SignWave: Neural Network Deep Learning for American Sign Language Classification

A Computer Vision and Machine Learning Project with TensorFlow and Keras

American Sign Language (ASL) Introduction

- ASL is a **visual-spatial language** used in the deaf communities of the U.S. and Canada (Clemson University, 2024).
- About **11 million** Americans were deaf or had serious difficulty hearing in 2021 (National Deaf Center, 2024).
- ASL course enrollment grew by **6,583%** from 1990 to 2016, making it the 3rd most-studied language on U.S. college campuses (Clemson University, 2024).

ASL Alphabet Gestures.
Source: pocketsign.org

SignWave ASL Classification Project Overview

Dataset

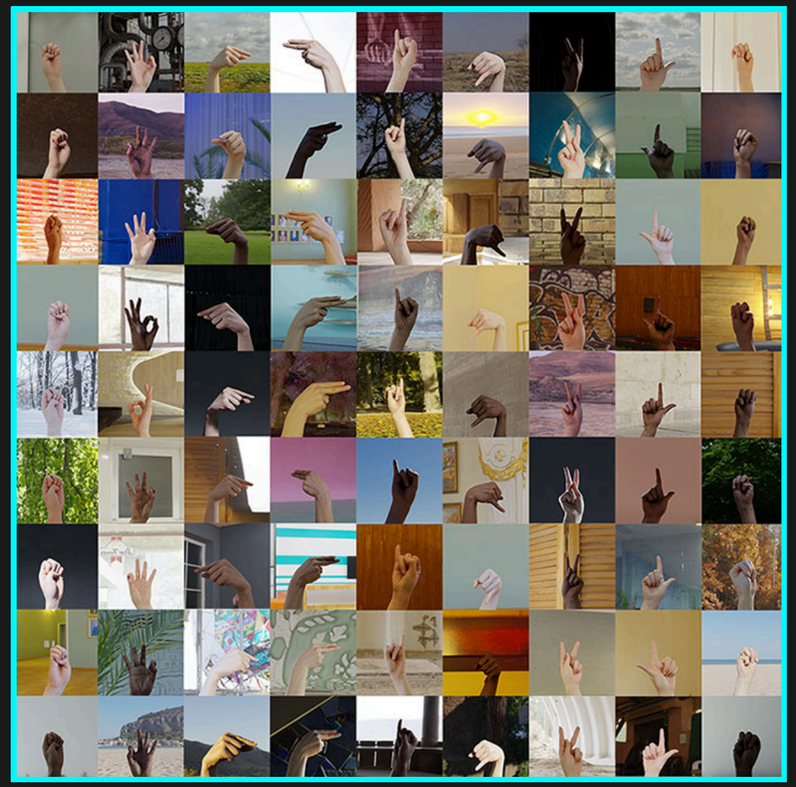
- Synthetic ASL Alphabet & ASL Numbers by Lexset
- 9.97 GB initial storage required
- 36,000 total 512 x 512 images for 36 classes (A-Z, 1-10)
    - 900 images per training class
    - 100 images per validation class

Preprocessing

Removed background with salient object detection

Model Architecture

- Deep learning convolutional neural network (CNN)
- Sequentially layered TensorFlow model using Keras
- Batch size of 25 images per training step
- Rescaled image size to 50 x 50 pixels



Synthetic ASL Alphabet Collage.
Source: Lexset via Kaggle

SignWave ASL Classification Project Requirements

Programming Language
Python 3.11.2

Deep Learning Framework
TensorFlow 2.16.1

Machine Specifications

- Google Cloud E2 virtual machine
- 16 GB RAM, 4 virtual CPUs
- 20 GB additional disk storage

Library Installations and Uses

- Matplotlib 3.9.0: Line graph visualizations
- NumPy 1.26.2: Numerical operations
- Pandas 2.1.4: Table DataFrame organization
- Rembg 2.0.57: Background removal
- Scikit-learn 1.5.1: Confusion matrix computation
- Seaborn 0.13.2: Confusion matrix visualization

python™

TensorFlow

Compute Engine

matplotlib

NumPy

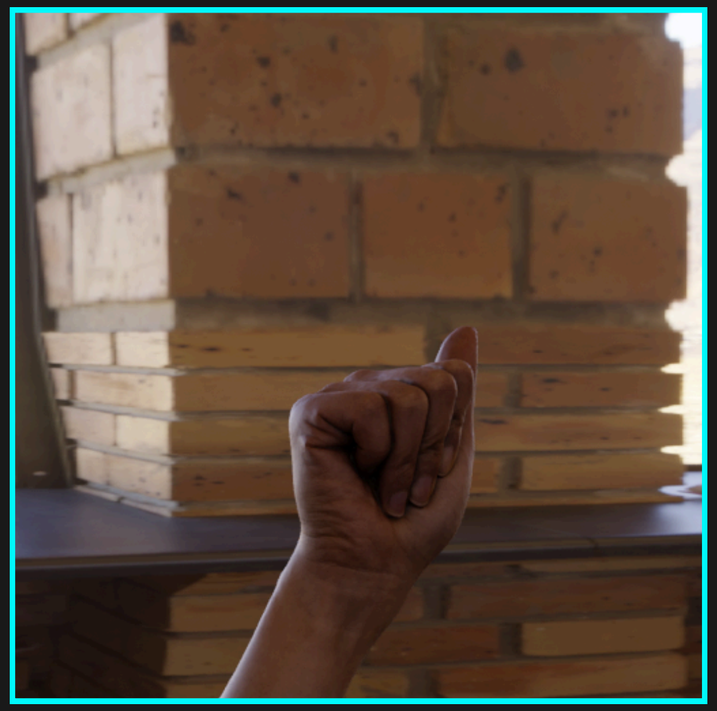
scikit learn

pandas

seaborn

SignWave ASL Letter Hand Gesture Preprocessing

Using Salient Object Detection to Remove Background Noise



Original Image: ASL Letter A

512 x 512 - 244 KB

File size reduced
by over 77%

Cropped Image: ASL Letter A

512 x 512 - 54 KB

SignWave ABC Convolutional Neural Network Layers

TensorFlow Keras Model Architecture for ASL Image Classification: Letters A-Z

PNG image cropped with salient object detection

1
Input

2
Conv2D
1,184 parameters

3
MaxPooling2D

4
Dropout

5
Conv2D
18,496 parameters

6
MaxPooling2D

7
Flatten

8
Dense
991,360 parameters

9
Dropout

10
Dense
3,354 parameters

1,014,394
Trainable
Parameters

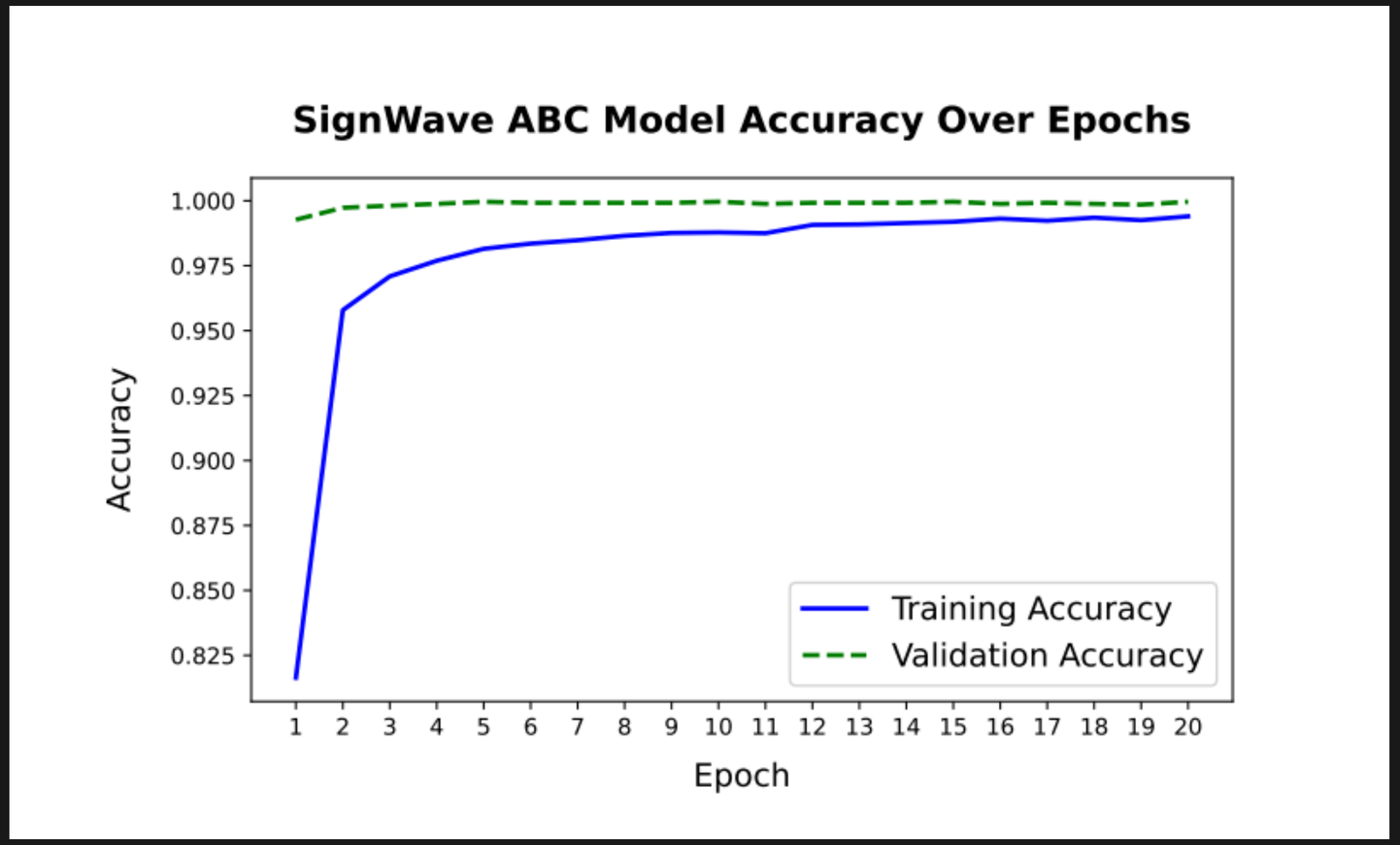
SignWave ABC Convolutional Neural Network Layers

TensorFlow Keras Model Architecture for ASL Image Classification: Letters A-Z

Layer Type	Output Shape	Parameters Calculation	Description
Input	(None, 50, 50, 4)	No parameters	Rescales image size to 50x50 and specifies 4 channels for RGBA.
Conv2D (32 units)	(None, 48, 48, 32)	$(3 * 3 * 4 + 1) * 32 = 1{,}184$	Uses 32 filters, kernel size 3x3, 4 input channels, applies ReLU.
MaxPooling2D	(None, 24, 24, 32)	No parameters	Splits each spatial dimension from previous Conv2D in half.
Dropout	(None, 24, 24, 32)	No parameters	Randomly drops 25% of units to prevent overfitting.
Conv2D (64 units)	(None, 22, 22, 64)	$(3 * 3 * 32 + 1) * 64 = 18{,}496$	Uses 64 filters, kernel size 3x3, 32 input channels, applies ReLU.
MaxPooling2D	(None, 11, 11, 64)	No parameters	Splits each spatial dimension from previous Conv2D in half.
Flatten	(None, 7744)	No parameters	Converts MaxPooling2D output to a single vector of size 7744.
Dense (128 units)	(None, 128)	$(7744 + 1) * 128 = 991{,}360$	Uses 128 units, input size of 7744, applies ReLU.
Dropout	(None, 128)	No parameters	Randomly drops 50% of units to prevent overfitting.
Dense (26 units)	(None, 26)	$(128 + 1) * 26 = 3{,}354$	Uses 26 units for classes A-Z, input size of 128, applies ReLU.
Total Parameters	–	1,014,394	Total number of parameters used to train the model.

SignWave ABC Training and Validation Accuracy Curves

Using Automated Data Plotting with Pandas, Matplotlib, and Math Libraries

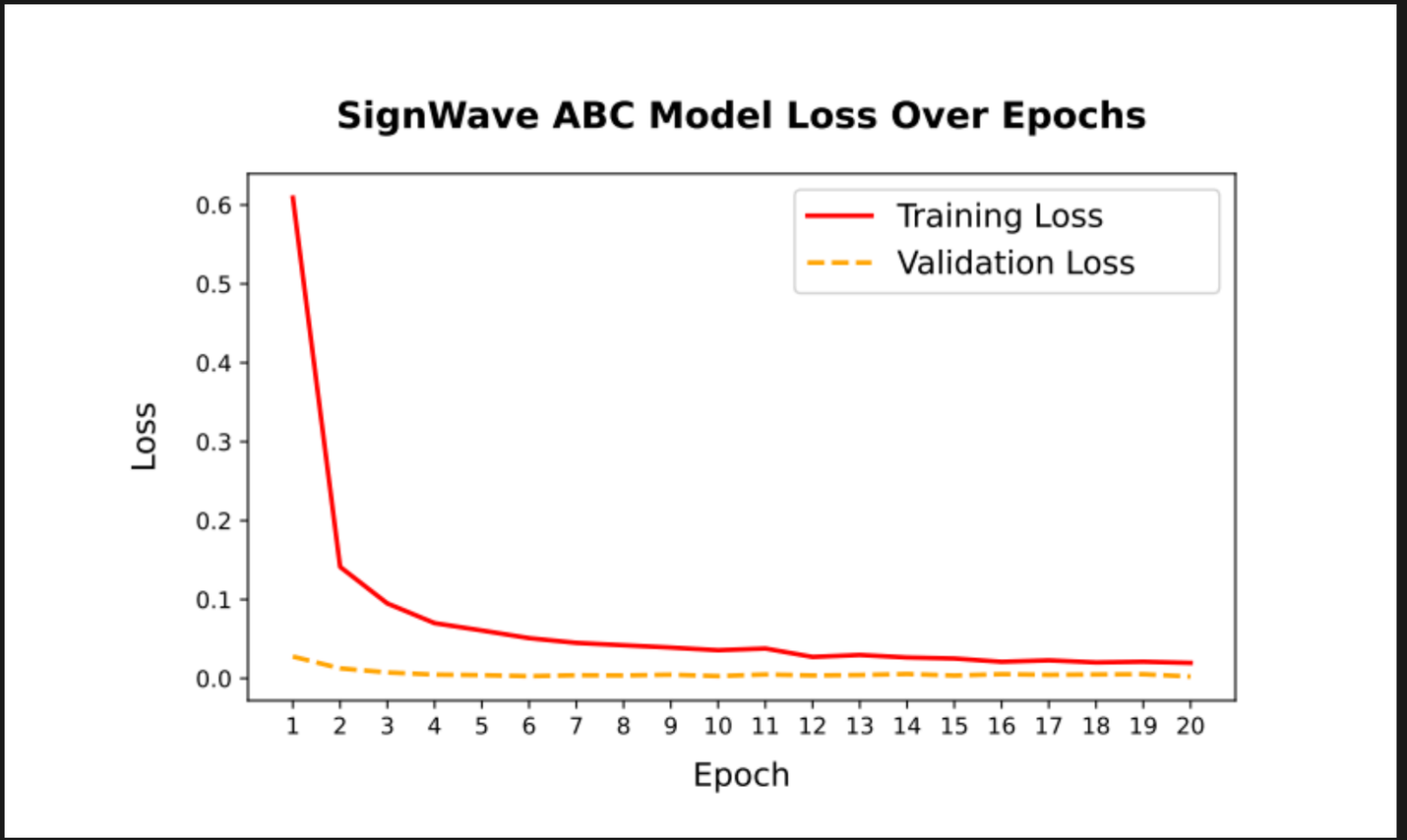


Delivered **best-in-class** accuracy among 1-million parameter ASL letter classification models.

Achieved 99.96% validation accuracy in 5 epochs.

SignWave ABC Training and Validation Loss Curves

Using Automated Data Plotting with Pandas, Matplotlib, and Math Libraries



Provided **best-in-class** loss among 1-million parameter ASL letter classification models.

Achieved 0.0023 cross-entropy validation loss in 20 epochs.

SignWave ABC Results

Training Accuracy
99.40% (23,260/23,400 images)

Validation Accuracy
99.96% (2599/2600 images)

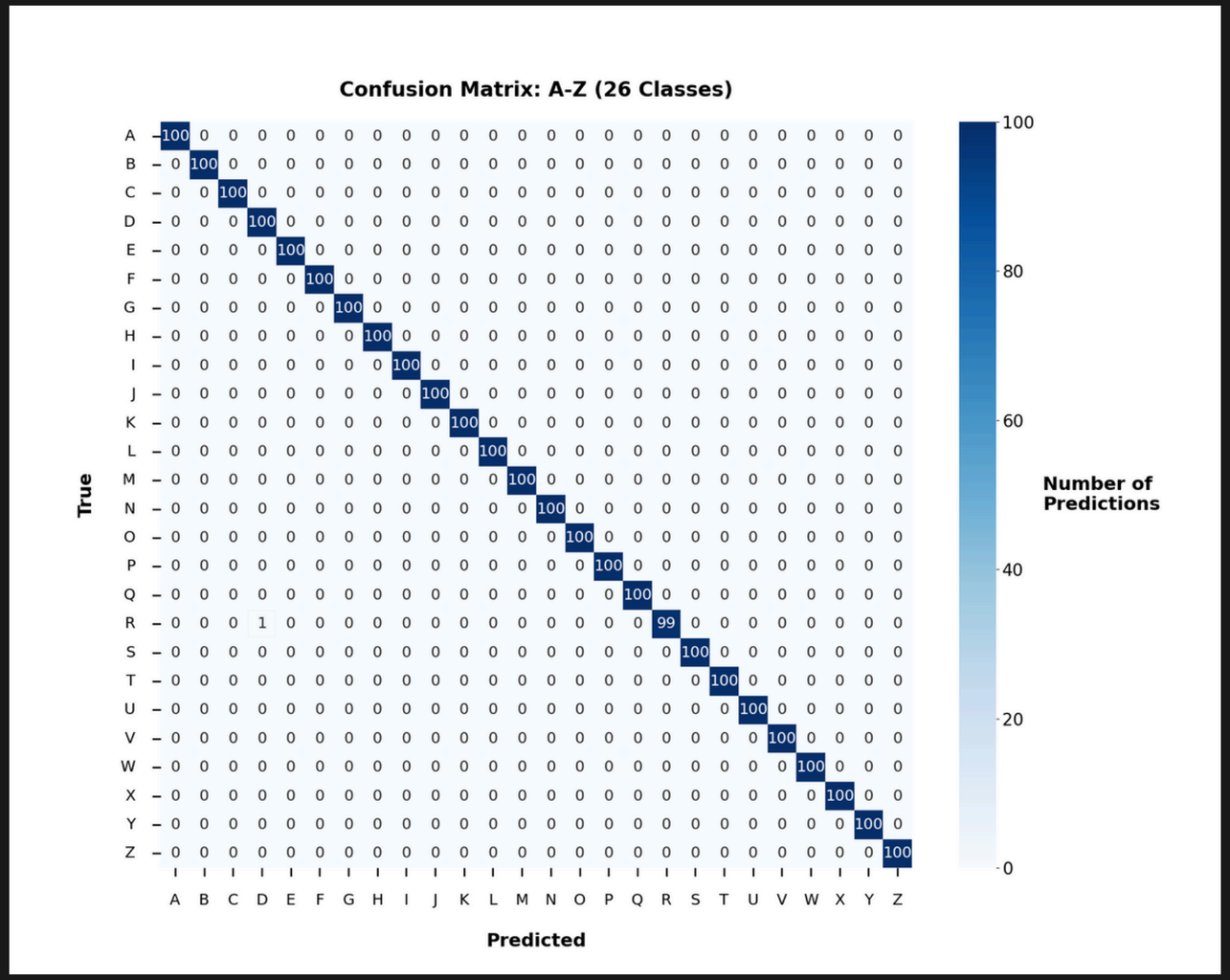
Training Loss
0.0196

Validation Loss
0.0023

Total Training and Validation Time
36 min, 21 sec

Epoch	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	0.8164	0.6092	0.9927	0.0277
2	0.9579	0.1413	0.9973	0.0127
3	0.9709	0.0950	0.9981	0.0075
4	0.9769	0.0700	0.9988	0.0048
5	0.9815	0.0607	0.9996	0.0041
6	0.9835	0.0510	0.9992	0.0028
7	0.9848	0.0450	0.9992	0.0040
8	0.9865	0.0420	0.9992	0.0037
9	0.9876	0.0391	0.9992	0.0047
10	0.9878	0.0357	0.9996	0.0029
11	0.9875	0.0379	0.9988	0.0050
12	0.9907	0.0272	0.9992	0.0037
13	0.9909	0.0297	0.9992	0.0043
14	0.9914	0.0265	0.9992	0.0056
15	0.9919	0.0252	0.9996	0.0036
16	0.9931	0.0210	0.9988	0.0054
17	0.9923	0.0229	0.9992	0.0045
18	0.9935	0.0202	0.9988	0.0050
19	0.9925	0.0211	0.9985	0.0053
20	0.9940	0.0196	0.9996	0.0023

SignWave ABC Confusion Matrix



A confusion matrix displays model prediction performance by class.

Perfect evaluation predictions for all letters except letter R (99/100).

2,599/2,600 overall evaluation accuracy.

SignWave 123 Convolutional Neural Network Layers

TensorFlow Keras Model Architecture for ASL Image Classification: Numbers 1-10

PNG image cropped with salient object detection

1
Input

2
Conv2D
1,184 parameters

3
MaxPooling2D

4
Dropout

5
Conv2D
18,496 parameters

6
MaxPooling2D

7
Flatten

8
Dense
991,360 parameters

9
Dropout

10
Dense
1,290 parameters

1,012,330
Trainable
Parameters

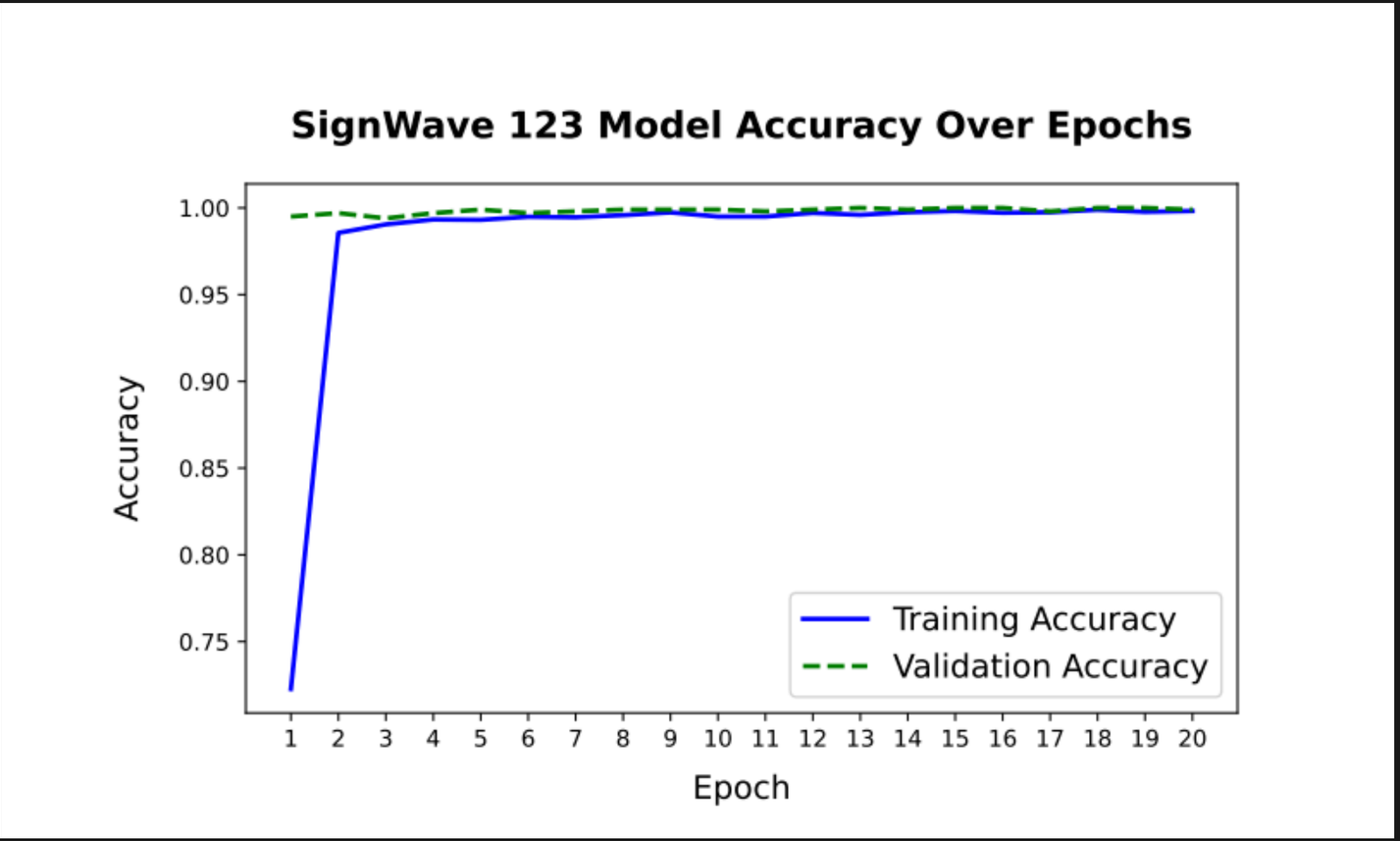
SignWave 123 Convolutional Neural Network Layers

TensorFlow Keras Model Architecture for ASL Image Classification: Numbers 1-10

Layer Type	Output Shape	Parameters Calculation	Description
Input	(None, 50, 50, 4)	No parameters	Rescales image size to 50x50 and specifies 4 channels for RGBA.
Conv2D (32 units)	(None, 48, 48, 32)	$(3 * 3 * 4 + 1) * 32 = 1{,}184$	Uses 32 filters, kernel size 3x3, 4 input channels, applies ReLU.
MaxPooling2D	(None, 24, 24, 32)	No parameters	Splits each spatial dimension from previous Conv2D in half.
Dropout	(None, 24, 24, 32)	No parameters	Randomly drops 25% of units to prevent overfitting.
Conv2D (64 units)	(None, 22, 22, 64)	$(3 * 3 * 32 + 1) * 64 = 18{,}496$	Uses 64 filters, kernel size 3x3, 32 input channels, applies ReLU.
MaxPooling2D	(None, 11, 11, 64)	No parameters	Splits each spatial dimension from previous Conv2D in half.
Flatten	(None, 7744)	No parameters	Converts MaxPooling2D output to a single vector of size 7744.
Dense (128 units)	(None, 128)	$(7744 + 1) * 128 = 991{,}360$	Uses 128 units, takes an input size of 7744, applies ReLU.
Dropout	(None, 128)	No parameters	Randomly drops 50% of units to prevent overfitting.
Dense (10 units)	(None, 10)	$(128 + 1) * 10 = 1{,}290$	Uses 10 units for classes 1-10, takes input size of 128, applies ReLU.
Total Parameters	–	1,012,330	Total number of parameters used to train the model.

SignWave 123 Training and Validation Accuracy Curves

Using Automated Data Plotting with Pandas, Matplotlib, and Math Libraries

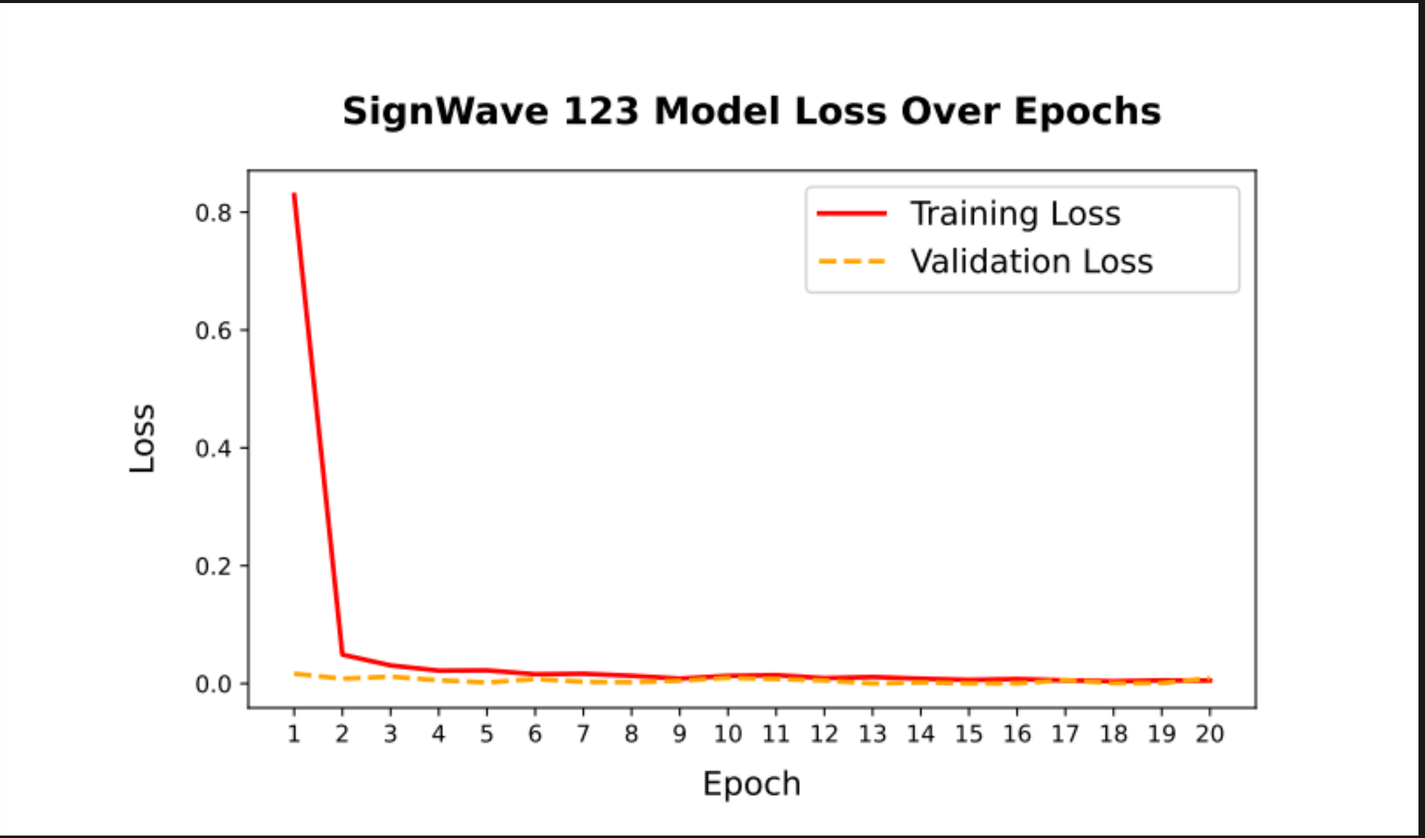


Achieved 100% validation accuracy in 13 epochs.

Delivered **best-in-class** accuracy among 1-million parameter ASL number classification models.

SignWave 123 Training and Validation Loss Curves

Using Automated Data Plotting with Pandas, Matplotlib, and Math Libraries



Provided **best-in-class** loss among 1-million parameter ASL number classification models.

Achieved 0.0002 cross-entropy validation loss in 13 epochs.

SignWave 123 Results

Training Accuracy
99.60% (8,964/9,000 images)

Validation Accuracy
100% (1,000/1,000 images)

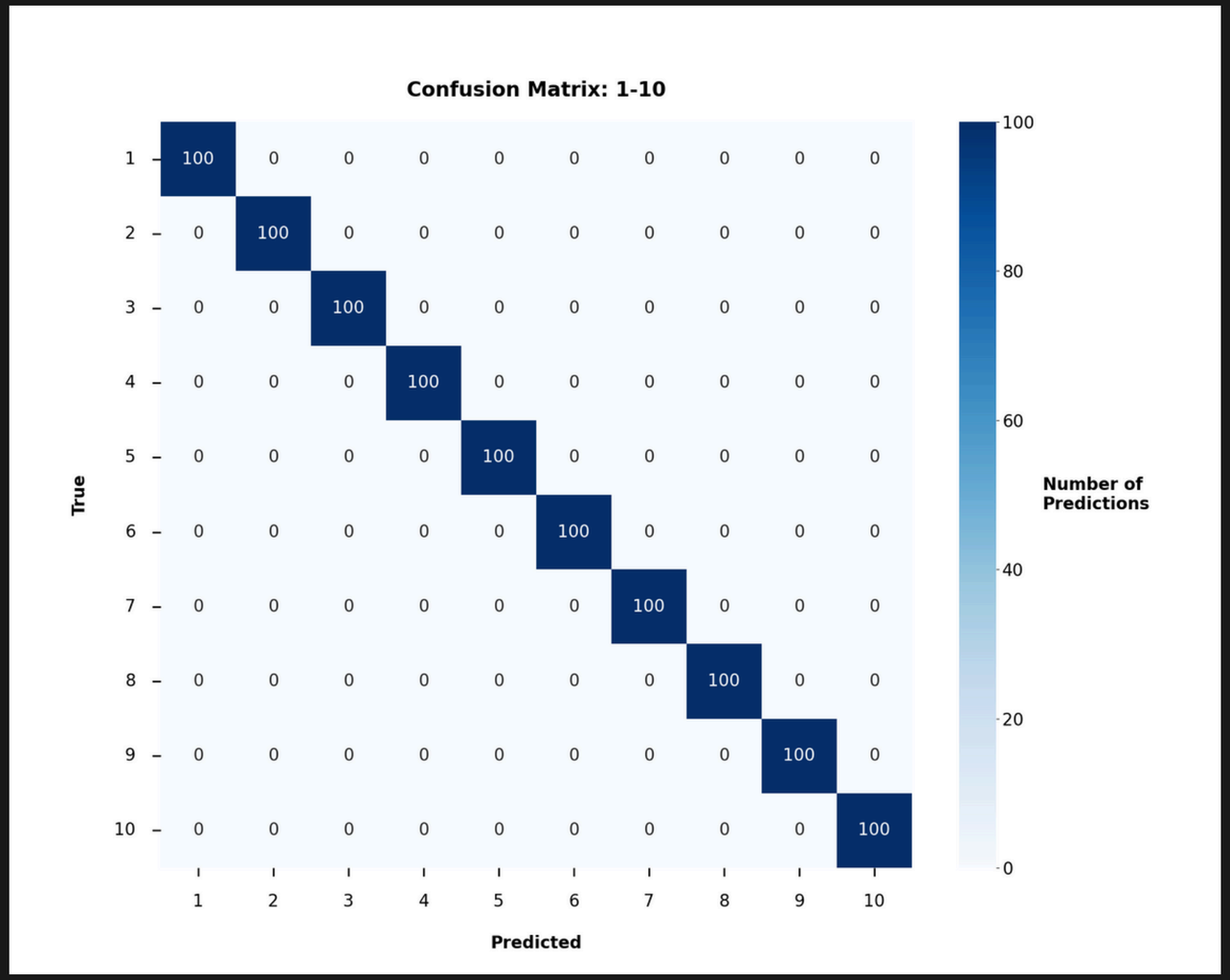
Training Loss
0.0111

Validation Loss
0.0002

Total Training and Validation Time
24 min, 51 sec

Epoch	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	0.7227	0.8292	0.9950	0.0168
2	0.9856	0.0491	0.9970	0.0082
3	0.9905	0.0308	0.9940	0.0117
4	0.9933	0.0219	0.9970	0.0056
5	0.9931	0.0224	0.9990	0.0018
6	0.9949	0.0158	0.9970	0.0076
7	0.9946	0.0167	0.9980	0.0029
8	0.9958	0.0132	0.9990	0.0021
9	0.9974	0.0084	0.9990	0.0047
10	0.9950	0.0131	0.9990	0.0096
11	0.9950	0.0143	0.9980	0.0076
12	0.9971	0.0092	0.9990	0.0052
13	0.9960	0.0111	1.000	0.0002
14	0.9975	0.0083	0.9990	0.0018
15	0.9982	0.0062	1.000	0.0003
16	0.9972	0.0077	1.000	0.0003
17	0.9975	0.0054	0.9980	0.0057
18	0.9989	0.0039	1.000	0.0006
19	0.9977	0.0054	1.000	0.0008
20	0.9982	0.0051	0.9990	0.0091

SignWave 123 Confusion Matrix



Perfect evaluation predictions for numbers 1-10.

1,000/1,000 overall evaluation accuracy.

SignWave ABC-123 Convolutional Neural Network Layers

TensorFlow Keras Model Architecture for ASL Image Classification: Letters A-Z & Numbers 1-10

PNG image cropped with salient object detection

1 **Input**

2 **Conv2D** 1,184 parameters

3 **Conv2D** 9,248 parameters

4 **MaxPooling2D**

5 **Dropout**

6 **Conv2D** 18,496 parameters

7 **Conv2D** 36,928 parameters

8 **MaxPooling2D**

9 **Dropout**

10 **Flatten**

11 **Dense** 1,982,720 parameters

12 **Dropout**

13 **Dense** 32,896 parameters

14 **Dropout**

15 **Dense** 4,644 parameters

2,086,116 Trainable Parameters

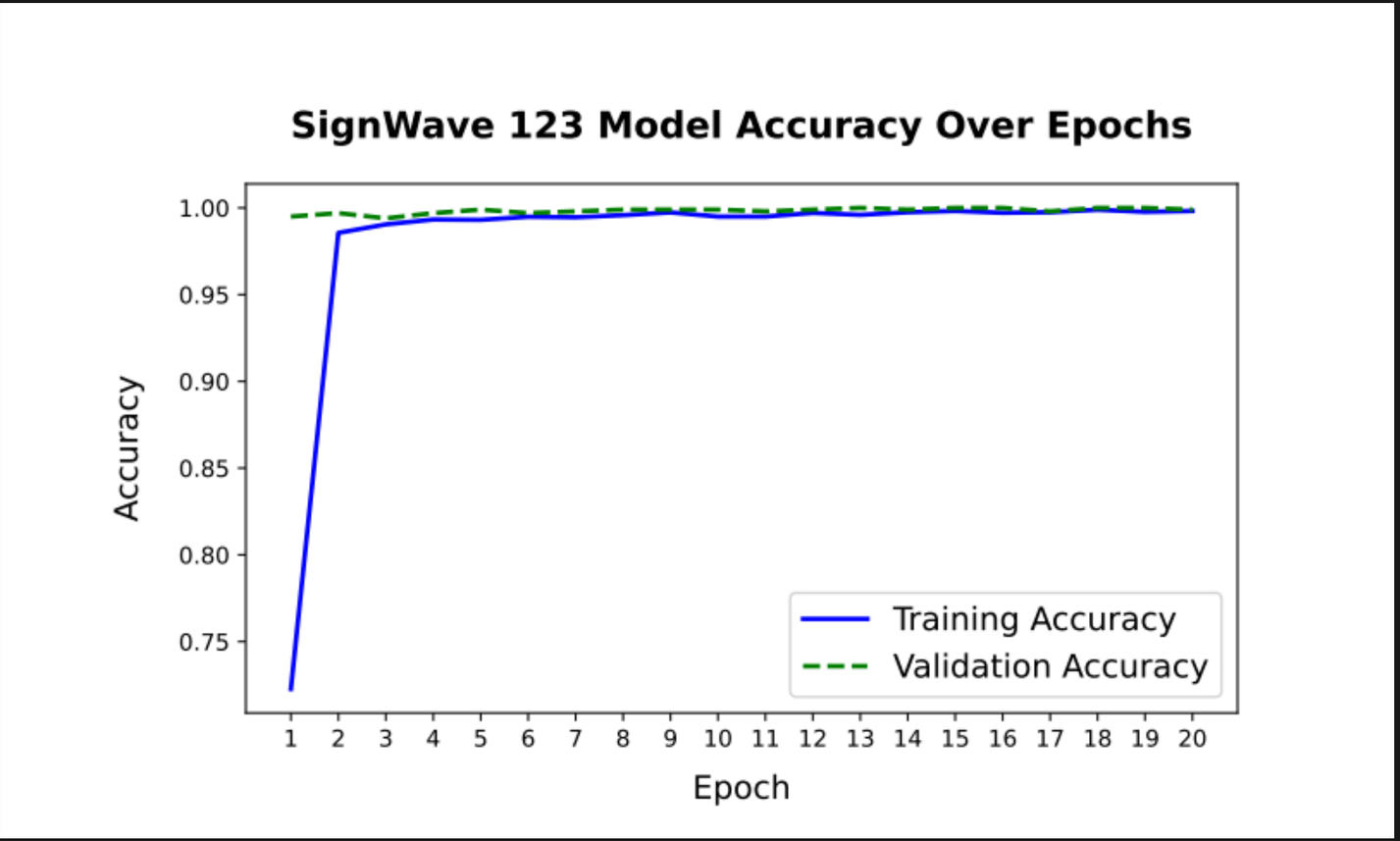
SignWave ABC-123 Convolutional Neural Network Layers

TensorFlow Keras Model Architecture for ASL Image Classification: Letters A-Z & Numbers 1-10

Layer Type	Output Shape	Parameters Calculation	Description
Input	(None, 50, 50, 4)	No parameters	Rescales image size to 50x50 and specifies 4 channels for RGBA.
Conv2D (32 units)	(None, 50, 50, 32)	(3 * 3 * 4 + 1) * 32 = 1,184	Uses 32 filters, kernel size 3x3, 4 input channels, applies ReLU and padding.
Conv2D (32 units)	(None, 48, 48, 32)	(3 * 3 * 32 + 1) * 32 = 9,248	Uses 32 filters, kernel size 3x3, 32 input channels, applies ReLU.
MaxPooling2D	(None, 24, 24, 32)	No parameters	Splits each spatial dimension from previous Conv2D in half.
Dropout	(None, 24, 24, 32)	No parameters	Randomly drops 20% of units to prevent overfitting.
Conv2D (64 units)	(None, 24, 24, 64)	(3 * 3 * 32 + 1) * 64 = 18,496	Uses 64 filters, kernel size 3x3, 32 input channels, applies ReLU and padding.
Conv2D (64 units)	(None, 22, 22, 64)	(3 * 3 * 64 + 1) * 64 = 36,928	Uses 64 filters, kernel size 3x3, 64 input channels, applies ReLU.
MaxPooling2D	(None, 11, 11, 64)	No parameters	Splits each spatial dimension from previous Conv2D in half.
Dropout	(None, 11, 11, 64)	No parameters	Randomly drops 20% of units to prevent overfitting.
Flatten	(None, 7744)	No parameters	Converts MaxPooling2D output to a single vector of size 7744.
Dense (256 units)	(None, 256)	(7744 + 1) * 256 = 1,982,720	Uses 256 units, takes an input size of 7744, applies ReLU.
Dropout	(None, 256)	No parameters	Randomly drops 40% of units to prevent overfitting.
Dense (128 units)	(None, 128)	(256 + 1) * 128 = 32,896	Uses 128 units, takes an input size of 256, applies ReLU.
Dropout	(None, 128)	No parameters	Randomly drops 40% of units to prevent overfitting.
Dense (36 units)	(None, 36)	(128 + 1) * 36 = 4,644	Uses 36 units for classes A-Z and 1-10, takes an input size of 128.
Total Parameters	-	2,086,116	Total number of parameters used to train the model.

SignWave ABC-123 Training and Validation Accuracy Curves

Using Automated Data Plotting with Pandas, Matplotlib, and Math Libraries

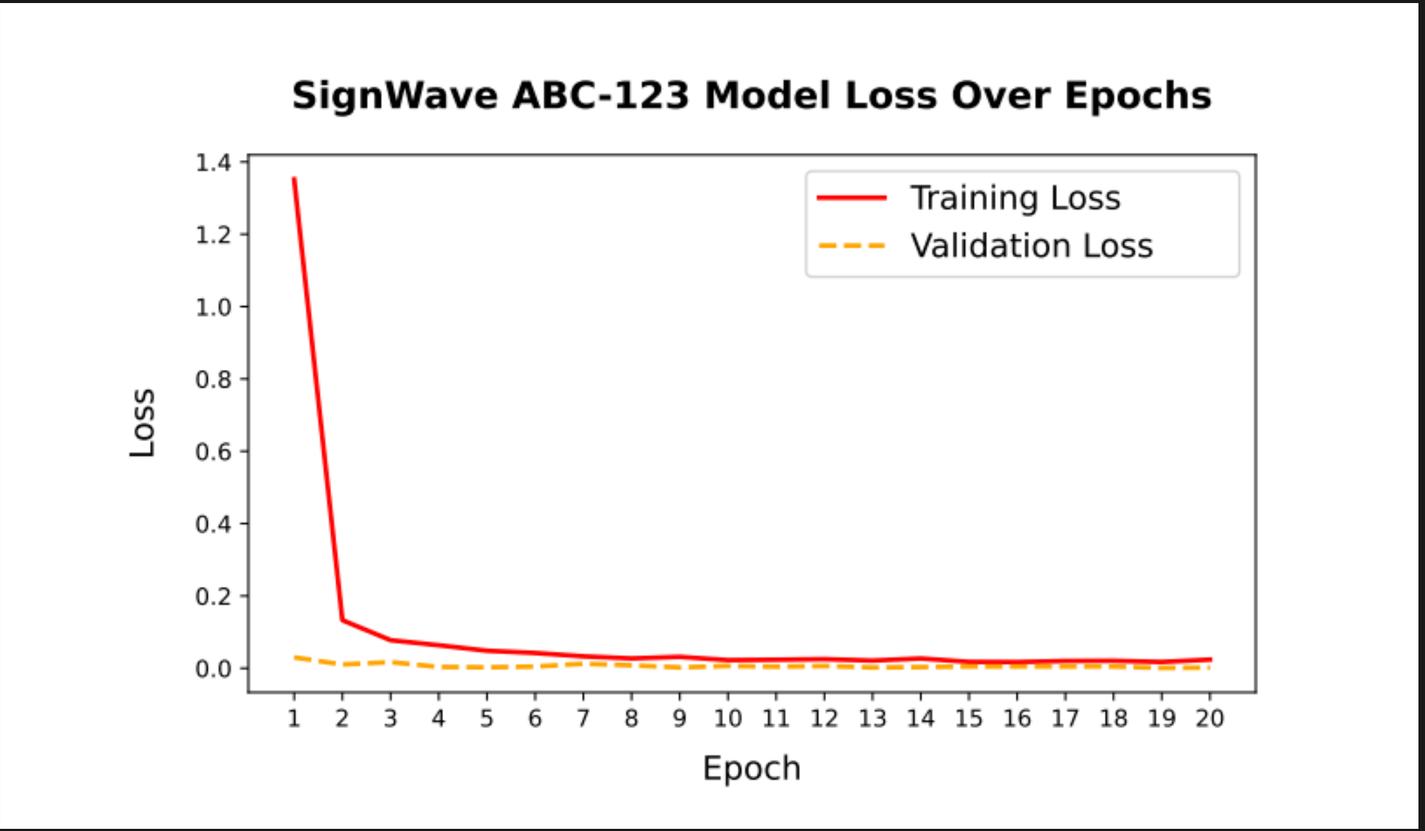


Achieved 99.97% validation accuracy in 19 epochs.

Delivered **best-in-class** accuracy among 2-million parameter ASL alpha-numeric classification models.

SignWave ABC-123 Training and Validation Loss Curves

Using Automated Data Plotting with Pandas, Matplotlib, and Math Libraries



Provided **best-in-class** loss among 2-million parameter ASL alpha-numeric classification models.

Achieved 0.0009 cross-entropy validation loss in under 20 epochs.

SignWave ABC-123 Results

Training Accuracy
99.54% (32,251/32,400 images)

Validation Accuracy
99.97% (3,599/3,600 images)

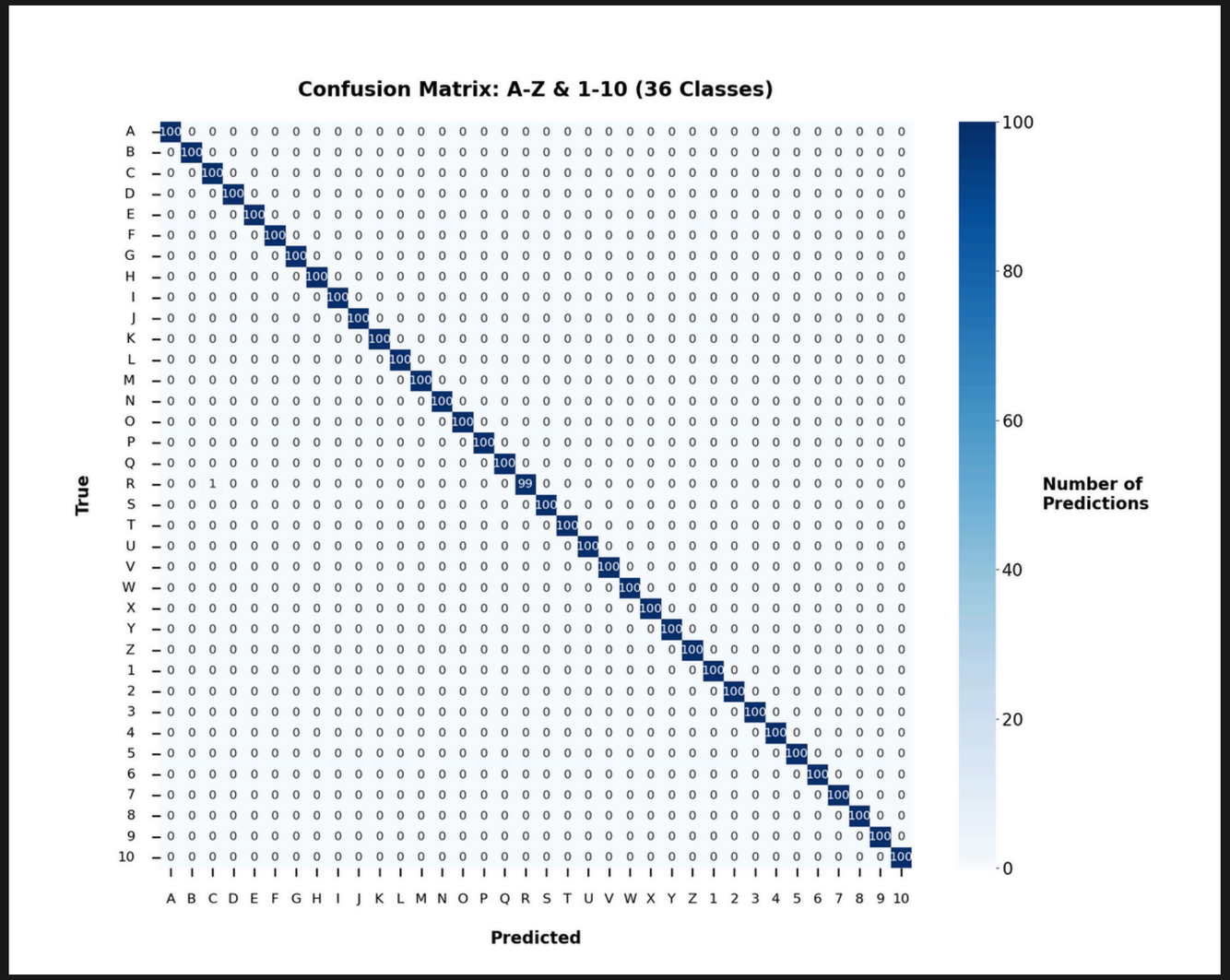
Training Loss
0.0174

Validation Loss
0.0009

Total Training and Validation Time
83 min, 9 sec

Epoch	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss
1	0.613	1.3518	0.99	0.0301
2	0.961	0.1329	0.9969	0.0103
3	0.9756	0.0776	0.9944	0.0169
4	0.9809	0.0636	0.9989	0.0036
5	0.9867	0.0484	0.9992	0.0027
6	0.9881	0.0422	0.9986	0.0046
7	0.9906	0.0329	0.9983	0.0125
8	0.992	0.0272	0.9983	0.0081
9	0.9904	0.0315	0.9992	0.0023
10	0.9929	0.0225	0.9986	0.0063
11	0.9933	0.0236	0.9981	0.0043
12	0.9925	0.0252	0.9983	0.0062
13	0.9944	0.0212	0.9989	0.0022
14	0.9924	0.0271	0.9992	0.0032
15	0.9947	0.0176	0.9986	0.0049
16	0.9955	0.0171	0.9981	0.005
17	0.9946	0.0201	0.9983	0.0048
18	0.9944	0.0206	0.9983	0.0049
19	0.9954	0.0174	0.9997	0.0009
20	0.9950	0.0239	0.9992	0.0019

SignWave ABC-123 Confusion Matrix



Perfect evaluation predictions for all classes except letter R (99/100).

3,599/3,600 overall evaluation accuracy.

SignWave Model Benchmark Comparison

Model	Validation Accuracy (%)	# of Parameters	Training Time	# of Classes	Initial Image Dataset Size	Final Image Dataset Size	Training Images per Class	Validation Images per Class	Total # of Training Images	Total # of Validation Images	Image Preprocessing
SignWave ABC (July 2024)	99.96	1,014,394	36 min, 21 sec	26 (A-Z)	26,000	26,000	900	100	23,400	2,600	Background removal
SignWave 123 (July 2024)	100	1,012,459	36 min, 29 sec	10 (1-10)	9,900	9,900	900	100	9,000	900	Background removal
SignWave ABC-123 (July 2024)	99.97	2,086,116	83 min, 9 sec	36 (A-Z, 1-10)	36,000	36,000	900	100	32,400	3,600	Background removal
Augmentation CNN Model (April 2022)	99.89	2,029,470	55 min, 46 sec	29 (A-Z, Space, Delete, Nothing)	87,000	108,627	~ 2,247	~ 749	65,183	21,721	Size reduction
Mediapipe CNN Model (May 2023)	99.95	Not specified	Not specified	26 (A-Z)	166,000	117,000	3,600	900	93,600	23,400	Not specified

SignWave ASL Classification Project References

Google Cloud Compute Engine
https://cloud.google.com/products/compute

Python
https://www.python.org

Matplotlib
https://matplotlib.org

NumPy
https://numpy.org

Pandas
https://pandas.pydata.org

Pocket Sign
https://pocketsign.org

Rembg
https://github.com/danielgatis/rembg

Scitkit-learn
https://scikit-learn.org/

Seaborn
https://seaborn.pydata.org/

TensorFlow
https://www.tensorflow.org

Synthetic ASL Alphabet
https://www.kaggle.com/datasets/lexset/synthetic-asl-alphabet

Synthetic ASL Numbers
https://www.kaggle.com/datasets/lexset/synthetic-asl-numbers

Mediapipe and CNNs for Real-Time ASL Gesture Recognition
https://arxiv.org/pdf/2305.05296

Vision-Based American Sign Language Classification Approach via Deep Learning
https://arxiv.org/pdf/2204.04235

SignWave

A mAInstream studIOs project.

Forrest Moulin, B.S. IST
https://linkedin.com/in/moulinf

mAInstream studIOs™
https://mainstreamstudios.ai