

Gesture Recognition

Problem Statement

Recognize five different gestures performed by the user which will help users control the TV without using a remote. The gestures are continuously monitored by the webcam mounted on the TV. Each gesture corresponds to a specific command:

- Thumbs up: Increase the volume
- Thumbs down: Decrease the volume
- Left swipe: 'Jump' backwards 10 seconds
- Right swipe: 'Jump' forward 10 seconds
- Stop: Pause the movie

Each video is a sequence of 30 frames (or images), provided for training and validation. These videos have been recorded by various people performing one of the five gestures in front of a webcam - similar to what the smart TV will use.

Different experiments have been performed to determine the better working model for gesture recognition.

The below parameter will be used throughout the document.

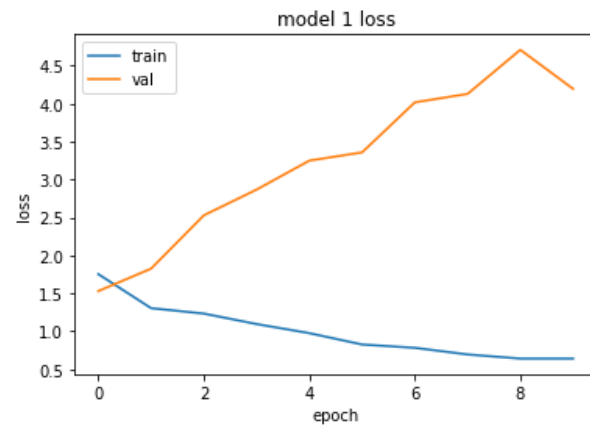
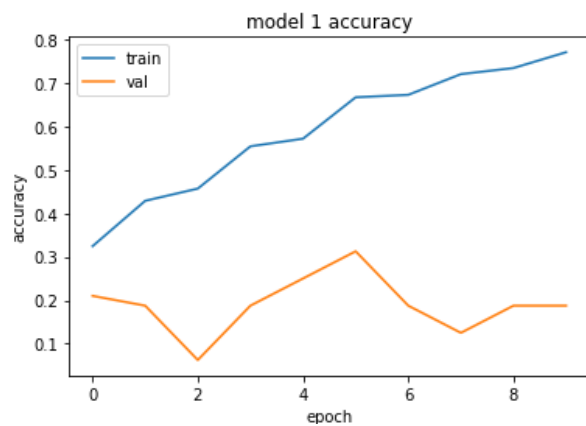
x # number of frames

y # image width

z # image height

Experiment # 1

Exp #	Model	Parameters	Result	
1	Conv3D Activation = relu Optimizer = Adam	batch_size = 32 x = 16, y = 100, z = 100 dropout = 0.25 epochs = 10		
			loss	0.6413
			categorical_accuracy	0.7703
			val_loss	4.1957
			val_categorical_accuracy	0.1875
			Learning rate	6.2500e-05



Conclusion

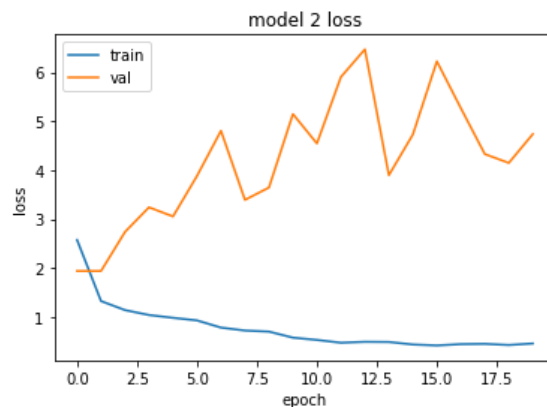
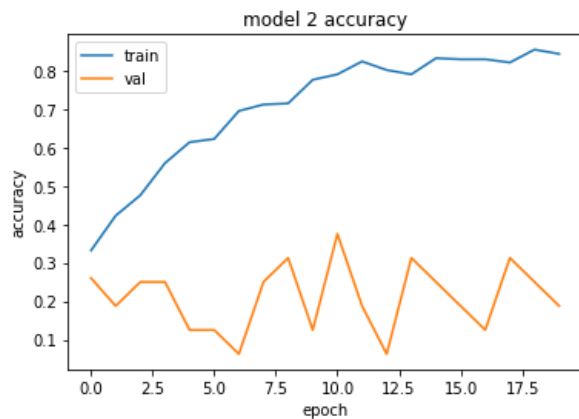
The model is performing well on training data. The accuracy increases as the number of epochs is increasing but performs poorly on validation data. The model is clearly overfitting and needs to be fixed.

Next Step

Increase the number of epochs for the model to capture the data trends better

Experiment # 2

Exp #	Model	Parameters	Result	
2	Conv3D Activation = relu Optimizer = Adam	batch_size = 32 x = 16, y = 100, z = 100 dropout = 0.10 epochs = 20		
			loss	0.4596
			categorical_accuracy	0.8431
			val_loss	4.7383
			val_categorical_accuracy	0.1875
			Learning rate	3.9063e-06



Conclusion

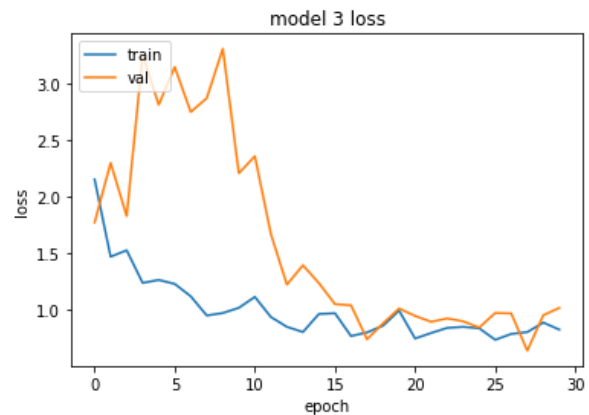
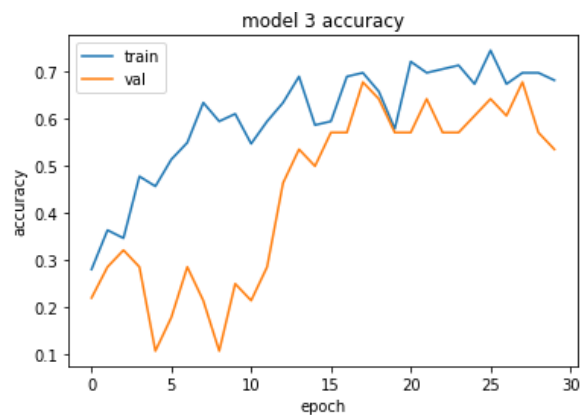
The model is performing well on training data. The accuracy increased over previous experiment as the number of epochs is increasing but performs poorly on validation data. The model is still not performing well and is overfitting.

Next Step

Reduce the batch size, increase the number of frames (number of images used for each video) and Increase the number of epochs for the model to capture the data trends better

Experiment # 3

Exp #	Model	Parameters	Result	
3	Conv3D Activation = relu Optimizer = Adam	batch_size = 16 x = 20, y = 100, z = 100 dropout = 0.10 epochs = 30	loss	0.8245
			categorical_accuracy	0.6825
			val_loss	1.0161
			val_categorical_accuracy	0.5357
			Learning rate	9.7656e-07



Conclusion

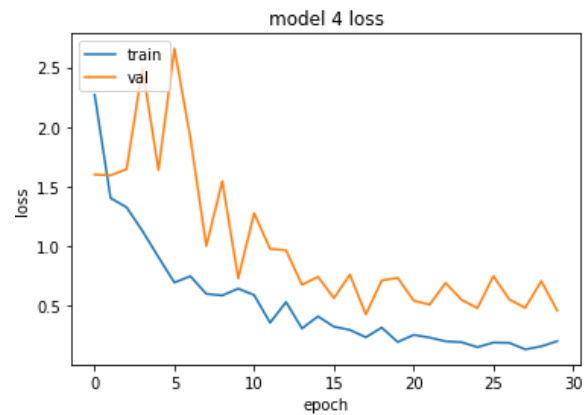
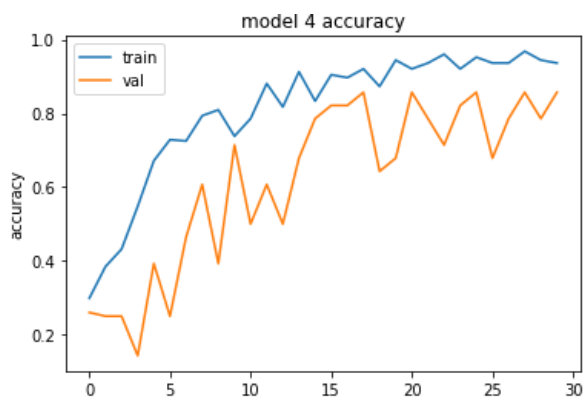
The model is performing well on both training and validation data. The accuracy of the model on validation data increased significantly over previous experiments as the number of epochs is increasing but overall accuracy is not at desired level. However, the loss is significantly higher for both training and validation data and the model is clearly underfitting.

Next Step

Increase the number of images per batch and increase the final size of input images. Reduce the number of internal layers for the model and optimize using SGD to cap the learning rate.

Experiment # 4

Exp #	Model	Parameters	Result	
4	Conv3D Activation = relu Optimizer = SGD	batch_size = 16 x = 20, y = 120, z = 120 dropout = 0.10 learning_rate=0.01 epochs = 30	loss	0.2050
			categorical_accuracy	0.9365
			val_loss	0.4621
			val_categorical_accuracy	0.8571
			Learning rate	3.9062e-05



Conclusion

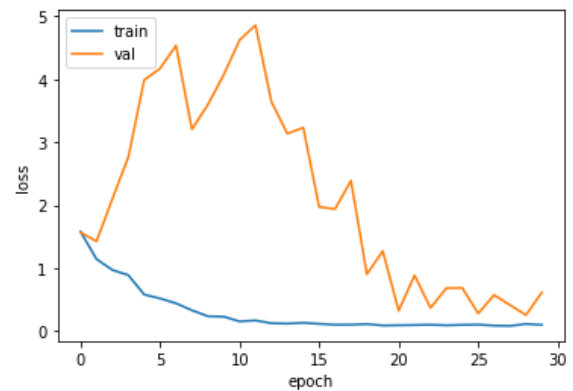
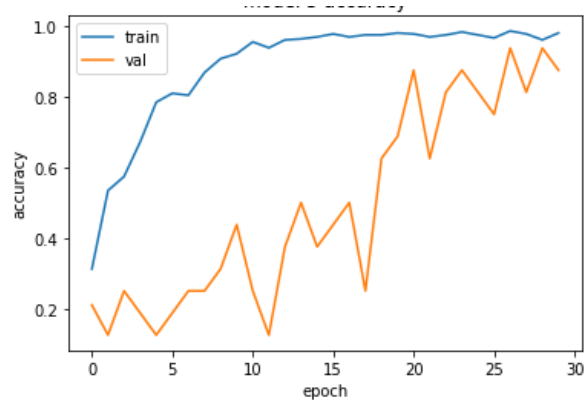
The model is performing well on both training and validation data. The accuracy of the model on validation and training data increased significantly over previous experiment as the number of epochs is increasing. The loss has significantly reduced for both training and validation data and the model is performing well.

Next Step

Increase the number of internal layers for the model and batch size to check if the accuracy increases.

Experiment # 5

Exp #	Model	Parameters	Result	
5	Conv3D Activation = relu Optimizer = SGD	batch_size = 16 x = 20, y = 120, z = 120 dropout = 0.10 learning_rate=0.01 momentum=0.9 nesterov=True epochs = 30		
			loss	0.0986
			categorical_accuracy	0.9804
			val_loss	0.6131
			val_categorical_accuracy	0.8750
			Learning rate	4.8828e-06



Conclusion

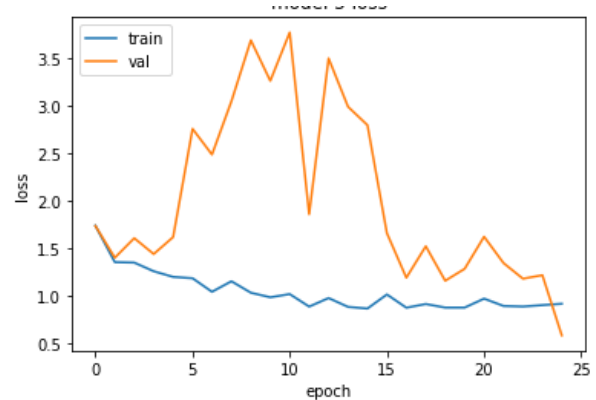
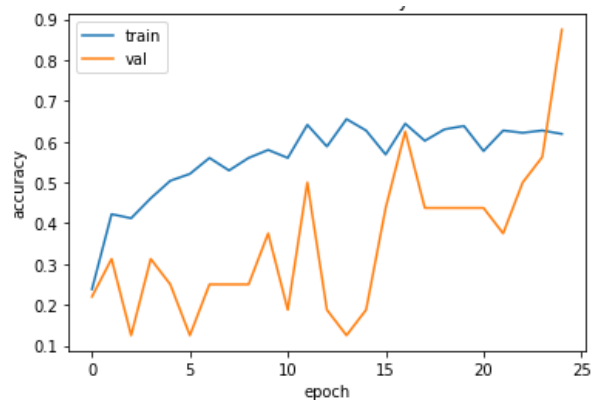
The model is performing well training data and accuracy of the validation data is good, but the validation loss is significant. Though it reduces as the number of epochs increase there is a clear nonlinearity. The model is slightly underfitting.

Next Step

Decrease the number of epochs for the model to check if the accuracy increases.

Experiment # 6

Exp #	Model	Parameters	Result	
6	Conv3D Activation = relu Optimizer = SGD	batch_size = 16 x = 20, y = 120, z = 120 dropout = 0.10 learning_rate=0.01 momentum=0.9 nesterov=True epochs = 25		
			loss	0.9194
			categorical_accuracy	0.6190
			val_loss	0.5864
			val_categorical_accuracy	0.8750
			Learning rate	1.9531e-05



Conclusion

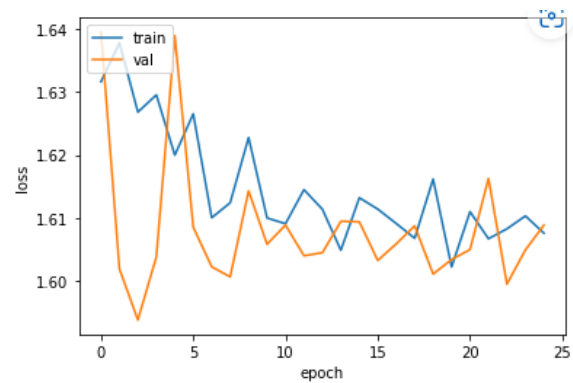
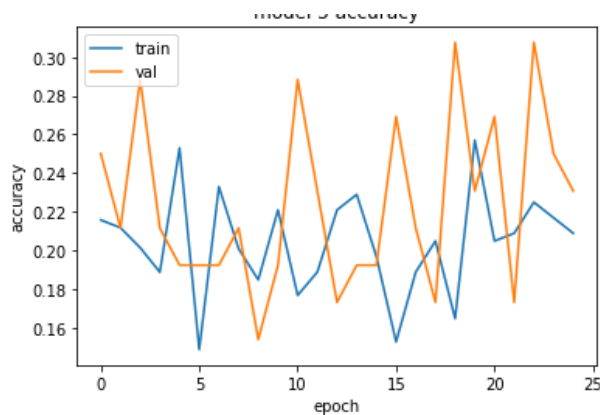
The model's performance has degraded on training data. Both accuracy and loss of the training data have degraded, but the validation accuracy is significant. The reduction in the number of epochs has affected the model accuracy.

Next Step

Build the model using Conv2D+RNN, by reducing the batch size

Experiment # 7

Exp #	Model	Parameters	Result										
7	Conv2D + GRU Activation = relu, softmax Optimizer = Adam	batch_size = 8 x = 20, y = 120, z = 120 dropout = 0.25 epochs = 25	<table><tr><td>loss</td><td>1.6076</td></tr><tr><td>categorical_accuracy</td><td>0.2088</td></tr><tr><td>val_loss</td><td>1.6089</td></tr><tr><td>val_categorical_accuracy</td><td>0.2308</td></tr><tr><td>Learning rate</td><td>9.7656e-07</td></tr></table>	loss	1.6076	categorical_accuracy	0.2088	val_loss	1.6089	val_categorical_accuracy	0.2308	Learning rate	9.7656e-07
loss	1.6076												
categorical_accuracy	0.2088												
val_loss	1.6089												
val_categorical_accuracy	0.2308												
Learning rate	9.7656e-07												



Conclusion

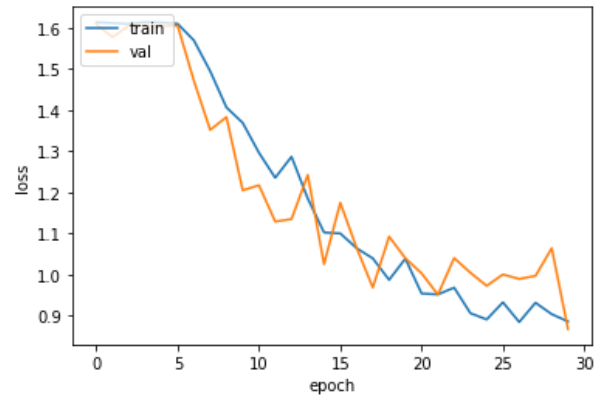
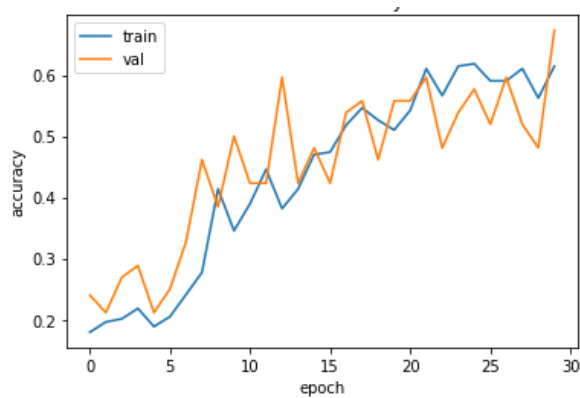
The model is performance is very bad for both training and validation data. The model needs to perform better as accuracy stands at 20% and the loss is very significant. Batch size reduction and number of epochs have significant affect on the accuracy and loss metrics of the model.

Next Step

Build the model using Conv2D+RNN, by increasing the number of epochs, reducing the number of images four use for each video and increase the number of flatten layers.

Experiment # 8

Exp #	Model	Parameters	Result	
8	Conv2D + GRU Activation = relu, softmax Optimizer = Adam	batch_size = 8 x = 15, y = 120, z = 120 dropout = 0.25 epochs = 30		
			loss	0.8861
			categorical_accuracy	0.6145
			val_loss	0.8678
			val_categorical_accuracy	0.6731
			Learning rate	3.9063e-06



Conclusion

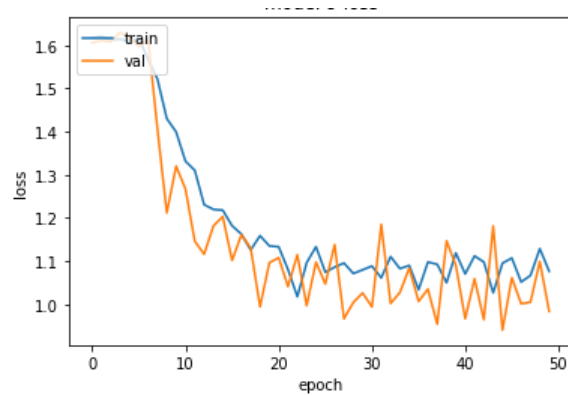
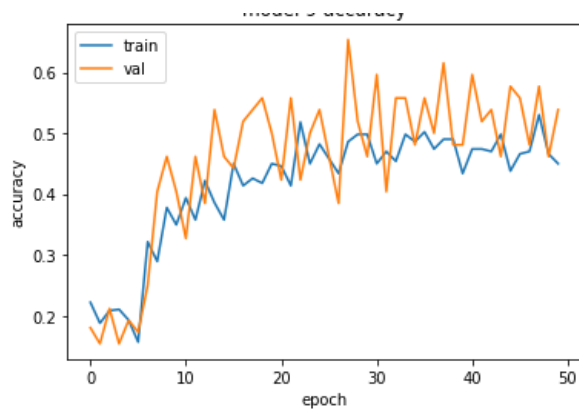
The model accuracy has increased for both training and validation data. But the loss is significant for both validation and training datasets. The model errors are significant and would need to be tuned to stop this.

Next Step

Increase the number of epochs and try to use LSTM to check the model performance for any improvement. Increase the number of flatten layers.

Experiment # 9

Exp #	Model	Parameters	Result	
9	Conv2D + LSTM Activation = relu, softmax Optimizer = Adam	batch_size = 8 x = 15, y = 120, z = 120 dropout = 0.25 epochs = 50	loss	1.0766
			categorical_accuracy	0.4498
			val_loss	0.9837
			val_categorical_accuracy	0.5385
			Learning rate	3.8147e-09



Conclusion

The model accuracy and loss have degraded for both training and validation data. The loss is significant for both validation and training datasets. The model errors are significant and would need to be tuned to stop this.

Model Selection

After performing experiments with Conv3D and Conv2D+RNN, experiment # 4 has clearly outperformed other models in terms of accuracy, loss, learning rate and performance.

Exp #	Model	Parameters	Result	
4	Conv3D Activation = relu Optimizer = SGD	batch_size = 16 x = 20, y = 120, z = 120 dropout = 0.10 learning_rate=0.01 epochs = 30	loss	0.2050
			categorical_accuracy	0.9365
			val_loss	0.4621
			val_categorical_accuracy	0.8571
			Learning rate	3.9062e-05