

## Gesture Recognition Case Study- Write Up

We have videos of 30 frames each to train the model. There are 663 videos in the train set and 100 videos in the validation set.

### Preprocessing steps and hyper parameters

Input image dimension: 100\*100

We have 2 different resolutions for the images. One is 360\*360, other is 120\*160. For the model, all images have to be of the same dimension. For that, I have cropped images where necessary and resize all images to 100\*100.

Batch Size: 30 I have chosen the batch size as 30 for a start. So, the number of batches will be 22 for the train set. I will be experimenting with this as we proceed.

Image Indexes taken: 9 out of 30 As a starting model, I am only adding 9 images out of the 30. I have selected 9 images from in between the indexes 0 to 29.

[0,3,7,11,15,19,23,27,29]

Number of epochs: 2 To start with I have chosen the number of epochs as 2. This will be increased later.

### Architecture

I have chosen Conv3D architecture as the model for this problem statement since the architecture is simpler and we have fewer parameters to train.

As a base model I am adding the following layers to the model.

Conv3D layer to learn the features: 16 filters, kernel size- (3,3), activation='relu'

Maxpooling3D layer to reduce the dimension and get summarized output.

Flatten layer to convert pooled features to single linear vector.

Dense layer to make classification.

Softmax activation layer for multi label classification.

All the below experiments are on top of this Conv3D architecture.

No	Important model parameters	Result + Inference
	<b>Base model</b> Batch_size- 30 img_idx = [0,3,7,11,15,19,23,27,29] Images per sequence = 9 Input image size = 100*100	

1	# of epoch- 2 batch_size- 30 Num of batches = 22 Trainable params: 34,422,821	Train Accuracy- 58.6% Validation Accuracy- 31% Model is working. But accuracy is low. May be the model needs more epochs to try <b>Next:</b> Lets train the model for 30 epochs
2	<b># of epoch- 30</b> batch_size- 30 Num of batches = 22 Trainable params: 34,422,821	Epoch 27 gave below best result Train Accuracy- 100% Validation Accuracy- 52% Train accuracy is maximum. Model has learned the train data completely. But validation accuracy is very low. The model has overfitted. Also, the model is taking a lot of time to learn.
Going to try different batch sizes and see if the system is able to handle it. Also, running for 5 epochs		
3	<b># of epochs- 5</b> <b>batch_size = 50</b> <b>Num of batches = 13</b> Trainable params: 34,422,821	Epoch 5 gave highest accuracy Train Accuracy- 89.9% Validation Accuracy- 53% The model is still overfitting. <b>Next:</b> Increasing batch size to 70
4	# of epochs- 5 <b>batch_size = 70</b> <b>Num of batches = 9</b> Trainable params: 34,422,821	Epoch 5 gave highest accuracy Train Accuracy- 73.1% Validation Accuracy- 49% The accuracy has reduced. The training is now slow due to an increase in batch_size. <b>Next:</b> Let's go back to batch_size 50.
Fixing batch size at 50. Let's try changing image indexes from the sequence to use in model. Using all 30 images as input to model.		
5	# of epochs- 5 batch_size = 50 Num of batches = 13 <b># of images per seq = 30</b> Trainable params: 137,685,029	ValueError: Could not find a format to read the specified file in single-image mode One of the images in the data might be corrupt or not opening. <b>Next:</b> Have to find the erroneous image

After trying several times, I found that Image index 4 is having error. So, we will not use that in further models. Now lets try all images except 4.

Reducing batch size to 30 since image per sequence is more.

Fixing batch size at 30.

6	# of epochs- 5 <b># of images per sequence = 29</b> <b>Trainable params:132,767,781</b>	Epoch 5 gave highest accuracy Train Accuracy- 98.9% Validation Accuracy- 43%  The model is overfitting and takes a lot of time to train 5 epochs. This is because we have too many parameters.  <b>Next:</b> Let's reduce the images per sequence from 29 to 15. Will remove few of the images
7	# of epochs- 5 <b>Number of images used per sequence = 15</b> <b>Trainable params:63,926,309</b>	Epoch 5 gave highest accuracy Train Accuracy- 99.55% Validation Accuracy- 42%  The model too is overfitting, but has almost same accuracy as using 29 images in a sequence.  <b>Next:</b> Lets reduce the number of images per sequence further to 9
8	# of epochs- 5 <b># of images per seq = 9</b> <b>Trainable params: 34,422,821</b>	Epoch 5 gave highest accuracy Train Accuracy- 98.64% Validation Accuracy- 45%  With just 9 images in sequence we are getting more validation accuracy than last time.  <b>Next:</b> Let's experiment with input image size.
Changing input image size from 100*100 to 50*50. This will reduce the model parameters and make training faster. batch_size = 30 Num of batches = 22 Number of images used per sequence = 9		

9	# of epochs- 5 Input image size = 50*50 Trainable params: 8,259,621	Epoch 5 gave highest accuracy Train Accuracy- 86.23% Validation Accuracy- 51%  With only 8M trainable params, the training is much faster and the result is better. But the model is still overfitting. <b>Next:</b> Increasing batch size to 70, since model is training faster now.
10	# of epochs- 5 <b>Batch size = 70</b> <b># of batches = 9</b> Input image size = 50*50 Trainable params: 8,259,621	Epoch 5 gave highest accuracy Train Accuracy- 65.76% Validation Accuracy- 52%  Best validation accuracy so far. <b>Next:</b> Lets increase batch size to 100 and run for 20 epochs
11	<b># of epochs- 20</b> <b>Batch size = 100</b> Num batches = 7 Trainable params: 8,259,621	Epoch 10 gave best result Train accuracy: 96.83% Validation Accuracy: 64%  Best accuracy so far. But the model is not training further after a few epochs <b>Next:</b> Adding a dropout layer to reduce overfitting.
12	# of epochs- 20 <b>Dropout(0.5) added</b> Trainable params: 8,259,621	Train accuracy reached max Train accuracy: 100% Validation Accuracy: 62%  <b>Next:</b> Removing dropout layer as it did not help much. Let's try changing optimizer from 'adam' to 'sdg' to see if that helps the learning rate.
13	# of epochs- 20 <b>Optimiser- sdg</b> Trainable params: 8,259,621	Train accuracy: 62.75% Validation Accuracy: 54%  Accuracy has not increased <b>Next:</b> Increasing number of filters from 16 to 32. Changing optimizer back to 'adam'

14	# of epochs- 20 <b># of filters- 32</b> <b>Optimiser = 'adam'</b> <b>Trainable params: 16,518,469</b>	Train accuracy maxed out Train accuracy: 100% Validation Accuracy: 60% Accuracy has not improved much. <b>Next:</b> Changing kernel size from (3,3,3,) to (2,2,2) and going back to 16 filters
15	# of epochs- 20 <b>Kernel size- (2,2,2)</b> <b>Number of filters= 16</b> <b>Trainable params: 9,438,357</b>	Best result so far in epoch 10 Train accuracy: 91.5% Validation Accuracy: 70% Validation accuracy is good. We will try to make it better. <b>Next:</b> The model may not be able to learn more. Adding another conv3D. This might help in learning more combination of features which is outputted from previous layer. It will also help in reducing the number of parameters.
16	2 filters added- 16*(3,3,3) <b>32*(3,3,3)</b> # of epochs- 20 <b>Trainable params: 2,494,021</b>	Best result so far in epoch 19 Train accuracy: 99.1% Validation Accuracy: 71% Since the parameters are reduced, the model is training much faster now. Also accuracy has improved by around 10%. <b>Next:</b> Changing first conv3D layer filter to (2,2,2) since that gave better results earlier.
17	Filters- 16*(2,2,2) 32*(3,3,3) # of epochs- 20 <b>Trainable params: 2,988,725</b>	Best result so far in epoch 13 Train accuracy: 95.48% Validation Accuracy: 73% <b>Next:</b> Adding dropout layer to handle overfitting
18	<b>dropout(0.5) layer added.</b> Kernel size- 16*(2,2,2) 32*(3,3,3) batch_size = 100 Trainable params: 2,988,725	Best result so far in epoch 13 Train accuracy: 95.32% Validation Accuracy: 77% Model-00013-0.18151-0.95324-0.96800-0.77000.h5 This will be our final model.

## **FINAL MODEL**

## **Generator Settings**

Input image size: 50\*50

Batch Size: 100

Number of batches: 7

Image indexes input: [3,7,9,11,15,19,23,27,29]

number of images per sequence: 9

Number of epochs: 20

## **Model Layers**

Conv3D layer: 16 filters, kernel size (2,2,2), activation = 'relu'

MaxPooling3D layer: pool\_size=(1, 2, 2), strides=(1, 2, 2)

Conv3D layer: 32 filters, kernel size (3,3,3), activation = 'relu'

MaxPooling3D layer: pool\_size=(1, 2, 2), strides=(1, 2, 2)

Flatten layer

Dense layer: 128 neurons

Dropout layer: Rate- 0.5

Dense layer: 5 neurons, activation = 'softmax'

## **Accuracy**

Train accuracy: 95.32%

Validation Accuracy: 77%

Model: model-00013-0.18151-0.95324-0.96800-0.77000.h5