	41.09
ResNet50V2, RMSprop, additional layer (1024 units), with data aug, input shape (64, 64, 3)	1e-4	32	42.67	41.67
ResNet50V2, RMSprop, no data aug, input shape (256, 256, 3)	1e-4	32	93.76	62.72
ResNet50V2, Adam, no data aug, input shape (256, 256, 3)	1e-4	32	93.57	60.75
ResNet50V2, RMSprop, additional layer (512 units), dropout 0.5, with data aug, input shape (256, 256, 3)	1e-4	2	77.32	70.29
ResNet50V2, Adam, additional layer (512 units), dropout 0.5, with data aug, input shape (256, 256, 3)	1e-4	2	63.14	73.73
ResNet50V2, Adam, additional layer (512 units), dropout 0.3, with data aug, input shape (256, 256, 3)	**1e-4**	**2**	**77.15**	**77.67**

Figure 8: Hyperparameter optimization for the pre-trained ResNet50V2 model on Arabic (AASL) dataset

Figure 9 shows the hyperparameter optimization for the pre-trained model on the ASL Alphabet dataset after training for 10 epochs. Originally, we were trying to optimize the performance on the pre-trained ResNet50 model, but

even after adding an additional fully connected layer with 1024 hidden units, the validation accuracy could only reach 47.59% after 10 epochs. After changing to the ResNet50V2 model, the performance improved significantly to 87.89% without data augmentation. One interesting keypoint is that data augmentation for this dataset on the pre-trained model actually worsens the performance to 85.51%. After we added a fully connected layer with 512 hidden units and dropout layer with alpha value of 0.6, the performance improved to 98.06% without even using data augmentation. When reducing the learning rate of the Adam optimizer from 1e-3 in trial 7 to 1e-4 in trial 8, underfitting is reduced significantly by around 4% and validation accuracy reaches 98.95%. Note that since we are only training for 10 epochs with regularization (from the dropout layer), the training accuracy is slightly lower than the validation accuracy. Thus, the hyperparameters we will use to train the pre-trained ResNet50V2 model for the ASL Alphabet dataset are: Adam optimizer with learning rate of 1e-4, batch size of 32, additional fully connected layer with 512 hidden units, and dropout layer of 0.6.

Hyperparameters	Learning rate	Batch size	Train Accuracy	Val accuracy
ResNet50, Adam, no data aug	1e-4	32	31.95	39.02
ResNet50, Adam, with data aug	1e-3	32	46.53	53.79
ResNet50, Adam, additional layer (1024 units), with data aug	1e-3	32	41.20	47.59
ResNet50V2, Adam, no data aug	1e-3	32	84.10	87.89
ResNet50V2, Adam, additional layer (512 units), with data aug	1e-3	32	82.89	85.51
ResNet50V2, Adam, additional layer (512 units), dropout 0.5, no data aug	1e-3	32	98.98	98.03
ResNet50V2, Adam, additional layer (512 units), dropout 0.6, no data aug	1e-3	32	93.01	98.06
ResNet50V2, Adam, additional layer (512 units), dropout 0.6, no data aug	**1e-4**	**32**	**97.80**	**98.95**

Figure 9: Hyperparameter optimization for the pre-trained ResNet50V2 model on ASL dataset

5 Training Results

This section will show the results from training the CNN and pre-trained ResNet50V2 models for 50 epochs per trial for 5 trials each with the set of hyperparameters as chosen in section 4 on both the ASL and AASL datasets.

5.1 ASL Dataset

5.1.1 CNN Model

Figure 9 shows the training, validation, and test accuracy of the CNN model on the ASL dataset after 50 epochs for 5 trials. The average training accuracy is 98.95%, the average validation accuracy is 99.66%, and the average test accuracy is 99.73%. The standard errors are 0.06713, 0.05576, and 0.05738, respectively.

Trial #	Train Accuracy	Validation Accuracy	Test Accuracy
1	99.26	99.89	99.97
2	98.56	99.66	99.82
3	99.28	99.8	99.79
4	98.65	99.18	99.23
5	99.01	99.75	99.83
Avg	**98.95**	**99.66**	**99.73**
Std Err	0.06713	0.05576	0.05738

Figure 10: CNN model results on ASL dataset

Figure 10 shows the training accuracy and loss curves for the CNN model on the ASL dataset. Note that this is the training curves from the last trial.

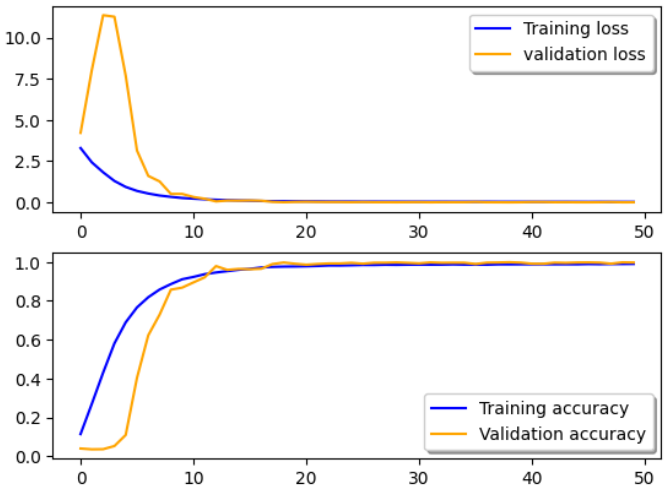


Figure 11: CNN model learning curve on ASL dataset

Figure 11 shows the mean classification report and figure 12 shows the mean confusion matrix from training the CNN model on the ASL dataset for 5 trials.

	precision	recall	f1-score	support
0	0.988	0.996	0.990	595.0
1	1.000	0.982	0.992	617.0
2	1.000	1.000	1.000	585.0
3	1.000	1.000	1.000	600.0
4	1.000	1.000	1.000	615.0
5	1.000	1.000	1.000	654.0
6	0.996	1.000	1.000	609.0
7	1.000	0.990	1.000	615.0
8	1.000	1.000	1.000	614.0
9	1.000	1.000	1.000	621.0
10	1.000	1.000	1.000	582.0
11	1.000	1.000	1.000	616.0
12	1.000	1.000	1.000	584.0
13	1.000	1.000	1.000	578.0
14	1.000	1.000	1.000	617.0
15	1.000	1.000	1.000	569.0
16	1.000	1.000	1.000	583.0
17	1.000	1.000	1.000	594.0
18	1.000	1.000	1.000	591.0
19	0.984	1.000	0.988	576.0
20	1.000	1.000	1.000	593.0
21	1.000	1.000	1.000	597.0
22	1.000	1.000	1.000	568.0
23	1.000	1.000	1.000	633.0
24	1.000	0.992	0.998	611.0
25	1.000	1.000	1.000	610.0
26	0.990	1.000	1.000	635.0
27	1.000	1.000	1.000	605.0
28	0.996	1.000	1.000	533.0
accuracy	0.000	0.000	1.000	17400.0
macro_avg	1.000	1.000	1.000	17400.0
weighted_avg	1.000	1.000	1.000	17400.0

Figure 12: CNN mean classification report on ASL dataset

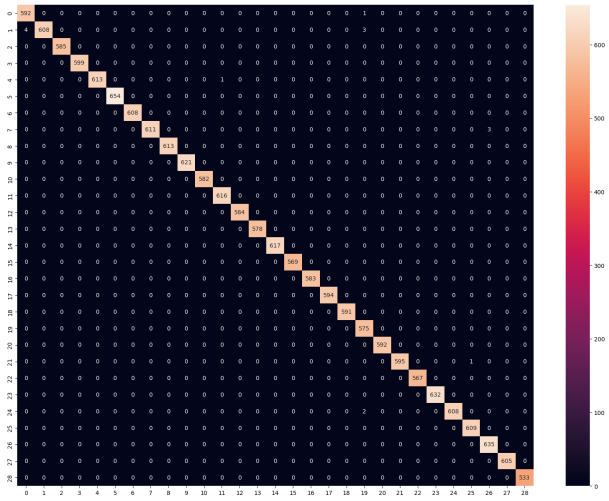


Figure 13: CNN mean confusion matrix on ASL dataset

5.1.2 Pre-trained ResNet50V2 Model

Figure 14 shows the training results of the pre-trained ResNet50V2 model on the ASL dataset after 50 epochs for 5 trials. The average training accuracy is 99.68%, the average validation accuracy is 99.73%, and the average test accuracy is 99.67%. The standard errors are 0.01828, 0.01625, and 0.01568, respectively.

Trial #	Train Accuracy	Validation Accuracy	Test Accuracy
1	99.67	99.71	99.71
2	99.7	99.71	99.62
3	99.63	99.78	99.69
4	99.74	99.75	99.66
5	99.67	99.69	99.69
Avg	**99.68**	**99.73**	**99.67**
Std Err	0.01828	0.01625	0.01568

Figure 14: Results from training pre-trained model on ASL dataset

Figure 15 shows the training accuracy and loss curve from training the pre-trained model for 50 epochs. These curves are generated from the best trial.

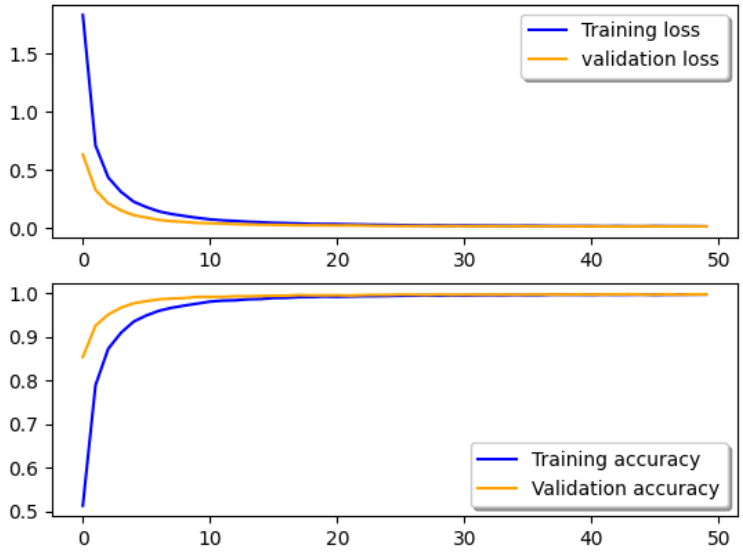


Figure 15: Training accuracy and loss curve from pre-trained model on ASL dataset

Figure 16 shows the mean confusion matrix from 5 trials and Figure 17 shows the classification report from the best trial of the pre-trained ResNet50V2 model on the ASl dataset.

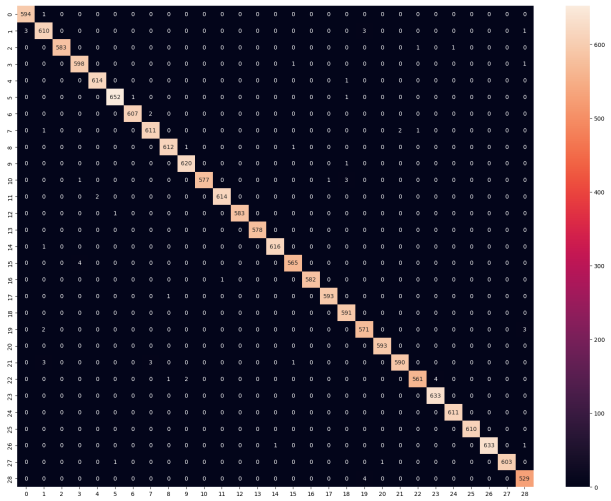


Figure 16: ResNet50V2 mean confusion matrix on ASL dataset

	precision	recall	f1-score	support
0	0.99	1.00	1.00	595
1	0.99	0.99	0.99	617
2	1.00	1.00	1.00	585
3	0.99	1.00	0.99	600
4	1.00	1.00	1.00	615
5	1.00	1.00	1.00	654
6	1.00	1.00	1.00	609
7	0.99	0.99	0.99	615
8	1.00	1.00	1.00	614
9	1.00	1.00	1.00	621
10	1.00	0.99	1.00	582
11	1.00	1.00	1.00	616
12	1.00	1.00	1.00	584
13	1.00	1.00	1.00	578
14	1.00	1.00	1.00	617
15	0.99	0.99	0.99	569
16	1.00	1.00	1.00	583
17	1.00	1.00	1.00	594
18	0.99	1.00	0.99	591
19	0.99	0.99	0.99	576
20	1.00	1.00	1.00	593
21	1.00	0.99	0.99	597
22	1.00	0.99	0.99	568
23	0.99	1.00	1.00	633
24	1.00	1.00	1.00	611
25	1.00	1.00	1.00	610
26	1.00	1.00	1.00	635
27	1.00	1.00	1.00	605
28	0.99	0.99	0.99	533
accuracy			1.00	17400
macro avg	1.00	1.00	1.00	17400
weighted avg	1.00	1.00	1.00	17400

Figure 17: ResNet50V2 best classification report on ASL dataset

5.2 AASL Dataset

5.2.1 CNN Model

Figure 18 shows the training results of the CNN model on the Arabic dataset (AASL) after 50 epochs. The average training accuracy is 67.38%, the average validation accuracy is 70.15%, and the average test accuracy is 69.95%. The standard errors are 1.0612, 1.0613, and 0.9299, respectively. The results show that the model is currently underfitting when only trained for 50 epochs. However, as shown in the hyperparameter optimization section for this model and dataset, additional dropout layers with values 0.25 and 0.5 are required to improve the test accuracy of the baseline model on this challenging dataset. Since regularization is performed, the training accuracy is lower than the validation accuracy for these number of epochs.

Trial #	Train Accuracy	Validation Accuracy	Test Accuracy
1	69.62	72.58	72.14
2	67.92	71.88	70.29
3	63.48	67.62	66.79
4	67.11	67.62	69.21
5	68.77	71.06	71.31
Avg	**67.38**	**70.15**	**69.95**
Std Err	1.0612	1.0613	0.9299

Figure 18: Results from training model 2 (RNN with dropout layer)

We continue to train each trial for an additional 50 epochs (100 epochs in total) and record the results in Figure 19. With increasing number of training epochs, the underfitting is resolved and results in an average of 74.98% test accuracy (an improvement of 5.03%). These results show that for more diverse and challenging dataset, it is important to have additional training data (i.e., by increasing the number of training epochs) to ensure adequate prediction results. However, when comparing to the pre-trained model, we will only use the results of the first 50 epochs.

Figure 20 shows the training accuracy and loss curves for 50 epochs. Both curves have not converged yet and require additional training time or data to improve performance. Note that these curves are generated from trial 2 in Figure 18.

Figure 21 shows the training accuracy and loss curves from training for a total of 100 epochs.

Figure 22 shows the mean confusion matrix from training the CNN model on the Arabic dataset for 50 epochs. Figure 23 shows a classification report from the best trial for the CNN model.

Trial #	Train Accuracy	Validation Accuracy	Test Accuracy
1	82.09	76.78	75.83
2	78.74	76.40	75.00
3	71.42	72.46	70.81
4	84.64	78.05	78.82
5	77.89	73.66	74.43
Avg	**78.96**	**75.47**	**74.98**
Std Err	2.237	1.039	1.288

Figure 19: Results from training model 2 (RNN with dropout layer)

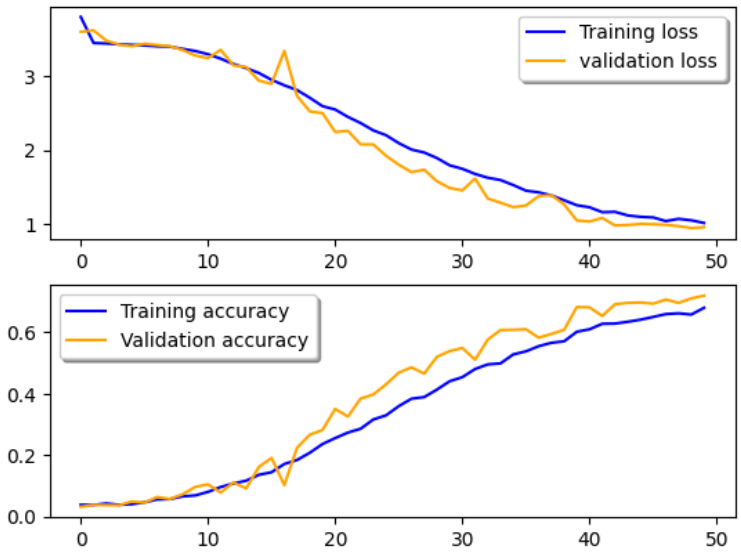


Figure 20: Results from training model 2 (RNN with dropout layer)

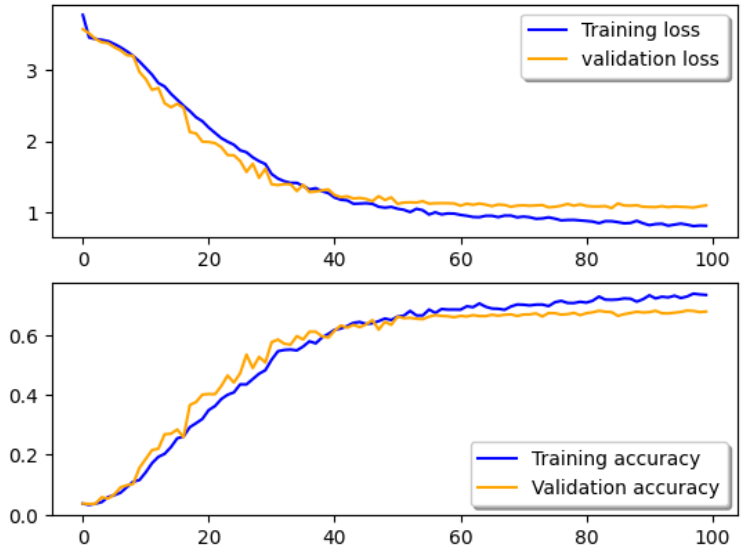


Figure 21: Results from training model 2 (RNN with dropout layer)

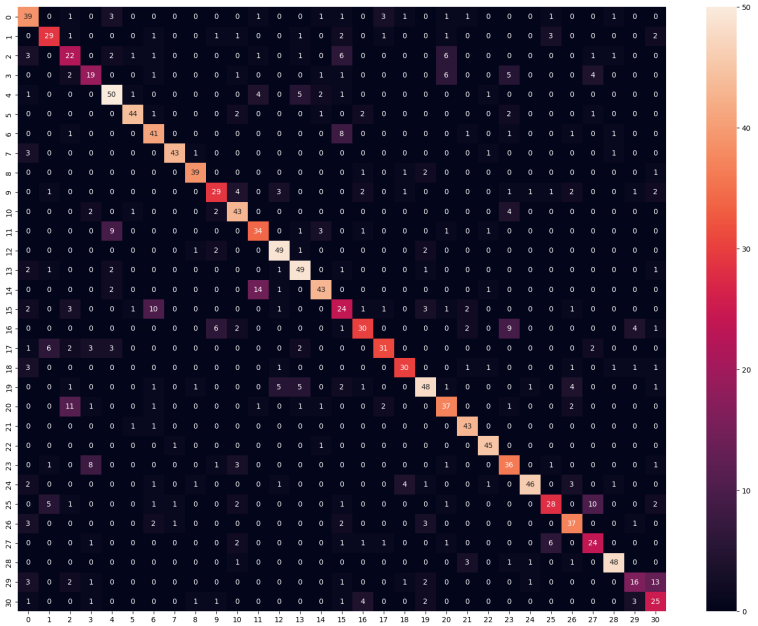
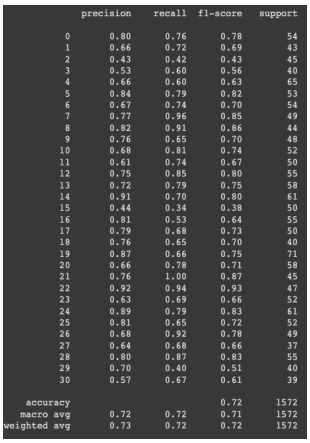


Figure 22: Mean confusion matrix for CNN model on AASL dataset (50 epochs)



	precision	recall	f1-score	support
0	0.80	0.76	0.78	54
1	0.66	0.72	0.69	43
2	0.43	0.42	0.43	45
3	0.53	0.60	0.56	40
4	0.66	0.60	0.63	65
5	0.84	0.79	0.82	53
6	0.67	0.74	0.70	54
7	0.77	0.96	0.85	49
8	0.82	0.91	0.86	44
9	0.76	0.65	0.70	48
10	0.68	0.81	0.74	52
11	0.61	0.74	0.67	50
12	0.75	0.85	0.80	55
13	0.72	0.79	0.75	58
14	0.91	0.70	0.80	61
15	0.44	0.34	0.38	50
16	0.81	0.53	0.64	55
17	0.79	0.68	0.73	50
18	0.76	0.65	0.70	40
19	0.87	0.66	0.75	71
20	0.66	0.78	0.71	58
21	0.76	1.00	0.87	45
22	0.92	0.94	0.93	47
23	0.63	0.69	0.66	52
24	0.89	0.79	0.83	61
25	0.81	0.65	0.72	52
26	0.68	0.92	0.78	49
27	0.64	0.68	0.66	37
28	0.80	0.87	0.83	55
29	0.70	0.40	0.51	40
30	0.57	0.67	0.61	39
accuracy			0.72	1572
macro avg	0.72	0.72	0.71	1572
weighted avg	0.73	0.72	0.72	1572

Figure 23: Classification report for CNN model on AASL dataset (50 epochs) from best trial

5.2.2 Pre-trained ResNet50V2 Model

Figure 24 shows the training results of the pre-trained ResNet50V2 model on the Arabic, or AASL, dataset after 50 epochs for 5 trials. The average training accuracy is 85.31%, the average validation accuracy is 79.52%, and the average test accuracy is 78.77%. The standard errors are 0.2500, 0.2971, and 0.3261, respectively.

Trial #	Train Accuracy	Validation Accuracy	Test Accuracy
1	84.85	78.37	79.52
2	86.07	79.99	78.44
3	85.55	79.64	78.05
4	85.39	79.64	78.24
5	84.68	79.96	79.58
Avg	**85.308**	**79.52**	**78.766**
Std Err	0.2500	0.2971	0.3261

Figure 24: Results from training pre-trained model on AASL dataset

Figure 25 shows the training accuracy and loss curve from training the pre-trained model for 50 epochs. These curves are generated from the best trial.

Figure 26 shows the mean confusion matrix from 5 trials and Figure 27 shows the classification report from the best trial of the pre-trained ResNet50V2 model on the AASl dataset.

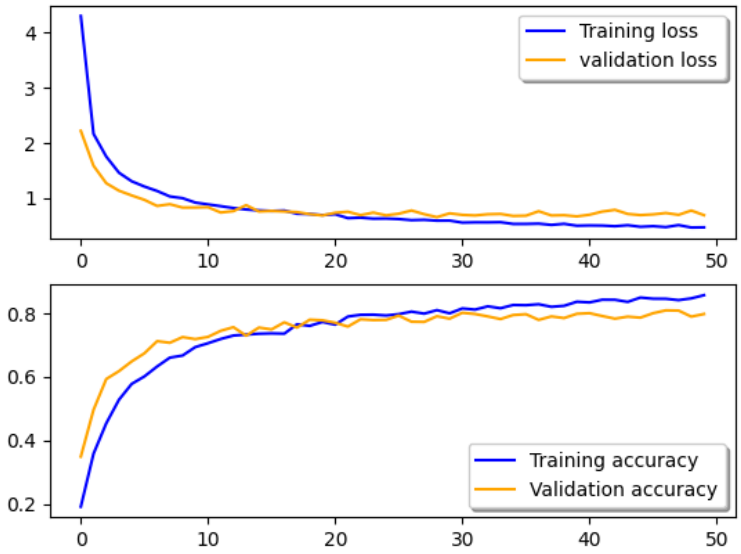


Figure 25: Training accuracy and loss curve from pre-trained model on AASL dataset

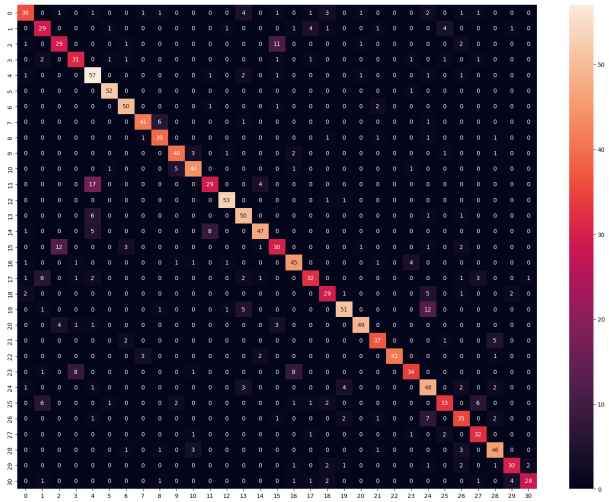
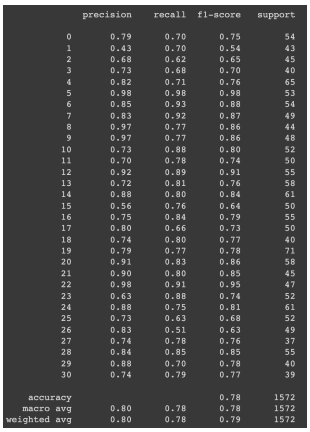


Figure 26: ResNet50V2 mean confusion matrix on AASL dataset



	precision	recall	f1-score	support
0	0.79	0.70	0.75	54
1	0.43	0.70	0.54	43
2	0.68	0.62	0.65	45
3	0.73	0.68	0.70	40
4	0.82	0.71	0.76	65
5	0.98	0.98	0.98	53
6	0.85	0.93	0.88	54
7	0.83	0.92	0.87	49
8	0.97	0.77	0.86	44
9	0.97	0.77	0.86	48
10	0.73	0.88	0.80	52
11	0.70	0.78	0.74	50
12	0.92	0.89	0.91	55
13	0.72	0.81	0.76	58
14	0.88	0.80	0.84	61
15	0.56	0.76	0.64	50
16	0.75	0.84	0.79	55
17	0.80	0.66	0.73	50
18	0.74	0.80	0.77	40
19	0.79	0.77	0.78	71
20	0.91	0.83	0.86	58
21	0.90	0.80	0.85	45
22	0.98	0.91	0.95	47
23	0.63	0.88	0.74	52
24	0.88	0.75	0.81	61
25	0.73	0.63	0.68	52
26	0.83	0.51	0.63	49
27	0.74	0.78	0.76	37
28	0.84	0.85	0.85	55
29	0.88	0.70	0.78	40
30	0.74	0.79	0.77	39
accuracy			0.78	1572
macro avg	0.80	0.78	0.78	1572
weighted avg	0.80	0.78	0.79	1572

Figure 27: ResNet50V2 best classification report on AASL dataset

6 Discussion of Results and Comparison

For the American Sign Language (ASL) Alphabet dataset, both models achieve a high test accuracy of above 99%. Without much hyperparameter optimization and the need for complex layers, the baseline CNN model performs slightly better than the pre-trained ResNet50V2 model, with an average of 99.73% as compared to 99.67%. Since we are using pre-trained weights, it is quite difficult to optimize the hyperparameters for the pre-trained model to match up with the data from the dataset, especially since data augmentation actually worsens the model's performance in this case. Thus, these results suggest that more hyperparameter optimization is required to achieve better performance for the pre-trained model. However, when comparing the learning curves between the 2 models (Figures 11 and 15), the curves show that the pre-trained model quickly generalizes and learns patterns from the data as compared to the baseline CNN model.

For the Arabic Alphabet Sign Language (AASL) dataset, the pre-trained model performs significantly better in the first 50 epochs, with an average test accuracy of 78.77% as compared to 69.95% by the baseline CNN model. The baseline model underfits with 50 epochs and requires more training data and time to improve its performance to 74.98%. This shows the generalizability of the pre-trained model as compared to our baseline CNN model. However, these accuracy scores are still not as good as expected, given the pre-trained model. Thus, either more data augmentation or more training time is required to achieve an accuracy of higher than 80% for the pre-trained model. However, due to the limited computing power, we were not able to train this model for more than 50 epochs.

7 Conclusions

The project compares the performance of a baseline CNN model against a pre-trained model on 2 image datasets, a large and a small, but challenging dataset. The results show the ability of simple convolutional networks to achieve high accuracy on simple datasets, but imply the need for more training data and time to perform and generalize well on more diverse and unseen data.

References

[1] Zhang, J., Hu, F., Li, L., Xu, X., Yang, Z., Chen, Y. (2018). An adaptive mechanism to achieve learning rate dynamically. Neural Computing and Applications, 31(10), 6685–6698. https://doi.org/10.1007/s00521-018-3495-0.