Model Card

Model Overview

This model is designed for recognizing American Sign Language (ASL) gestures. It is built on two architectures:

1. CNN: Trained on the ASL Alphabet Dataset for recognizing 29 static gesture classes (A-Z, SPACE, DELETE, NOTHING).

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 32, 64, 64]             896
              ReLU-2          [-1, 32, 64, 64]               0
         MaxPool2d-3          [-1, 32, 32, 32]               0
            Conv2d-4          [-1, 64, 32, 32]          18,496
              ReLU-5          [-1, 64, 32, 32]               0
         MaxPool2d-6          [-1, 64, 16, 16]               0
            Conv2d-7         [-1, 128, 16, 16]          73,856
              ReLU-8         [-1, 128, 16, 16]               0
         MaxPool2d-9           [-1, 128, 8, 8]               0
           Flatten-10                 [-1, 8192]               0
            Linear-11                  [-1, 256]       2,097,408
              ReLU-12                  [-1, 256]               0
            Linear-13                   [-1, 29]           7,453
================================================================
Total params: 2,198,109
Trainable params: 2,198,109
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.05
Forward/backward pass size (MB): 4.00
Params size (MB): 8.39
Estimated Total Size (MB): 12.44
----------------------------------------------------------------
```

2. CNN-LSTM: Trained on a subset of the WLASL Dataset, which focuses on dynamic gesture recognition for selected word-level ASL classes.

```
==========================================================================================================================================
Layer (type:depth-idx)                             Input Shape               Output Shape              Param #
==========================================================================================================================================
CNNLSTM                                            [1, 10, 3, 224, 224]      [1, 29]                   --
├─Sequential: 1-1                                  [10, 3, 224, 224]         [10, 512, 1, 1]           --
│    └─Conv2d: 2-1                                 [10, 3, 224, 224]         [10, 64, 112, 112]        9,408
│    └─BatchNorm2d: 2-2                            [10, 64, 112, 112]        [10, 64, 112, 112]        128
│    └─ReLU: 2-3                                   [10, 64, 112, 112]        [10, 64, 112, 112]        --
│    └─MaxPool2d: 2-4                              [10, 64, 112, 112]        [10, 64, 56, 56]          --
│    └─Sequential: 2-5                             [10, 64, 56, 56]          [10, 64, 56, 56]          --
│    │    └─BasicBlock: 3-1                        [10, 64, 56, 56]          [10, 64, 56, 56]          73,984
│    │    └─BasicBlock: 3-2                        [10, 64, 56, 56]          [10, 64, 56, 56]          73,984
│    └─Sequential: 2-6                             [10, 64, 56, 56]          [10, 128, 28, 28]         --
│    │    └─BasicBlock: 3-3                        [10, 64, 56, 56]          [10, 128, 28, 28]         230,144
│    │    └─BasicBlock: 3-4                        [10, 128, 28, 28]         [10, 128, 28, 28]         295,424
│    └─Sequential: 2-7                             [10, 128, 28, 28]         [10, 256, 14, 14]         --
│    │    └─BasicBlock: 3-5                        [10, 128, 28, 28]         [10, 256, 14, 14]         919,040
│    │    └─BasicBlock: 3-6                        [10, 256, 14, 14]         [10, 256, 14, 14]         1,180,672
│    └─Sequential: 2-8                             [10, 256, 14, 14]         [10, 512, 7, 7]           --
│    │    └─BasicBlock: 3-7                        [10, 256, 14, 14]         [10, 512, 7, 7]           3,673,088
│    │    └─BasicBlock: 3-8                        [10, 512, 7, 7]           [10, 512, 7, 7]           4,720,640
│    └─AdaptiveAvgPool2d: 2-9                      [10, 512, 7, 7]           [10, 512, 1, 1]           --
├─LSTM: 1-2                                        [1, 10, 512]              [1, 10, 256]              1,314,816
├─Linear: 1-3                                      [1, 256]                  [1, 29]                   7,453
==========================================================================================================================================
...
Forward/backward pass size (MB): 397.41
Params size (MB): 50.00
Estimated Total Size (MB): 453.43
==========================================================================================================================================
```

Model Details

1. Model Type: Deep Learning (Image and Video Classification).

2. Framework: PyTorch

3. Architectures:

CNN: Three convolutional layers, followed by fully connected layers.

CNN-LSTM: CNN extracts spatial features from video frames, passed sequentially to an LSTM for temporal modeling.

Datasets

1. **ASL Alphabet Dataset**:

Input: Static RGB images, 64x64 resolution.

Classes: 29 (A-Z, SPACE, DELETE, NOTHING).

Train/Validation/Test Accuracy:

Train: 99.62%, Validation: 99.67%, Test: 92.86%.

2. **WLASL Dataset**:

Input: Video sequences processed as frames.

Selected Classes:

10 Classes: Test Accuracy: 72.73%; 15 Classes: Test Accuracy: 83.76%.

Performance

1. Training Observations:

On the ASL Alphabet dataset, the model achieves near-perfect accuracy on training and validation, with a slight drop on test data.

On the WLASL dataset, performance improves significantly when trained on a larger number of classes (15), showing better generalization and fewer misclassifications.

2. Confusion Matrix Analysis:

For 10 classes, misclassifications occur more frequently due to limited class diversity.

For 15 classes, the model demonstrates stronger performance, with improved diagonal dominance and fewer off-diagonal errors.

Intended Use

1. Designed for real-time ASL gesture recognition applications.

2. Suitable for static gesture recognition (e.g., alphabets) and dynamic gesture recognition (word-level ASL).

3. Can be integrated into accessibility tools, educational platforms, and communication devices.

Limitations

Static Alphabet: The model may misclassify similar gestures with overlapping visual features.

Dynamic Gestures: Performance decreases with fewer training classes or unbalanced data.

Generalization: May struggle under varying lighting conditions, complex backgrounds, or unseen gestures.