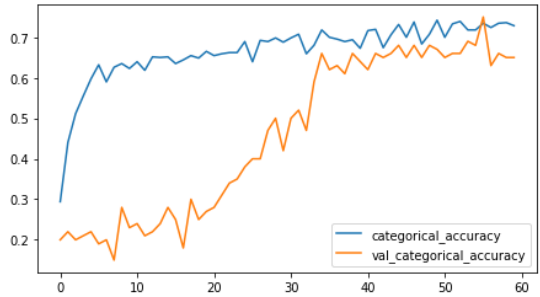
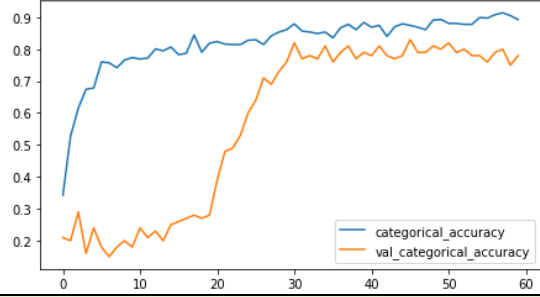
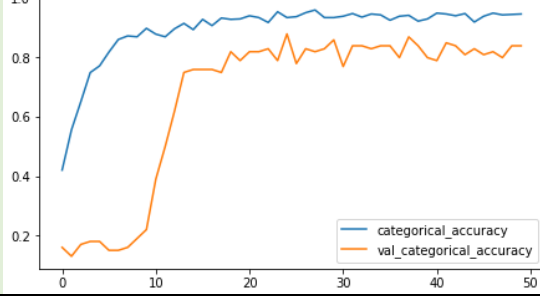
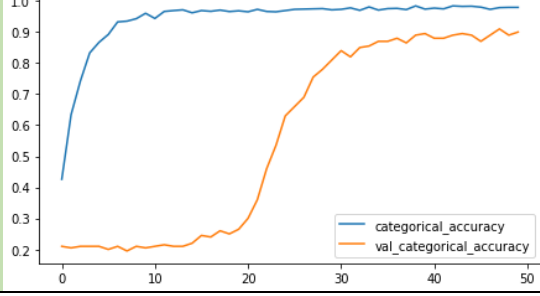
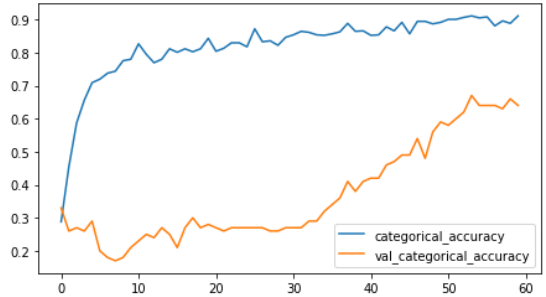
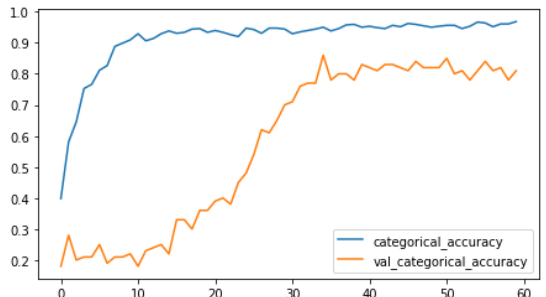
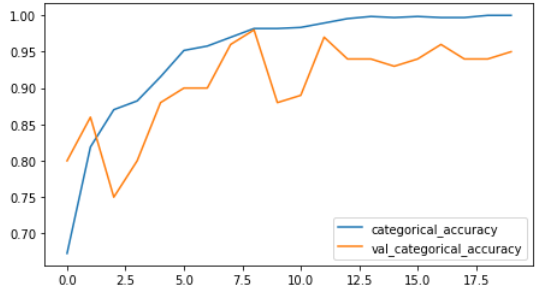
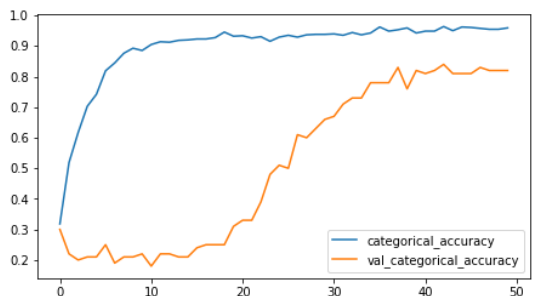


Gesture Recognition: Case Study

<https://github.com/hypercalm>

| No. | Model | Architecture | Results |
|------------------|--|---|---|
| 1 | Conv3D Shape = (16, 120, 120, 3) Batch size = 40 Epochs = 60 Optim = Adam (LR = 0.0002) | 5 Conv layers: 8 to 128 1 Dense layer Dropouts: 0.25 in final conv and dense layer Total Parameters: 222,109 | Best: 74% T, 75% V (Epoch 56) Last: 73% T, 65% V  |
| 2 | Conv3D Shape = (16, 120, 120, 3) Batch size = 32 Epochs = 60 Optim = Adam (LR = 0.0002) | 4 Conv layers: 16 to 128 2 Dense layers Dropouts: 0.5 in dense layer Total Parameters: 1,762,613 | Best: 88% T, 83% V (Epoch 46) Last: 89% T, 78% V  |
| 3 Second best | Conv3D Shape = (12, 120, 120, 3) Batch size = 16 Epochs = 50 Optim = Adam (LR = default) | 4 Conv layers: 16 to 128 2 Dense layers Dropouts: 0.5 in dense layers Total Parameters: 976,181 | Best: 94% T, 88% V (Epoch 25) Last: 95% T, 84% V  |
| 4 Final Model | Conv3D Shape = (16, 120, 120, 3) Batch size = 32 Epochs = 50 Optim = Adam (LR = default) | 5 Conv layers: 16 to 256 2 Dense layers Dropouts: 0.5 in dense layers Total Parameters: 682,549 | Best: 98% T, 91% V (Epoch 48) Last: 98% T, 90% V  |

| | | | |
|---|--|---|---|
| 5 | <p>Conv2D + RNN (GRU) Shape = (16, 100, 100, 3) Batch size = 64 Epochs = 60 Optim = Adam (LR = 0.0002)</p> | <p>4 TD Conv layers: 16 to 128 128 GRU units 1 Dense layer Dropouts: 0.4 in GRU and dense layer Total Parameters: 1,934,949</p> | <p>Best: 91% T, 67% V (Epoch 54) Last: 91% T, 64% V</p>  |
| 6 | <p>Conv2D + RNN (GRU) Shape = (16, 120, 120, 3) Batch size = 32 Epochs = 60 Optim = Adam (LR = default)</p> | <p>4 TD Conv layers: 16 to 128 128 GRU units 1 Dense layer Dropouts: 0.5 in TD flatten and dense layer Total Parameters: 1,346,405</p> | <p>Best: 95% T, 86% V (Epoch 35) Last: 97% T, 81% V</p>  |
| 7 | <p>Conv2D (Mobilenet with imagenet weights) + RNN (GRU) Shape = (16, 128, 128, 3) Batch size = 8 Epochs = 20 Optim = Adam (LR = default)</p> | <p>TD with Mobilenet with trainable weights 128 GRU units 1 Dense layer Dropouts: 0.5 in GRU and dense layer</p> | <p>Best: 98% T, 98% V (Epoch 9) Last: 100% T, 95% V</p>  |
| 8 | <p>Conv2D + RNN (LSTM) Shape = (16, 120, 120, 3) Batch size = 32 Epochs = 50 Optim = Adam (LR = default)</p> | <p>5 TD Conv layers: 16 to 256 128 LSTM units 1 Dense layer Dropouts: 0.5 in TD flatten and dense layer Total Parameters: 1,657,445</p> | <p>Best: 96% T, 84% V (Epoch 43) Last: 96% T, 82% V</p>  |

Model outline and justifications

In the present study, two types of models have been developed:

- (a) The CNN models based on 3D convolution layers, and
- (b) The CNN + RNN models based on 2D time distributed convolution layers, LSTM and GRU units.

Considerations for model design

1. Batch size: Though it is advisable to use the maximum possible batch size that can be handled by the GPU resources available, I decided to go with 32 to 40 batch size for CNN 3D convolution models due to GPU constraints. Batch sizes up to 64 are used in some CNN + RNN models.
2. Model complexity: As the data available for training is not too large, we needed to keep model complexity under check. As a complex model would need more data to be fit without suffering from high variance (overfitting).

Comments on individual models

- **Model 1 Conv3D:** This model was chosen as an arbitrary model after some earlier experimentation regarding memory constraints. We thought sampling roughly half the images in the video should be sufficient, and so we took 16. The model has low parameter counts due to it having 5 convolution layers and only one dense layer. We find that the model underfits the data. We can say it has high bias and medium variance.
- **Model 2 Conv3D:** This model tries to overcome the limitations of model 1 in terms of increasing the model parameters and adding increasing the dropout rate to ensure enough room for avoiding high, while keeping variance under control. We see that it performs much better, but there is still plenty of room for improving accuracy.
- **Model 3 Conv3D:** We lower the number of images sampled from the video from 16 to 12 in this model. This helps us reduce the parameter count drastically. Furthermore, we decided to go with the default Adam learning rate which is 0.001 instead of our specified value earlier. As we noticed we needed too many epochs to get to the desired levels in earlier models. This becomes immediately apparent in the graphs where we see the validation accuracy climb up much faster. With less parameters, the model bias was further reduced, and we got the second-best validation accuracy of 88 here.
- **Model 4 Conv3D:** In this model, we tried looking at improving the architecture further by adding a convolution layer. So, we have 5 layers here and 2 dense layers. While this is similar to the first model in terms of convolution layers, the filter sizes used here allow us to keep a low parameter count despite retaining 2 dense layers. This gave us the best results of all models with validation accuracy reaching 91%.

- **Conv2D + RNN (GRU):** A similar attempt was made to start with a basic model and improve gradually. It has high bias and high variance.
- **Conv2D + RNN (GRU):** We increased image size and reduced parameters. We also lowered the batch size here. The performance has increased significantly. Both bias and variance are reduced.
- **Conv2D (Mobilenet with imagenet weights) + RNN (GRU):** This is a cakewalk!
- **Conv2D + RNN (LSTM):** Similar to the GRU model above, but since performance is similar, GRU is preferable to this.