Dataset

100 Sports Classes, 13572 Training Samples, 500 Validation Samples, 500 Test Samples, 224x224 RGB images.

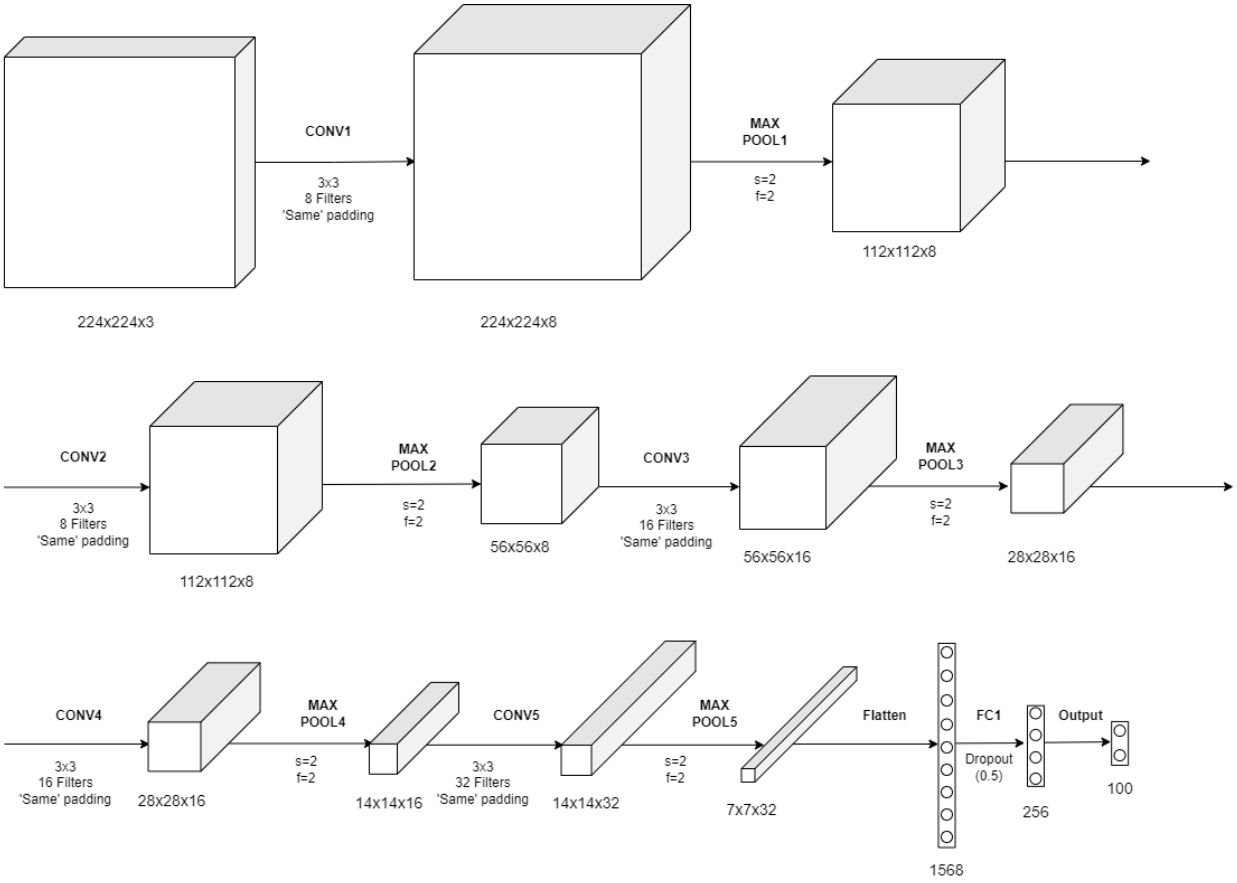
The model with Dropout and Data Augmentation

The second model is run with 5 Conv layers, 5 Max Pooling layers, 1 Fully Connected layer, and 1 Output layer. A dropout (0.5) is also added to the model. The total number of trainable parameters: 436,300

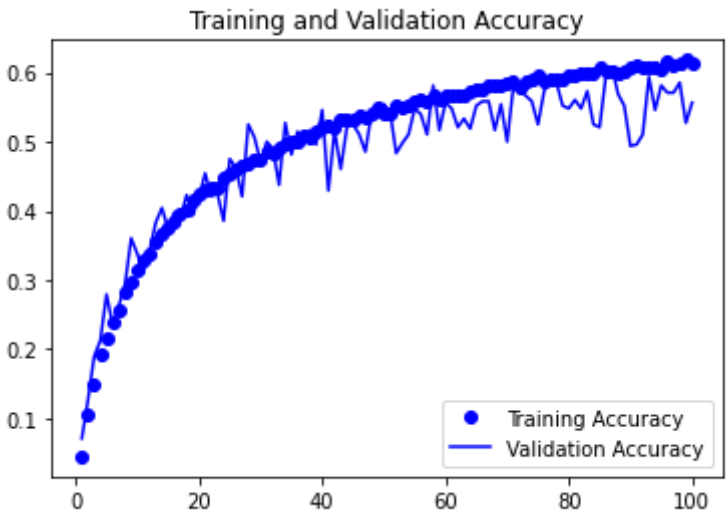
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 224, 224, 8)       224

 max_pooling2d (MaxPooling2D  (None, 112, 112, 8)      0
 )

 conv2d_1 (Conv2D)           (None, 112, 112, 8)       584

 max_pooling2d_1 (MaxPooling  (None, 56, 56, 8)        0
 2D)

 conv2d_2 (Conv2D)           (None, 56, 56, 16)        1168

 max_pooling2d_2 (MaxPooling  (None, 28, 28, 16)       0
 2D)

 conv2d_3 (Conv2D)           (None, 28, 28, 16)        2320

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 16)       0
 2D)

 conv2d_4 (Conv2D)           (None, 14, 14, 32)        4640

 max_pooling2d_4 (MaxPooling  (None, 7, 7, 32)         0
 2D)

 flatten (Flatten)           (None, 1568)              0

 dropout (Dropout)           (None, 1568)              0

 dense (Dense)               (None, 256)               401664

 dense_1 (Dense)             (None, 100)               25700

=================================================================
Total params: 436,300
Trainable params: 436,300
Non-trainable params: 0
_________________________________________________________________
```

Loss: Categorical-Crossentropy
Optimizer: RMSprop with a learning rate of 0.001
Metric: Categorical Accuracy
Train Batch Size: 64, **Validation Batch Size:** 32

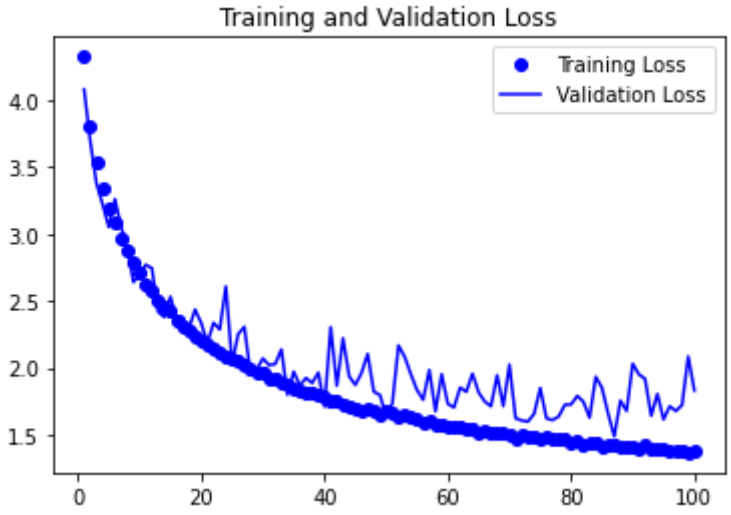
Network Architecture



Train and Validation Accuracy



Train and Validation Loss



Observation & Conclusion

Due to Dropout, Data Augmentation, and simplification of the model (From 56M -> 400k parameters), we are no longer overfitting the training data (At least there is no such severe overfitting) and the training curves are almost closely tracking the validation curves, however, we still couldn't get an improvement in the accuracy of the model.

The dataset is too small to learn a model with large number of parameters. Learning a small CNN is not able to learn good features and doesn’t produce good results.

Solution

A highly effective approach to deal with such a problem with a small dataset is to use transfer learning.