**Gesture Recognition**

**Neural\_Nets\_Project\_Starter\_Code (1)**

Batch Size= 32

Epoch = 50

**Model Architecture**

Model: "sequential\_2"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_6 (Conv3D) (None, 19, 100, 100, 32) 2624

activation\_6 (Activation) (None, 19, 100, 100, 32) 0

batch\_normalization\_6 (Batc (None, 19, 100, 100, 32) 128

hNormalization)

max\_pooling3d\_6 (MaxPooling (None, 9, 50, 50, 32) 0

3D)

conv3d\_7 (Conv3D) (None, 9, 50, 50, 64) 55360

activation\_7 (Activation) (None, 9, 50, 50, 64) 0

batch\_normalization\_7 (Batc (None, 9, 50, 50, 64) 256

hNormalization)

max\_pooling3d\_7 (MaxPooling (None, 4, 25, 25, 64) 0

3D)

conv3d\_8 (Conv3D) (None, 4, 25, 25, 128) 221312

activation\_8 (Activation) (None, 4, 25, 25, 128) 0

batch\_normalization\_8 (Batc (None, 4, 25, 25, 128) 512

hNormalization)

max\_pooling3d\_8 (MaxPooling (None, 2, 12, 12, 128) 0

3D)

flatten\_2 (Flatten) (None, 36864) 0

dense\_4 (Dense) (None, 512) 18874880

dropout\_2 (Dropout) (None, 512) 0

dense\_5 (Dense) (None, 5) 2565

=================================================================

Total params: 19,157,637

Trainable params: 19,157,189

Non-trainable params: 448

This is a model architecture for a 3D convolutional neural network (CNN) used for video classification. The model consists of 3 blocks of layers:

The first block starts with a 3D convolutional layer with 32 filters, each of size 3x3x3. This layer takes input of size 19 frames (i.e., time dimension), 100x100 pixels in height and width, and with 3 color channels (RGB). The layer is followed by an activation function (ReLU), a batch normalization layer, and a 3D max pooling layer with a pool size of 2x2x2, which reduces the spatial dimensions by a factor of 2.

The second block has a similar structure as the first block but with more filters. It starts with a 3D convolutional layer with 64 filters, followed by an activation function, a batch normalization layer, and a 3D max pooling layer with a pool size of 2x2x2.

The third block has the same structure as the previous two blocks but with even more filters. It starts with a 3D convolutional layer with 128 filters, followed by an activation function, a batch normalization layer, and a 3D max pooling layer with a pool size of 2x2x2.

After the convolutional layers, the model includes a flatten layer to convert the 3D feature maps into a 1D vector, followed by two dense layers, the first with 512 units and the second with 5 units (one for each output class). The dense layers are connected by a dropout layer with a rate of 0.5 to prevent overfitting.

The model has a total of 19,157,637 parameters, which include the weights and biases of the convolutional and dense layers. Of these, 19,157,189 are trainable parameters that will be adjusted during training, while the remaining 448 are non-trainable parameters that correspond to the moving statistics of the batch normalization layers.

**Neural\_Nets\_Project\_Starter\_Code (2)**

Batch Size= 32

Epoch = 50

**Model Architecture**

Model: "sequential\_19"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_56 (Conv3D) (None, 16, 100, 100, 16) 1312

max\_pooling3d\_44 (MaxPoolin (None, 8, 50, 50, 16) 0

g3D)

conv3d\_57 (Conv3D) (None, 8, 50, 50, 32) 13856

max\_pooling3d\_45 (MaxPoolin (None, 4, 25, 25, 32) 0

g3D)

conv3d\_58 (Conv3D) (None, 4, 25, 25, 64) 55360

max\_pooling3d\_46 (MaxPoolin (None, 2, 12, 12, 64) 0

g3D)

flatten\_12 (Flatten) (None, 18432) 0

dense\_26 (Dense) (None, 256) 4718848

dropout\_24 (Dropout) (None, 256) 0

dense\_27 (Dense) (None, 128) 32896

dropout\_25 (Dropout) (None, 128) 0

dense\_28 (Dense) (None, 5) 645

=================================================================

Total params: 4,822,917

Trainable params: 4,822,917

Non-trainable params: 0

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

None

The model consists of three convolutional layers followed by max pooling layers. Then, the output is flattened and passed through two fully connected layers with dropout regularization. Finally, the output layer uses softmax activation to classify inputs into one of the five categories. The model is compiled using stochastic gradient descent optimizer, categorical cross-entropy loss, and categorical accuracy metric.

**Neural\_Nets\_Project\_Starter\_Code (3)**

Batch Size= 50

Epoch = 50

**Model Architecture**

Model: "sequential\_3"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_8 (Conv3D) (None, 15, 100, 100, 32) 2624

batch\_normalization (BatchN (None, 15, 100, 100, 32) 128

ormalization)

activation (Activation) (None, 15, 100, 100, 32) 0

max\_pooling3d\_8 (MaxPooling (None, 7, 50, 50, 32) 0

3D)

conv3d\_9 (Conv3D) (None, 7, 50, 50, 64) 55360

batch\_normalization\_1 (Batc (None, 7, 50, 50, 64) 256

hNormalization)

activation\_1 (Activation) (None, 7, 50, 50, 64) 0

max\_pooling3d\_9 (MaxPooling (None, 3, 25, 25, 64) 0

3D)

conv3d\_10 (Conv3D) (None, 3, 25, 25, 128) 221312

batch\_normalization\_2 (Batc (None, 3, 25, 25, 128) 512

hNormalization)

activation\_2 (Activation) (None, 3, 25, 25, 128) 0

max\_pooling3d\_10 (MaxPoolin (None, 1, 12, 12, 128) 0

g3D)

flatten\_2 (Flatten) (None, 18432) 0

dense\_6 (Dense) (None, 64) 1179712

dropout\_4 (Dropout) (None, 64) 0

dense\_7 (Dense) (None, 5) 325

=================================================================

Total params: 1,460,229

Trainable params: 1,459,781

Non-trainable params: 448

This is a 3D convolutional neural network (CNN) designed to process 3D inputs with a shape of (15, 100, 100, 3). The network consists of 3 sets of convolutional layers with increasing number of filters (32, 64, and 128) each followed by batch normalization, ReLU activation function and max pooling layer. The output is flattened and fed into a fully connected layer with 64 neurons and a dropout rate of 0.5 to prevent overfitting. Finally, the output layer uses a softmax activation function with 5 units to predict the class probabilities. The model is trained using stochastic gradient descent (SGD) optimizer and the categorical cross-entropy loss function and the categorical accuracy metric are used to evaluate the performance of the model.

**Neural\_Nets\_Project\_Starter\_Code (4)**

Batch Size= 32

Epoch = 50

**Model Architecture**

Model: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d (Conv3D) (None, 30, 100, 100, 16) 1312

batch\_normalization (BatchN (None, 30, 100, 100, 16) 64

ormalization)

max\_pooling3d (MaxPooling3D (None, 15, 50, 50, 16) 0

)

conv3d\_1 (Conv3D) (None, 15, 50, 50, 32) 13856

batch\_normalization\_1 (Batc (None, 15, 50, 50, 32) 128

hNormalization)

max\_pooling3d\_1 (MaxPooling (None, 7, 25, 25, 32) 0

3D)

conv3d\_2 (Conv3D) (None, 7, 25, 25, 64) 55360

batch\_normalization\_2 (Batc (None, 7, 25, 25, 64) 256

hNormalization)

max\_pooling3d\_2 (MaxPooling (None, 3, 12, 12, 64) 0

3D)

flatten (Flatten) (None, 27648) 0

dense (Dense) (None, 256) 7078144

dense\_1 (Dense) (None, 128) 32896

dense\_2 (Dense) (None, 5) 645

=================================================================

Total params: 7,182,661

Trainable params: 7,182,437

Non-trainable params: 224

The model consists of three convolutional layers, followed by max pooling layers, batch normalization and ReLU activation. The flattened output is fed into two fully connected layers with ReLU activation and one output layer with softmax activation. The model is compiled with stochastic gradient descent optimizer, categorical cross-entropy loss and categorical accuracy metrics.

**Neural\_Nets\_Project\_Starter\_Code (5)**

Batch Size= 50

Epoch = 50

**Model Architecture**

Model: "sequential\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_3 (Conv3D) (None, 19, 100, 100, 32) 2624

activation\_3 (Activation) (None, 19, 100, 100, 32) 0

batch\_normalization\_3 (Batc (None, 19, 100, 100, 32) 128

hNormalization)

max\_pooling3d\_3 (MaxPooling (None, 9, 50, 50, 32) 0

3D)

conv3d\_4 (Conv3D) (None, 9, 50, 50, 64) 55360

activation\_4 (Activation) (None, 9, 50, 50, 64) 0

batch\_normalization\_4 (Batc (None, 9, 50, 50, 64) 256

hNormalization)

max\_pooling3d\_4 (MaxPooling (None, 4, 25, 25, 64) 0

3D)

conv3d\_5 (Conv3D) (None, 4, 25, 25, 128) 221312

activation\_5 (Activation) (None, 4, 25, 25, 128) 0

batch\_normalization\_5 (Batc (None, 4, 25, 25, 128) 512

hNormalization)

max\_pooling3d\_5 (MaxPooling (None, 2, 12, 12, 128) 0

3D)

flatten\_1 (Flatten) (None, 36864) 0

dense\_2 (Dense) (None, 512) 18874880

dropout\_1 (Dropout) (None, 512) 0

dense\_3 (Dense) (None, 5) 2565

=================================================================

Total params: 19,157,637

Trainable params: 19,157,189

Non-trainable params: 448

This code creates a 3D convolutional neural network for video classification. The model has three convolutional layers with increasing number of filters, each followed by ReLU activation, batch normalization, and max pooling. The output is flattened and fed into a fully connected layer with 512 units and a dropout of 0.5, and then into the output layer with a softmax activation. The model is trained using stochastic gradient descent (SGD) with categorical cross-entropy loss and categorical accuracy as the evaluation metric.

**Neural\_Nets\_Project\_Starter\_Code (6)**

Batch Size= 50

Epoch = 50

**Model Architecture**

Model: "sequential\_3"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param #

=================================================================

conv3d\_9 (Conv3D) (None, 30, 100, 100, 8) 656

batch\_normalization\_9 (Batc (None, 30, 100, 100, 8) 32

hNormalization)

max\_pooling3d\_9 (MaxPooling (None, 15, 50, 50, 8) 0

3D)

conv3d\_10 (Conv3D) (None, 15, 50, 50, 16) 3472

batch\_normalization\_10 (Bat (None, 15, 50, 50, 16) 64

chNormalization)

max\_pooling3d\_10 (MaxPoolin (None, 7, 25, 25, 16) 0

g3D)

conv3d\_11 (Conv3D) (None, 7, 25, 25, 32) 4640

batch\_normalization\_11 (Bat (None, 7, 25, 25, 32) 128

chNormalization)

max\_pooling3d\_11 (MaxPoolin (None, 3, 12, 12, 32) 0

g3D)

conv3d\_12 (Conv3D) (None, 3, 12, 12, 64) 18496

batch\_normalization\_12 (Bat (None, 3, 12, 12, 64) 256

chNormalization)

max\_pooling3d\_12 (MaxPoolin (None, 1, 6, 6, 64) 0

g3D)

flatten\_3 (Flatten) (None, 2304) 0

dense\_4 (Dense) (None, 128) 295040

batch\_normalization\_13 (Bat (None, 128) 512

chNormalization)

dropout\_2 (Dropout) (None, 128) 0

dense\_5 (Dense) (None, 64) 8256

batch\_normalization\_14 (Bat (None, 64) 256

chNormalization)

dropout\_3 (Dropout) (None, 64) 0

dense\_6 (Dense) (None, 5) 325

=================================================================

Total params: 332,133

Trainable params: 331,509

Non-trainable params: 624

The input shape of the video data is defined as 30 frames with a size of 100x100 pixels and 3 channels. The model consists of 4 3D convolutional layers with batch normalization, max pooling and ReLU activation, followed by a flatten layer. Two fully connected layers with batch normalization and dropout are then added, before the final output layer with softmax activation. The model uses the Adam optimizer and categorical cross-entropy loss function for training, and the categorical accuracy metric is used to evaluate the model's performance.

**Model Performance**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Batch Size** | **Epochs** | **Training Accuracy** | **Validation Accuracy** | **Input Size** |
| **Neural\_Nets\_Project\_Starter\_Code (1)** | **32** | **50** | **0.992** | **0.72** | **(19,100,100,3)** |
| **Neural\_Nets\_Project\_Starter\_Code (2)** | **50** | **50** | **0.8145** | **0.8100** | **(16,100,100,3)** |
| **Neural\_Nets\_Project\_Starter\_Code (3)** | **50** | **50** | **0.2066** | **0.2700** | **(15, 100, 100, 3)** |
| **Neural\_Nets\_Project\_Starter\_Code (4)** | **32** | **50** | **0.9940** | **0.7900** | **(30, 100, 100, 3)** |
| **Neural\_Nets\_Project\_Starter\_Code (5)** | **50** | **50** | **0.9578** | **0.6100** | **(19,100,100,3)** |
| **Neural\_Nets\_Project\_Starter\_Code (6)** | **50** | **50** | **0.9683** | **0.6700** | **(30, 100, 100, 3)** |

**Conclusion**

Based on the information provided, Model 1 with a batch size of 32, 50 epochs, training accuracy of 0.992 and validation accuracy of 0.72 appears to be the best model. This is because it has the highest validation accuracy among all the models.

Model 2 has a higher validation accuracy than Model 1 but it has a lower training accuracy which suggests that it may be overfitting the data.

Model 3 has a very low accuracy on both training and validation data and therefore, it is not a good model for this dataset.

Model 4 has a high training accuracy but a lower validation accuracy compared to Model 1 which suggests that it may be overfitting the data.

Model 5 has a lower validation accuracy and training accuracy compared to Model 1.

Model 6 has a higher validation accuracy compared to Model 5 but lower training accuracy which again suggests overfitting.

Therefore, based on the information provided, Model 1 is the best model with a batch size of 32, 50 epochs, a training accuracy of 0.992 and a validation accuracy of 0.72.