

CNN Image Classification of Formula One Cars

Cole Bennett (cmb170230) and Emily Painter (erp190001)

Using Two Free Days

1 Parameter Tuning Tables	1
1.1 Base Model	1
1.2 Tuning Learning Rate	1
1.3 Tuning Activation Function	2
1.4 Tuning Number of Neurons for One Dense Layer	2
1.5 Tuning Number of Neurons for Two Dense Layers	3
1.6 Tuning Dropout Rate for Two Dense Layers	4
1.7 Fine Tuning with Optimal Hyperparameters	5
2 Performance of Best Model on Test Data	6
3 History Plots	7
3.1 Base Model	7
3.1.1 MobileNetV2	7
3.1.2 ResNetV2	7
3.2 Tuning Learning Rate	8
3.2.1 MobileNetV2	8
3.2.2 ResNetV2	9
3.3 Tuning Activation Function	11
3.3.1 MobileNetV2	11
3.3.2 ResNetV2	12
3.4 Tuning Number of Neurons for One Dense Layer	13
3.4.1 MobileNetV2	13
3.4.2 ResNetV2	15
3.5 Tuning Number of Neurons for Two Dense Layers	16
3.5.1 MobileNetV2	16
3.5.2 ResNetV2	18
3.6 Tuning Dropout Rate for Two Dense Layers	19
3.6.1 MobileNetV2	19
3.6.2 ResNetV2	21
3.7 Fine Tuning with Optimal Hyperparameters	22
3.7.1 MobileNetV2	22
3.7.2 ResNetV2	23

1 Parameter Tuning Tables

All models used Adam optimizer and the categorical cross entropy loss function. The MobileNetV2 models were run for 20 epochs and the ResNetV2 models were run for 10 epochs as they began to demonstrate overfitting with further training. Each model also included three data augmentation layers—a random horizontal flip, random rotation by 0.1, and random zoom by 0.2—to account for our relatively small training sample size. The output layer for all models is a dense layer with eight neurons and softmax activation since our data has eight classes. In the tables below, the given accuracies represent the training and validation accuracies that occurred on the last epoch.

1.1 Base Model

Iteration	Parameters	MobileNetV2 Metrics	ResNetV2 Metrics
1	Learning Rate - 0.0001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.7341 Validation Accuracy: 0.7211	Training Accuracy: 0.8286 Validation Accuracy: 0.7842

1.2 Tuning Learning Rate

Iteration	Parameters	MobileNetV2 Metrics	ResNetV2 Metrics
1	Learning Rate - 0.1 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.1504 Validation Accuracy: 0.1474	Training Accuracy: 0.1522 Validation Accuracy: 0.1316
2	Learning Rate - 0.01 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.1551 Validation Accuracy: 0.1316	Training Accuracy: 0.8187 Validation Accuracy: 0.8211
3	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.7837 Validation Accuracy: 0.7263	Training Accuracy: 0.9160 Validation Accuracy: 0.8316
4	Learning Rate - 0.0001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.7120 Validation Accuracy: 0.6684	Training Accuracy: 0.8315 Validation Accuracy: 0.7947

1.3 Tuning Activation Function

Iteration	Parameters	MobileNetV2 Metrics	ResNetV2 Metrics
1	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.7650	Training Accuracy: 0.8997
		Validation Accuracy: 0.7421	Validation Accuracy: 0.8263
2	Learning Rate - 0.001 Activation Functions - Tanh # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.6519	Training Accuracy: 0.9015
		Validation Accuracy: 0.5895	Validation Accuracy: 0.8211
3	Learning Rate - 0.001 Activation Functions - Sigmoid # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - 0.2	Training Accuracy: 0.7703	Training Accuracy: 0.8618
		Validation Accuracy: 0.7000	Validation Accuracy: 0.7947

1.4 Tuning Number of Neurons for One Dense Layer

Iteration	Parameters	MobileNetV2 Metrics	ResNetV2 Metrics
1	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 1 # of Neurons - 128 Dropout Rate - None	Training Accuracy: 0.7889	Training Accuracy: 0.9090
		Validation Accuracy: 0.6632	Validation Accuracy: 0.8263
2	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 1 # of Neurons - 256 Dropout Rate - None	Training Accuracy: 0.8099	Training Accuracy: 0.9224
		Validation Accuracy: 0.6526	Validation Accuracy: 0.7737
3	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 1 # of Neurons - 512 Dropout Rate - None	Training Accuracy: 0.8187	Training Accuracy: 0.9219
		Validation Accuracy: 0.6632	Validation Accuracy: 0.8000
4	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 1 # of Neurons - 1024 Dropout Rate - None	Training Accuracy: 0.8239	Training Accuracy: 0.9184
		Validation Accuracy: 0.6789	Validation Accuracy: 0.8105

1.5 Tuning Number of Neurons for Two Dense Layers

Iteration	Parameters	MobileNetV2 Metrics	ResNetV2 Metrics
1	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 128 and 64 Dropout Rate - None	Training Accuracy: 0.7907	Training Accuracy: 0.8956
		Validation Accuracy: 0.7053	Validation Accuracy: 0.7737
2	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 256 and 128 Dropout Rate - None	Training Accuracy: 0.8169	Training Accuracy: 0.9172
		Validation Accuracy: 0.6684	Validation Accuracy: 0.8263
3	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 512 and 256 Dropout Rate - None	Training Accuracy: 0.8309	Training Accuracy: 0.9277
		Validation Accuracy: 0.7000	Validation Accuracy: 0.7579
4	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - None	Training Accuracy: 0.8064	Training Accuracy: 0.9195
		Validation Accuracy: 0.6211	Validation Accuracy: 0.8316

1.6 Tuning Dropout Rate for Two Dense Layers

Iteration	Parameters	MobileNetV2 Metrics	ResNetV2 Metrics
1	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - 0.1	Training Accuracy: 0.7994	Training Accuracy: 0.9096
		Validation Accuracy: 0.7158	Validation Accuracy: 0.8105
2	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - 0.2	Training Accuracy: 0.7907	Training Accuracy: 0.9085
		Validation Accuracy: 0.7263	Validation Accuracy: 0.8474
3	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - 0.33	Training Accuracy: 0.7341	Training Accuracy: 0.8828
		Validation Accuracy: 0.7158	Validation Accuracy: 0.8158
4	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - 0.5	Training Accuracy: 0.6286	Training Accuracy: 0.8641
		Validation Accuracy: 0.6474	Validation Accuracy: 0.8105

**Note: Dropout Rate of 0.33 selected for ResNetV2 due to more stable performance across previous epochs*

1.7 Fine Tuning with Optimal Hyperparameters

Performance of the MobileNetV2 vs ResNetV2 are separated into two tables since each used slightly different optimal hyperparameters.

MobileNetV2

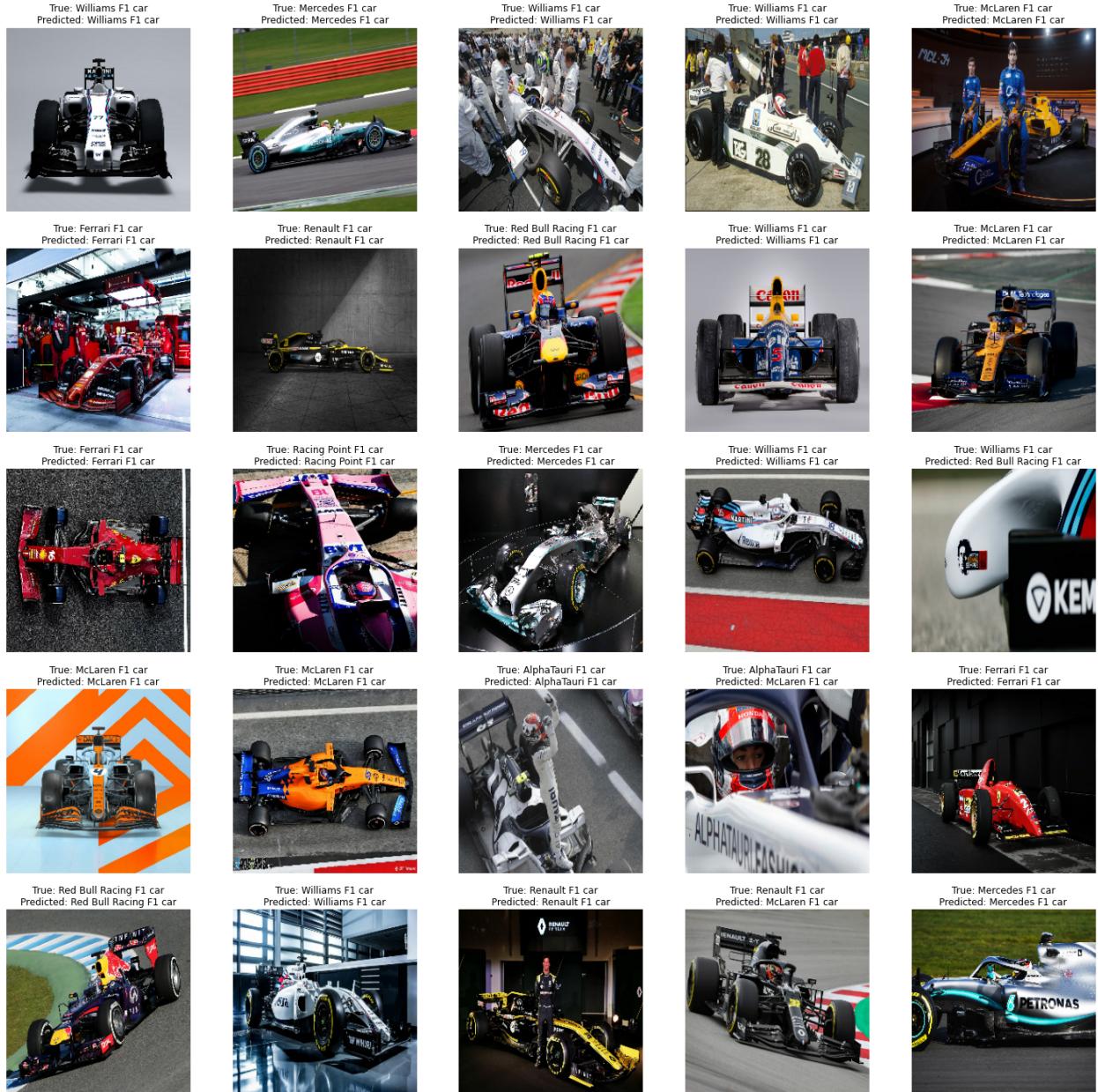
Iteration	Parameters	MobileNetV2 Metrics
1	Learning Rate - 0.001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - 0.2	Training Accuracy: 0.7971 Validation Accuracy: 0.7263

ResNetV2

Iteration	Parameters	ResNetV2 Metrics
1	Learning Rate - 0.00001 Activation Functions - Relu # of Hidden Dense Layers - 2 # of Neurons - 1024 and 512 Dropout Rate - 0.33 Epochs - 15	Training Accuracy: 0.9341 Validation Accuracy: 0.8895

ResNetV2 was run for more epochs than previous models due to validation accuracy still increasing monotonically, indicating a lack of overfitting with more training.

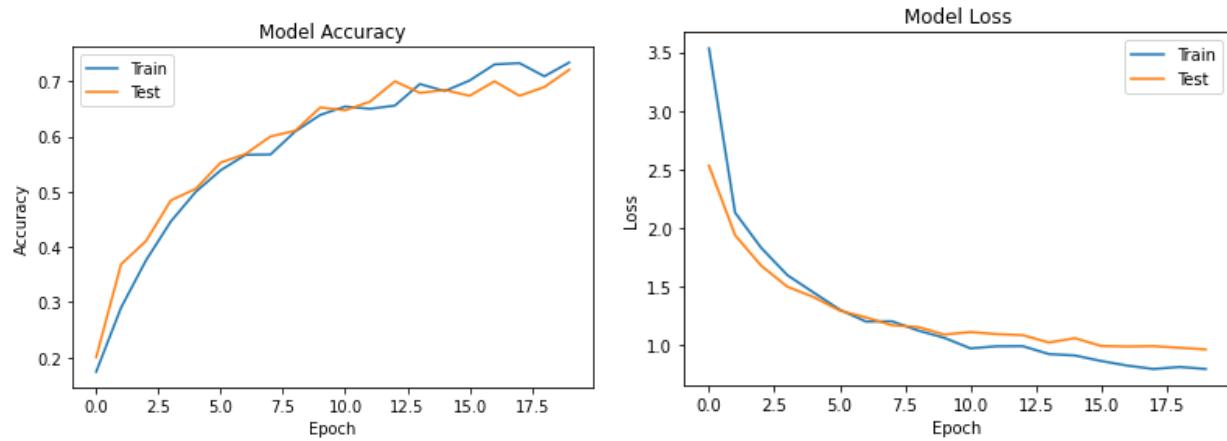
2 Performance of Best Model on Test Data



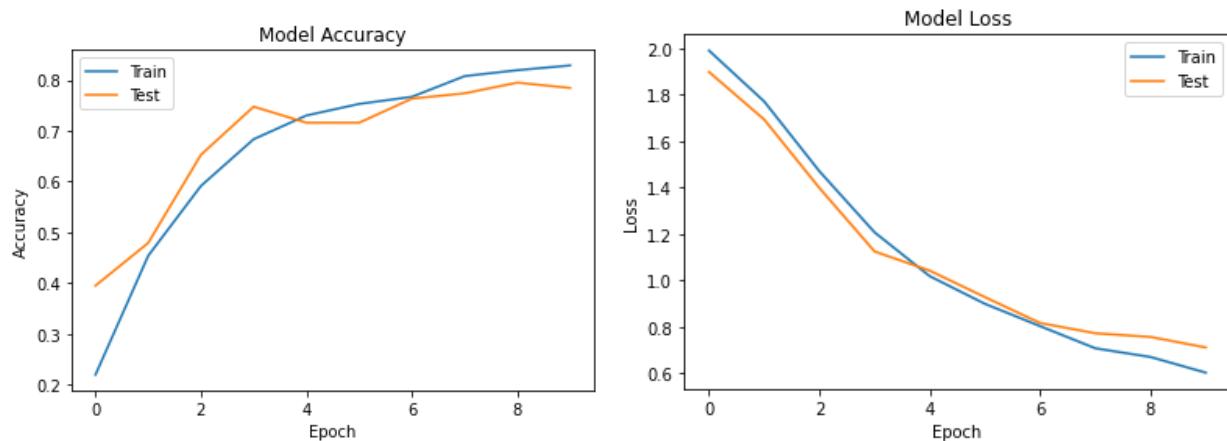
3 History Plots

3.1 Base Model

3.1.1 MobileNetV2



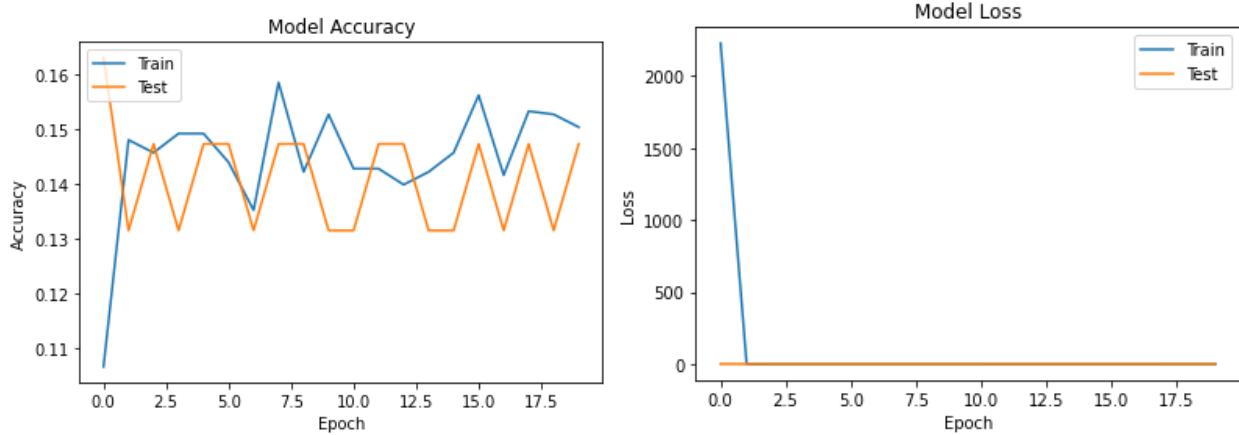
3.1.2 ResNetV2



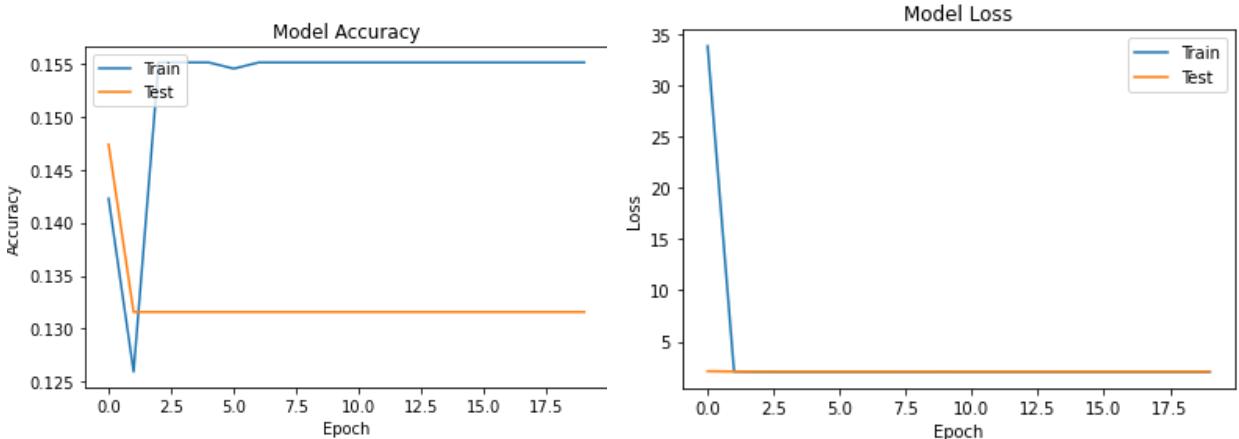
3.2 Tuning Learning Rate

3.2.1 MobileNetV2

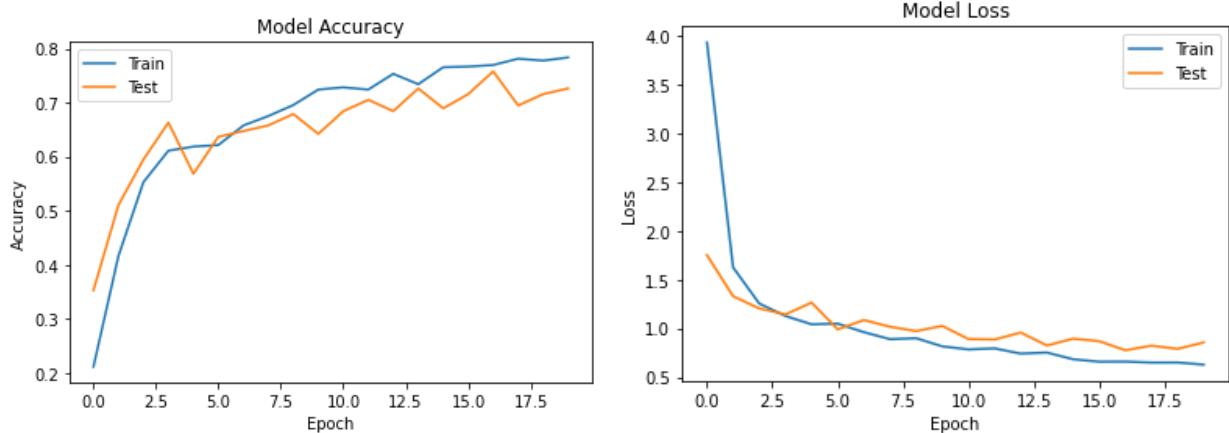
```
learning_rate = 0.1
```



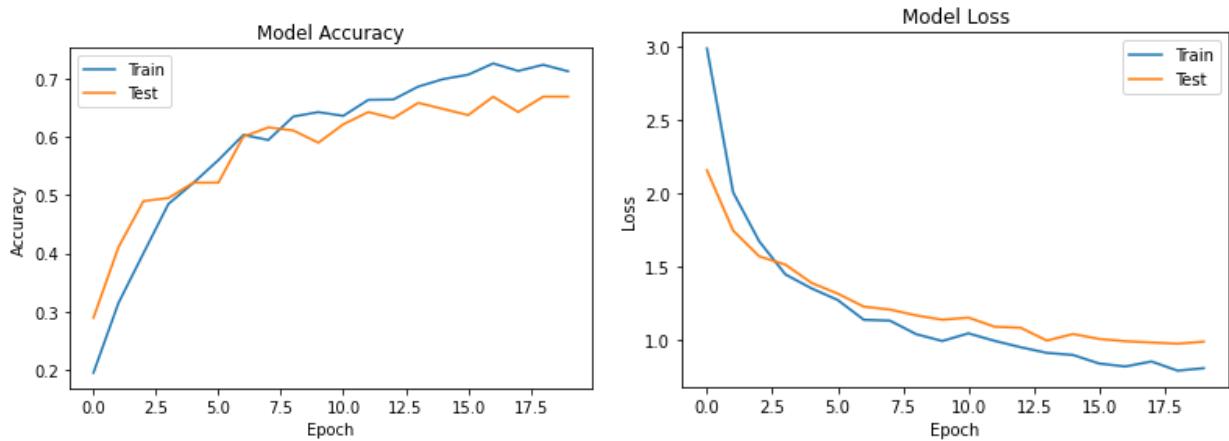
```
learning_rate = 0.01
```



`learning_rate = 0.001`

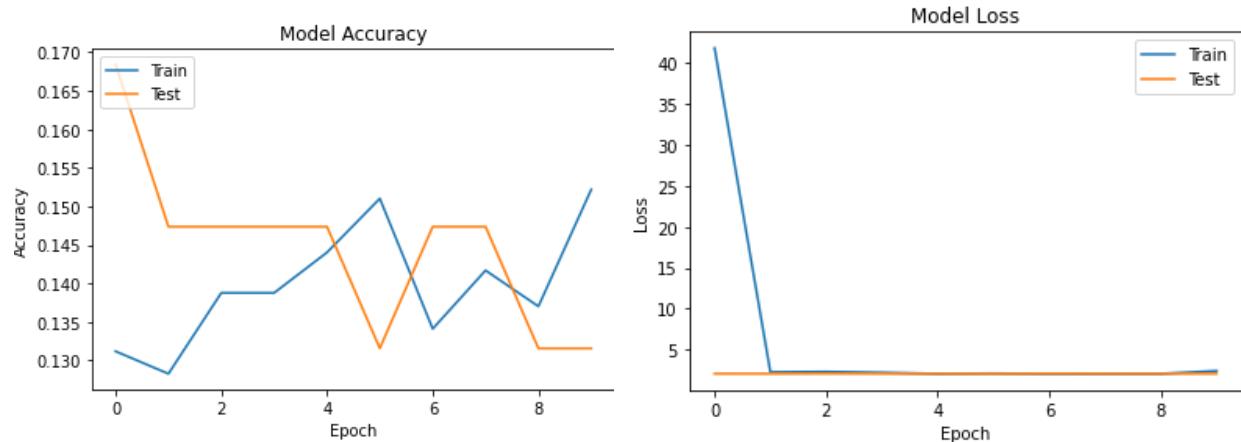


`learning_rate = 0.0001`

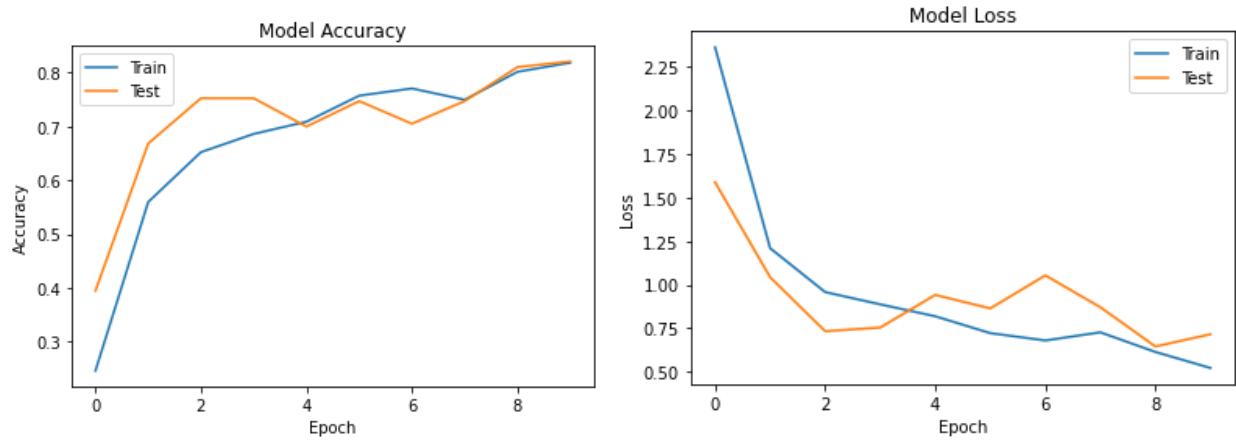


3.2.2 ResNetV2

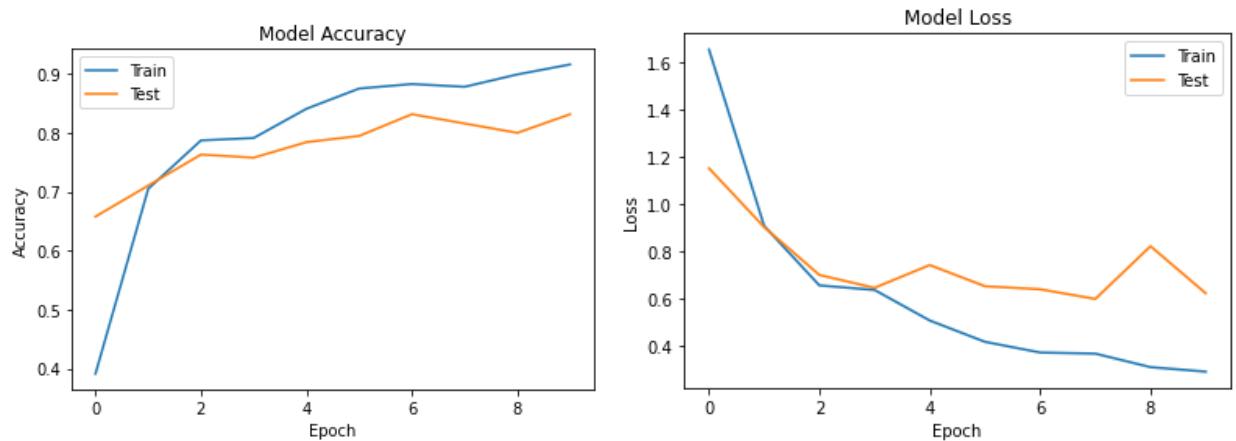
`learning_rate = 0.1`



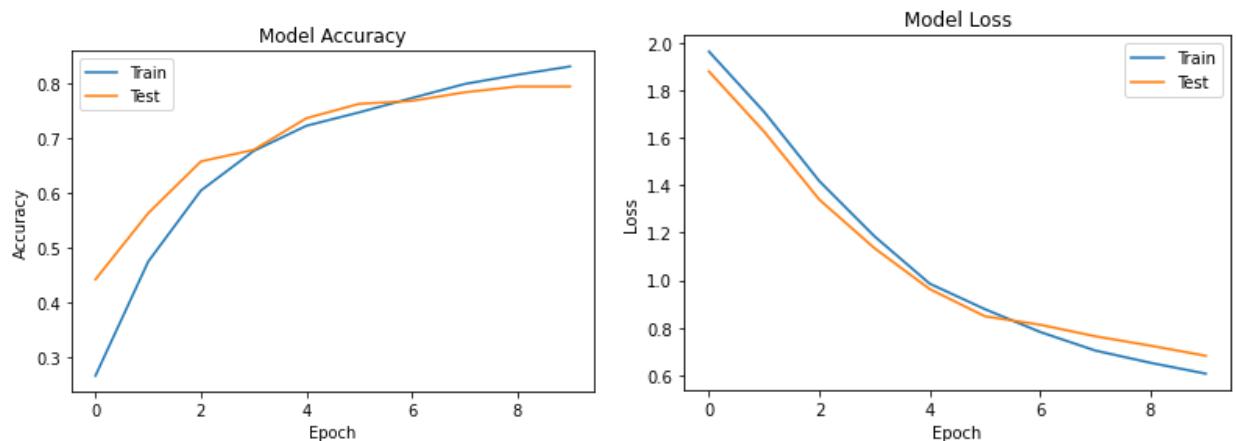
`learning_rate = 0.01`



`learning_rate = 0.001`



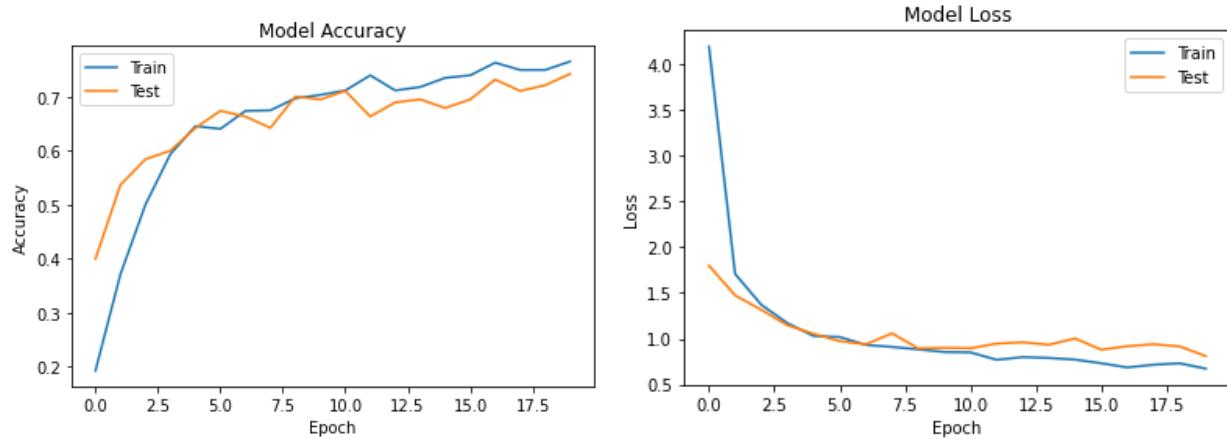
`learning_rate = 0.0001`



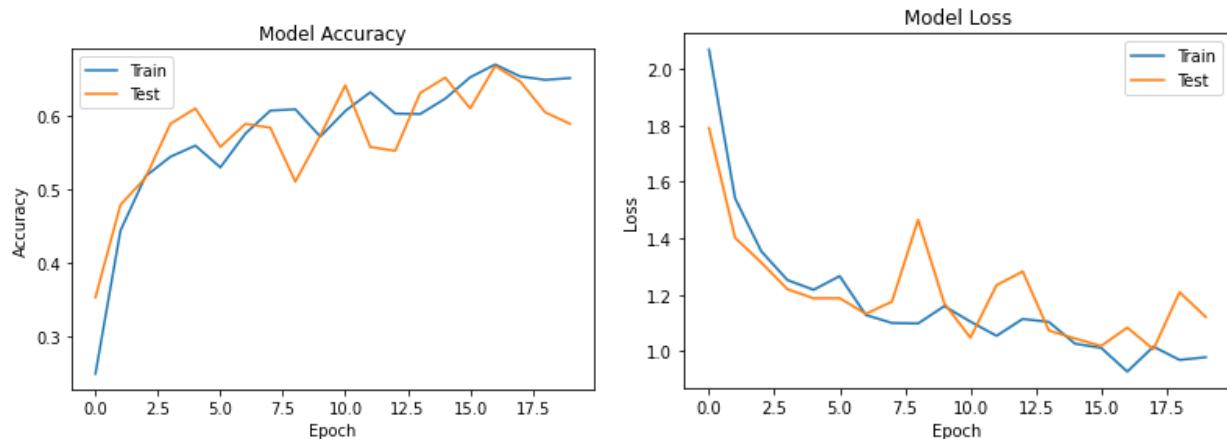
3.3 Tuning Activation Function

3.3.1 MobileNetV2

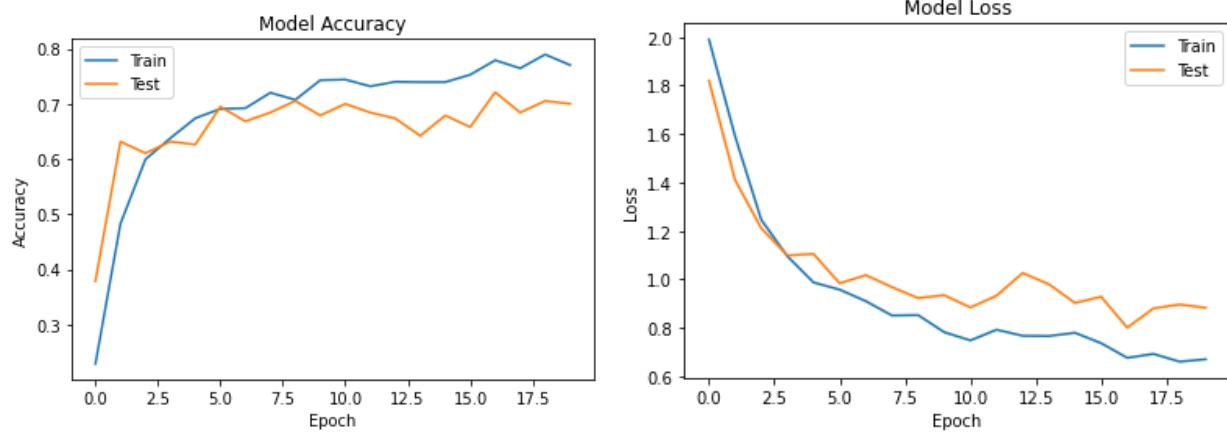
`activation_function = 'relu'`



`activation_function = 'tanh'`

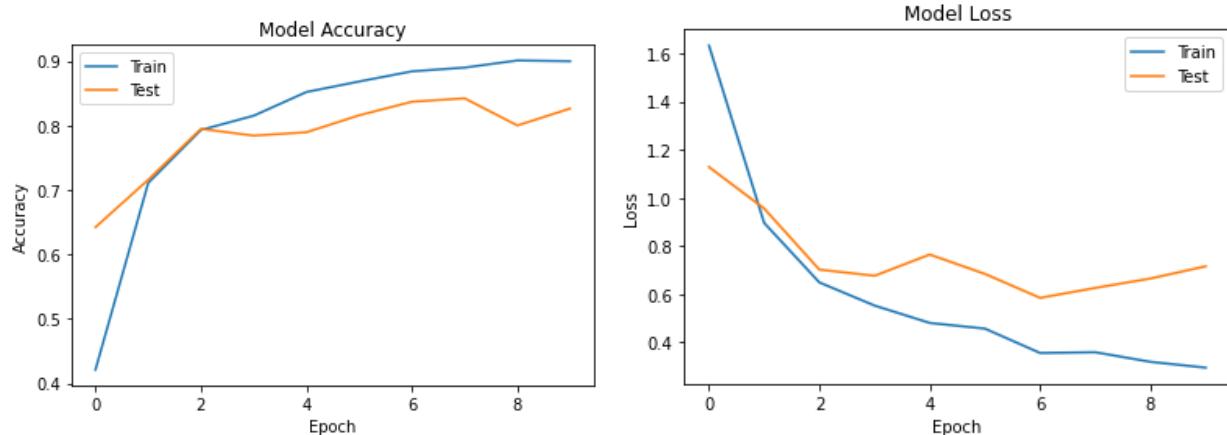


```
activation_function = 'sigmoid'
```

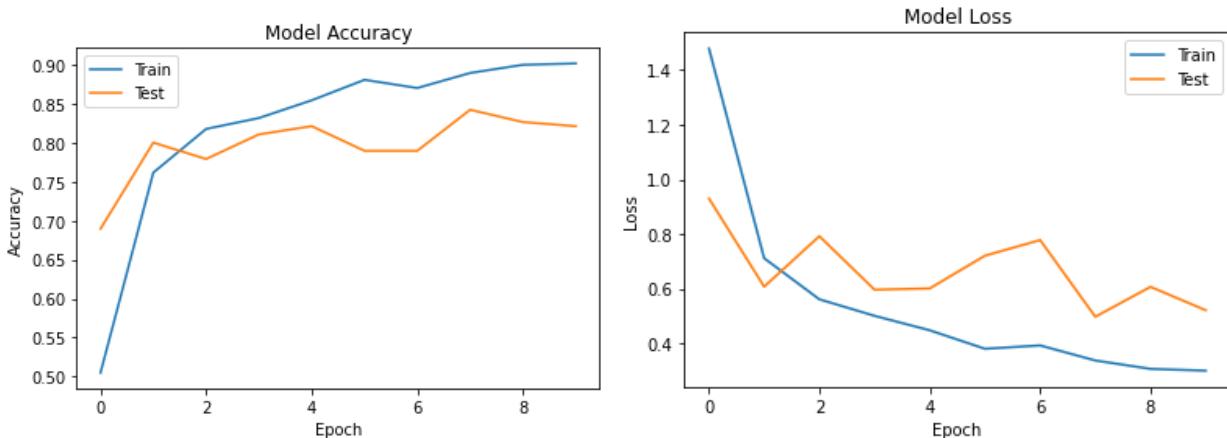


3.3.2 ResNetV2

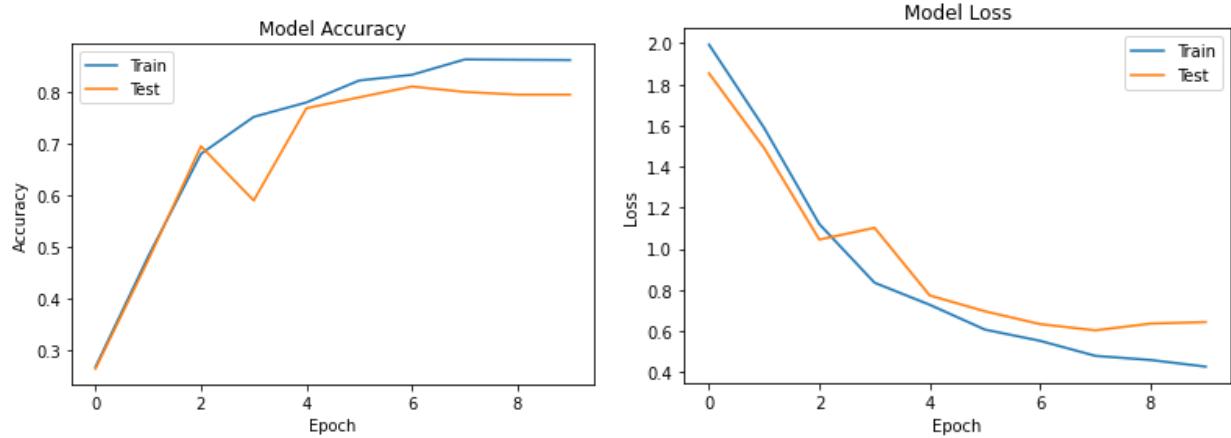
```
activation_function = 'relu'
```



```
activation_function = 'tanh'
```



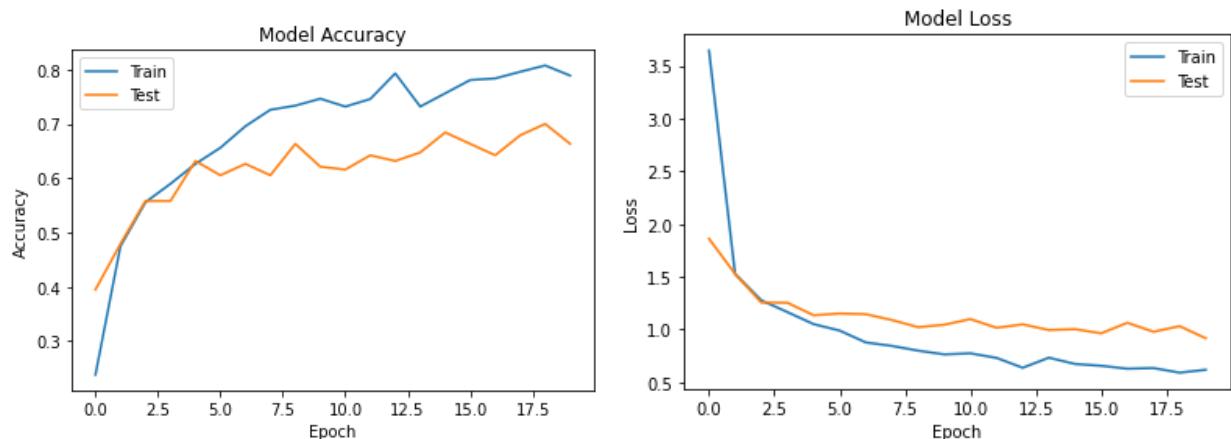
```
activation_function = 'sigmoid'
```



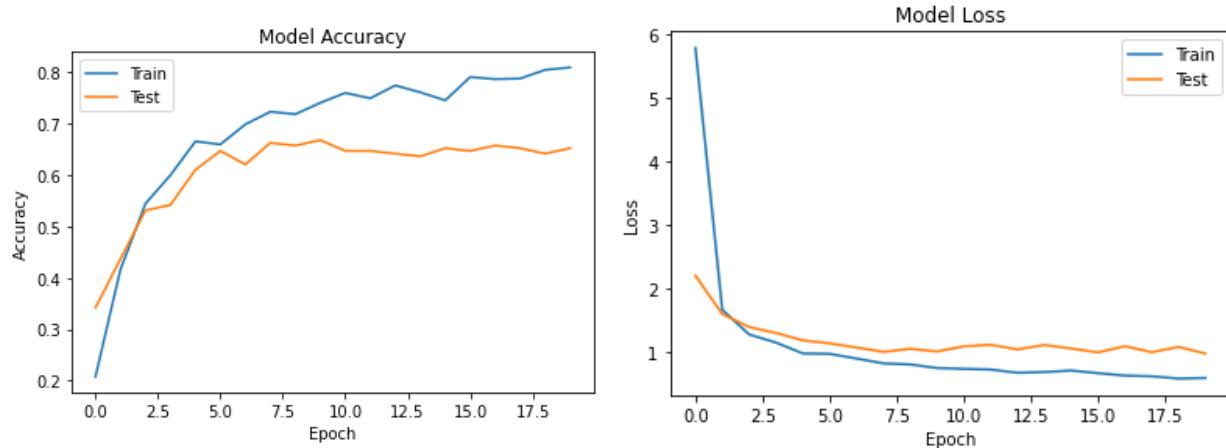
3.4 Tuning Number of Neurons for One Dense Layer

3.4.1 MobileNetV2

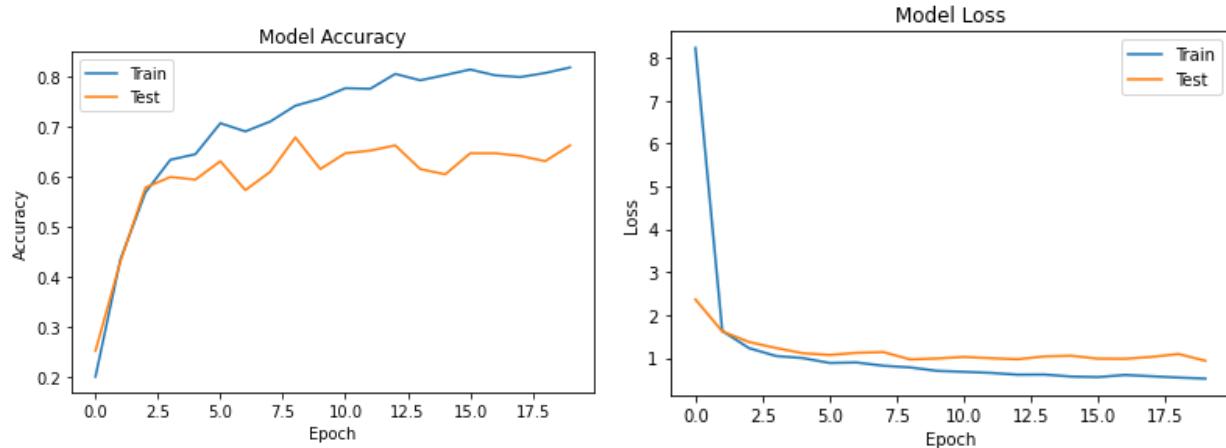
```
numNeurons = 128
```



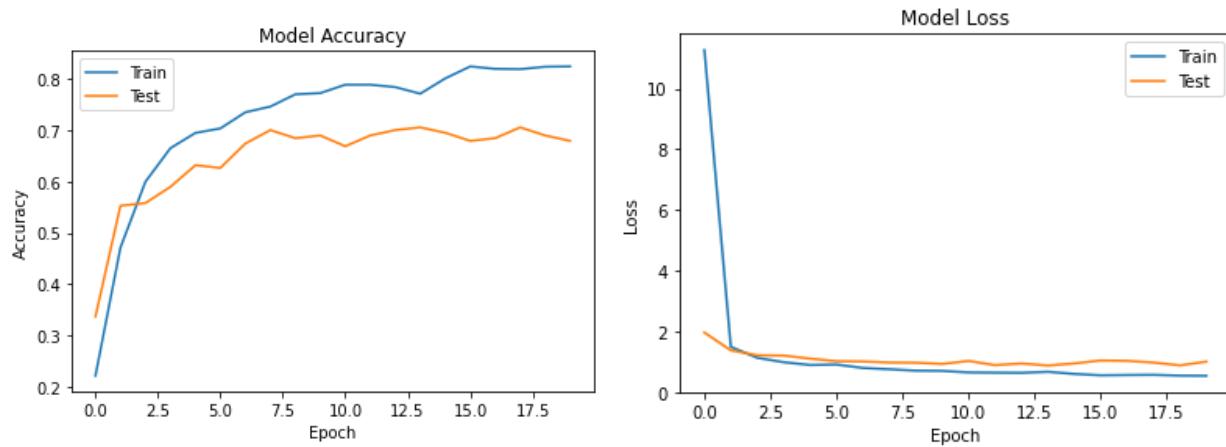
`numNeurons = 256`



`numNeurons = 512`

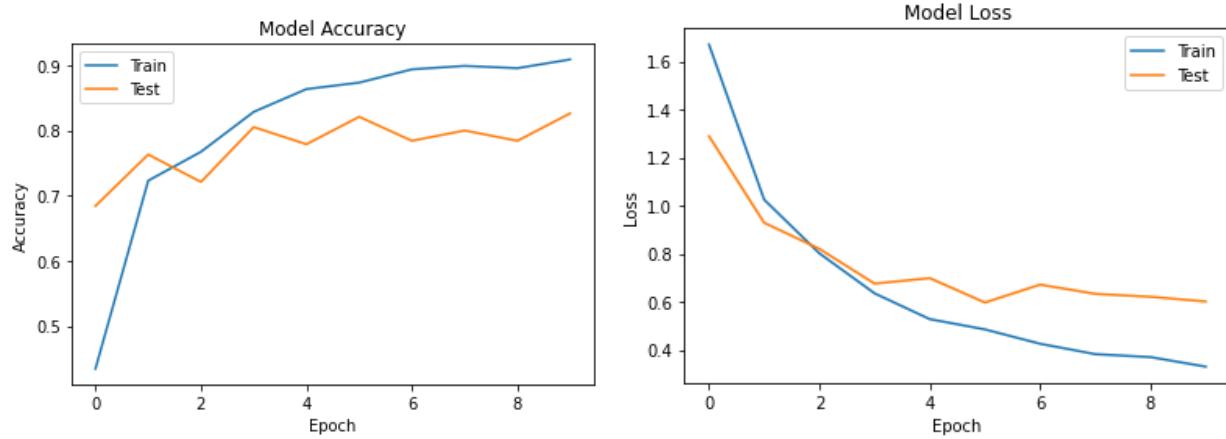


`numNeurons = 1024`

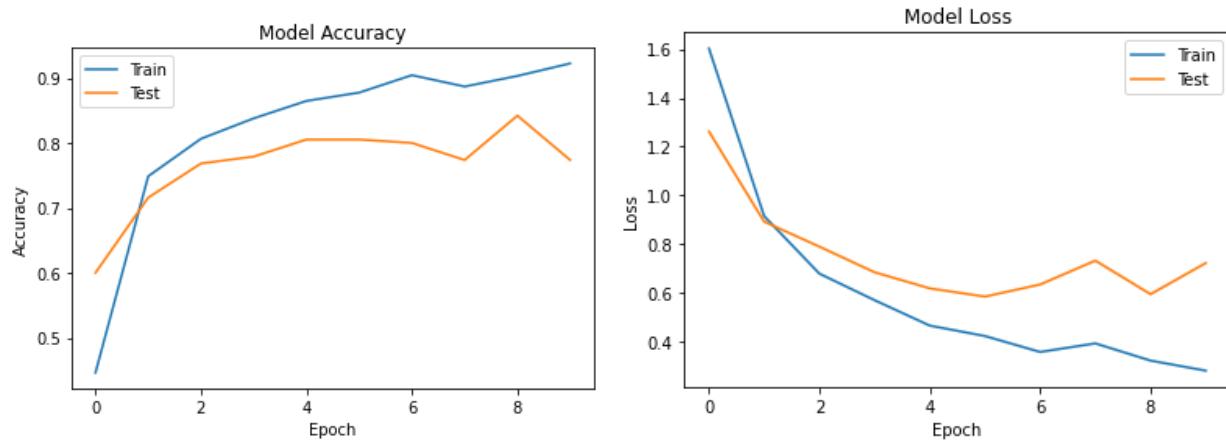


3.4.2 ResNetV2

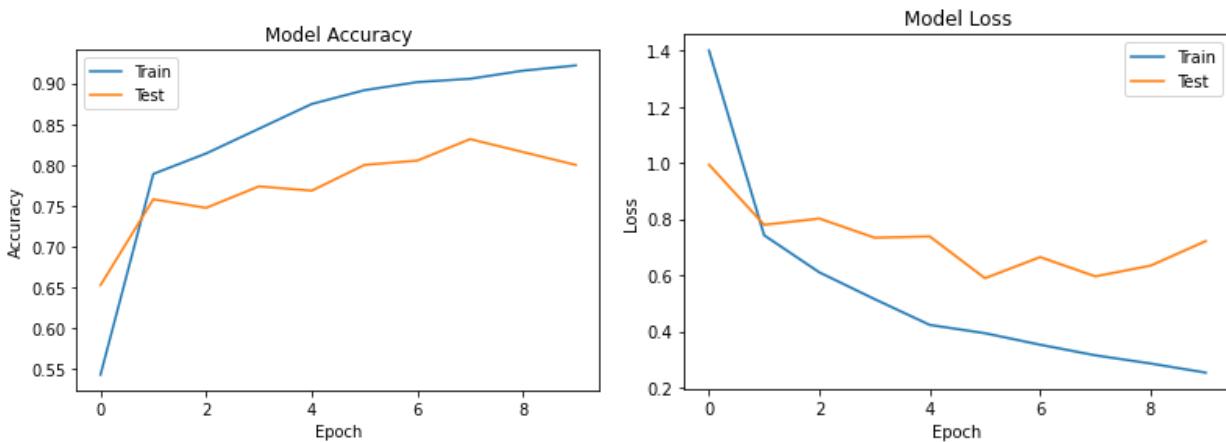
`numNeurons = 128`



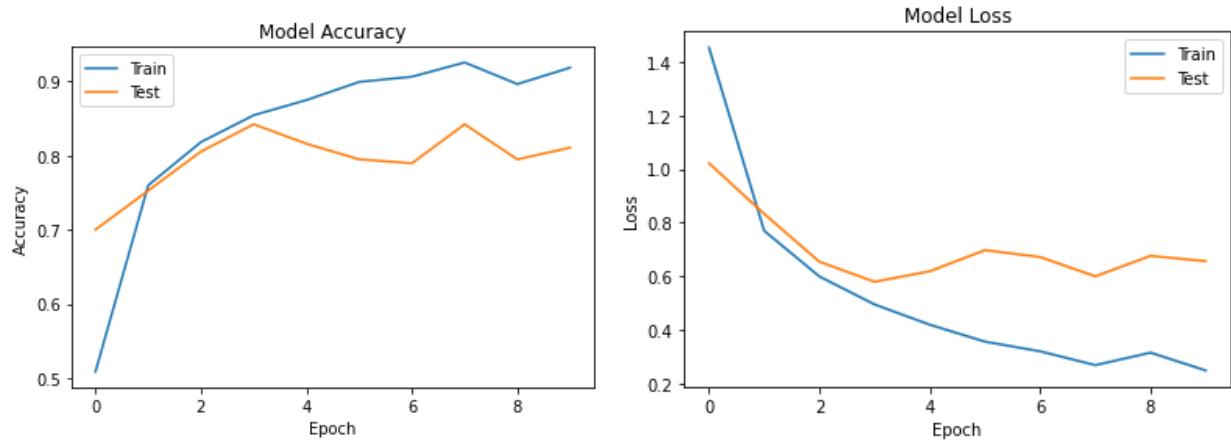
`numNeurons = 256`



`numNeurons = 512`



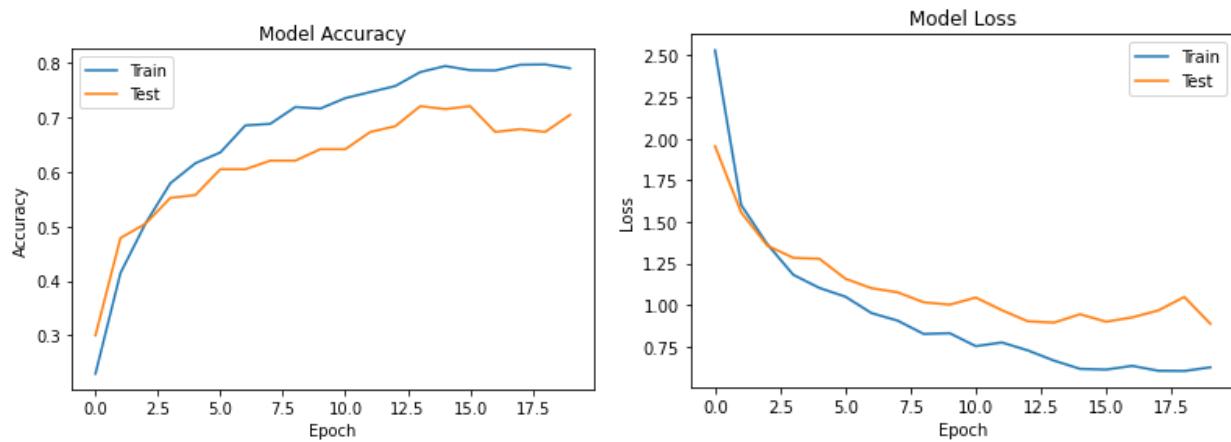
`numNeurons = 1024`



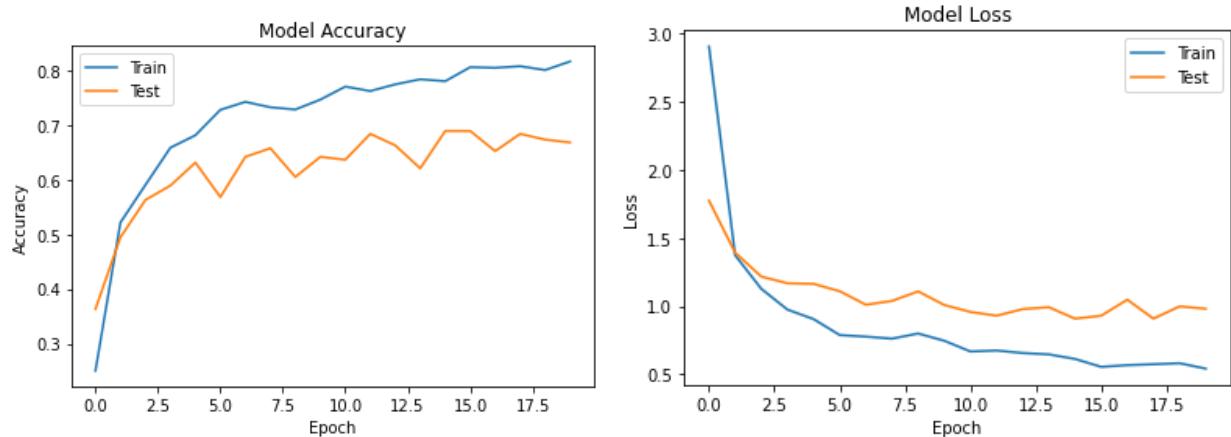
3.5 Tuning Number of Neurons for Two Dense Layers

3.5.1 MobileNetV2

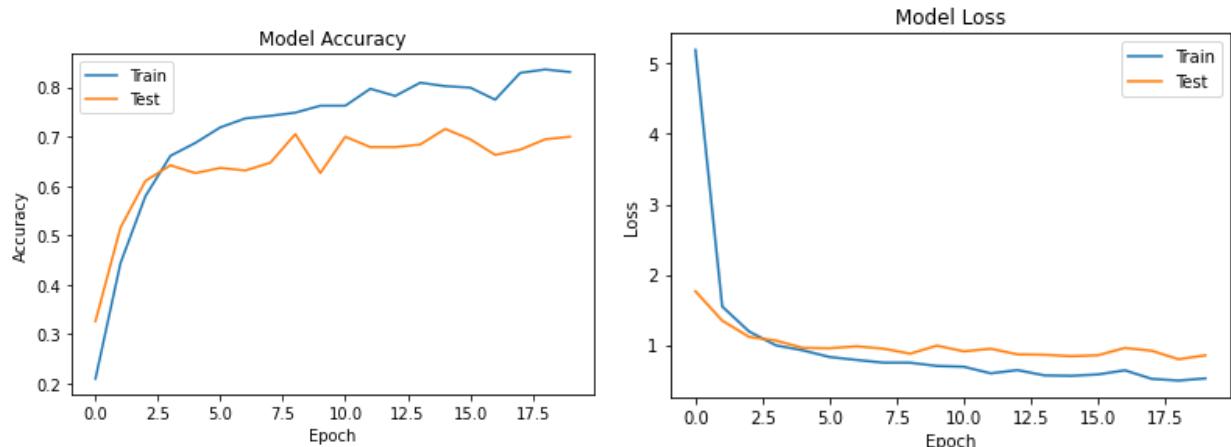
`numNeurons = 128`



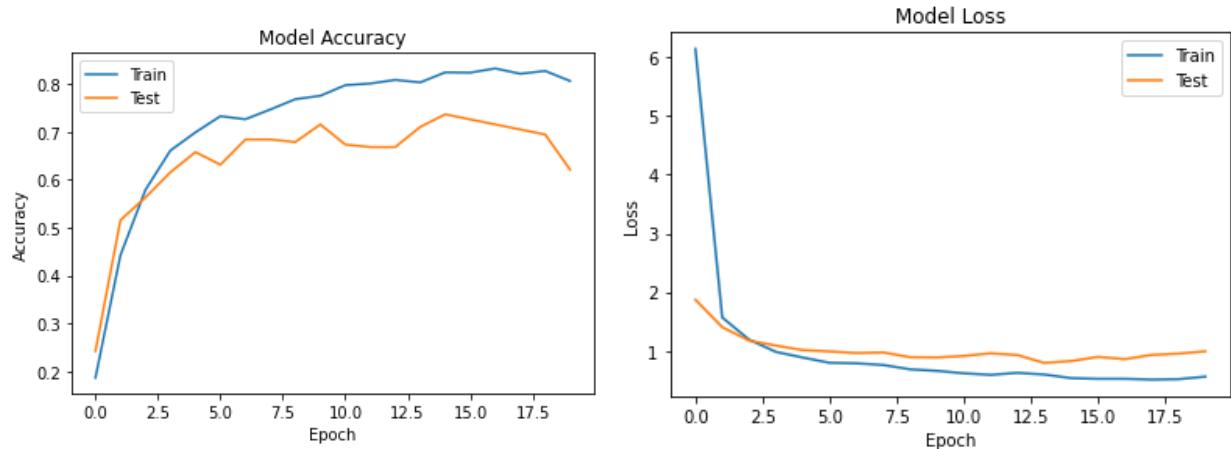
`numNeurons = 256`



`numNeurons = 512`

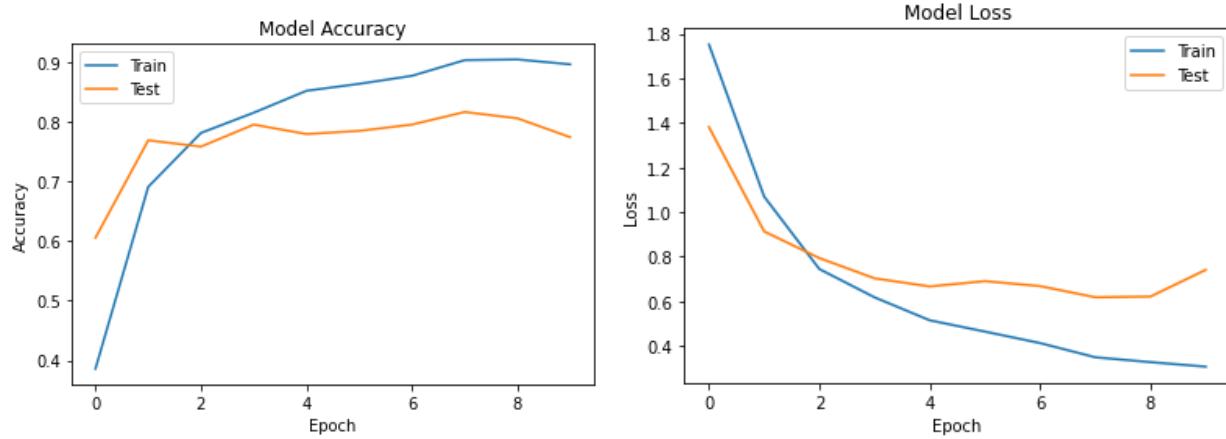


`numNeurons = 1024`

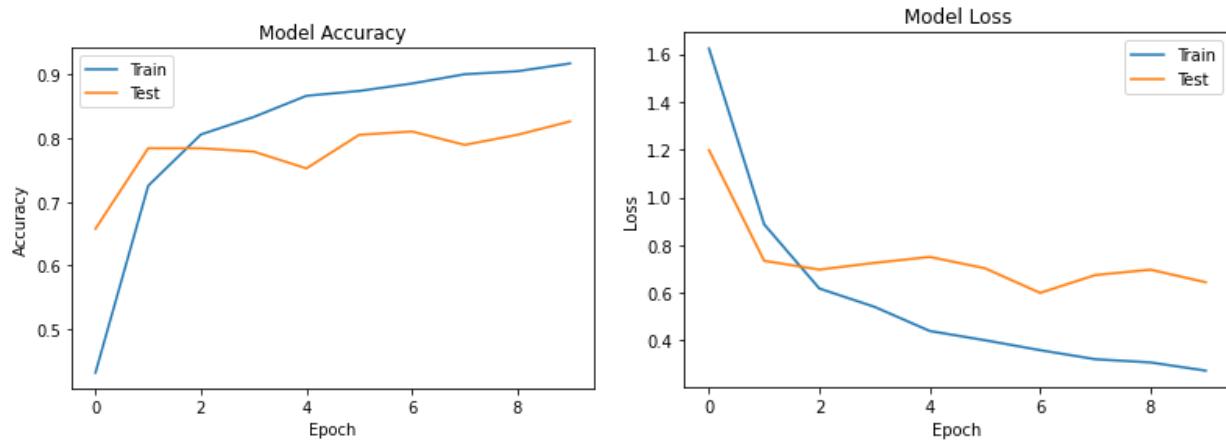


3.5.2 ResNetV2

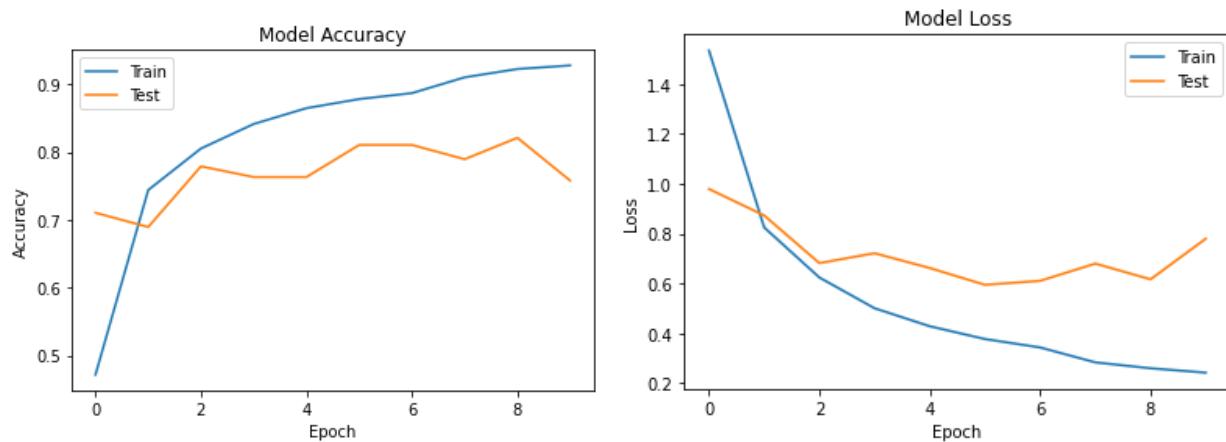
`numNeurons = 128`



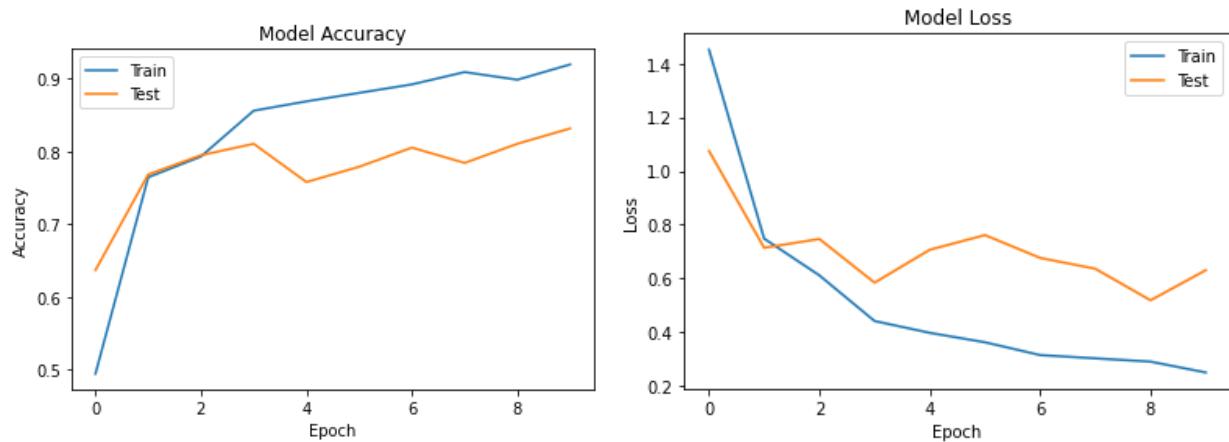
`numNeurons = 256`



`numNeurons = 512`



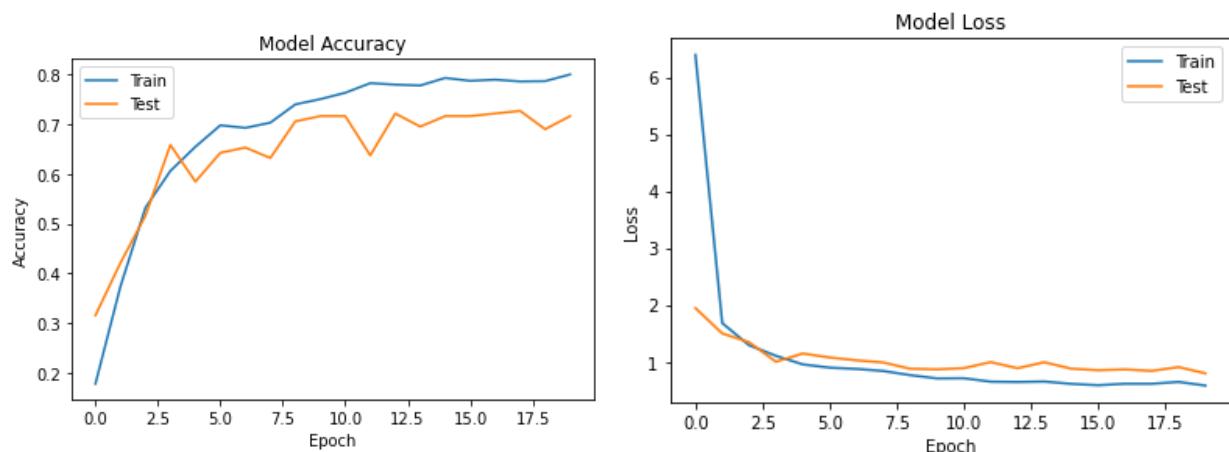
```
numNeurons = 1024
```



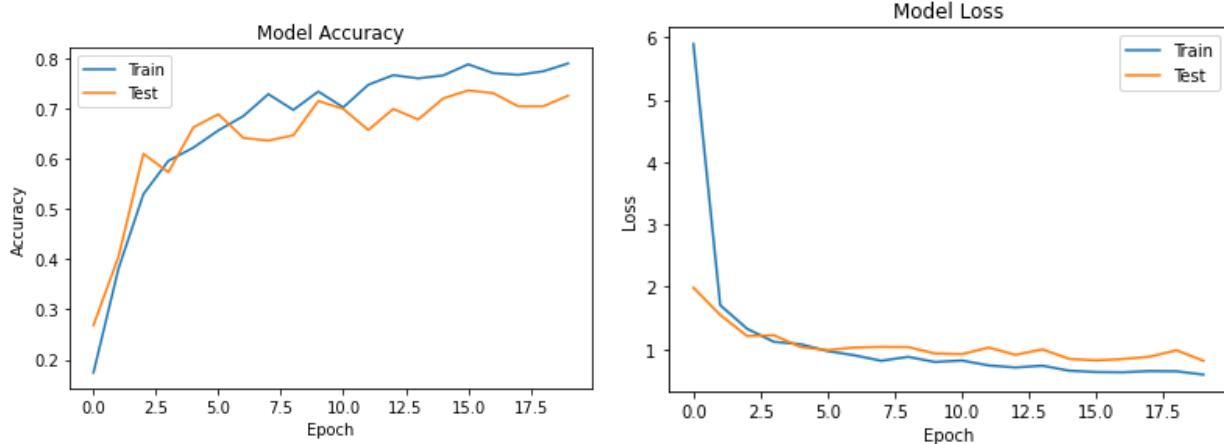
3.6 Tuning Dropout Rate for Two Dense Layers

3.6.1 MobileNetV2

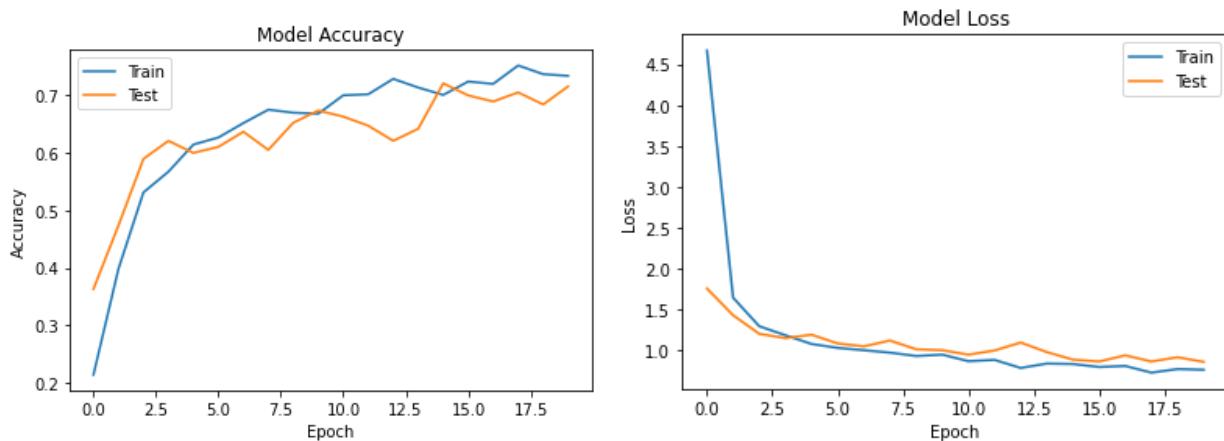
```
dropout_rate = 0.1
```



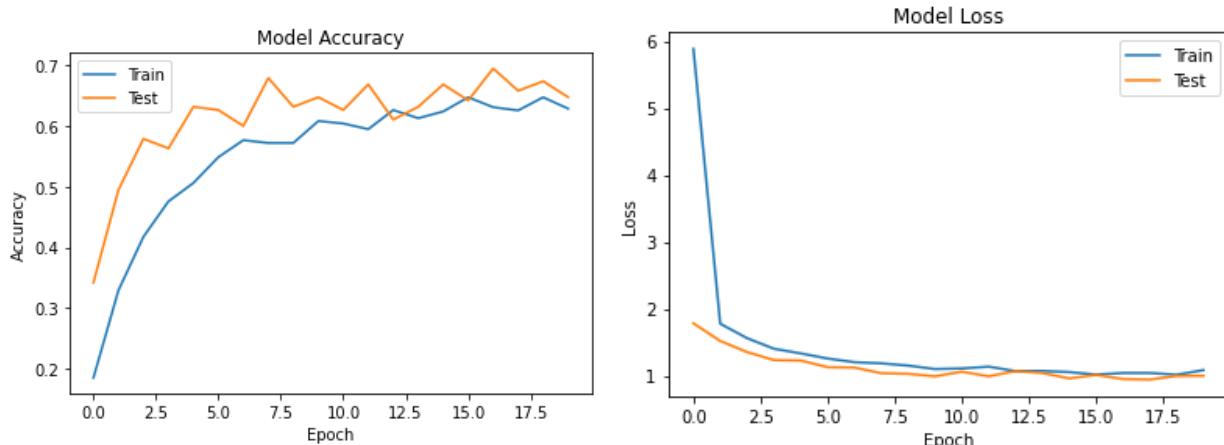
`dropout_rate = 0.2`



`dropout_rate = 0.33`

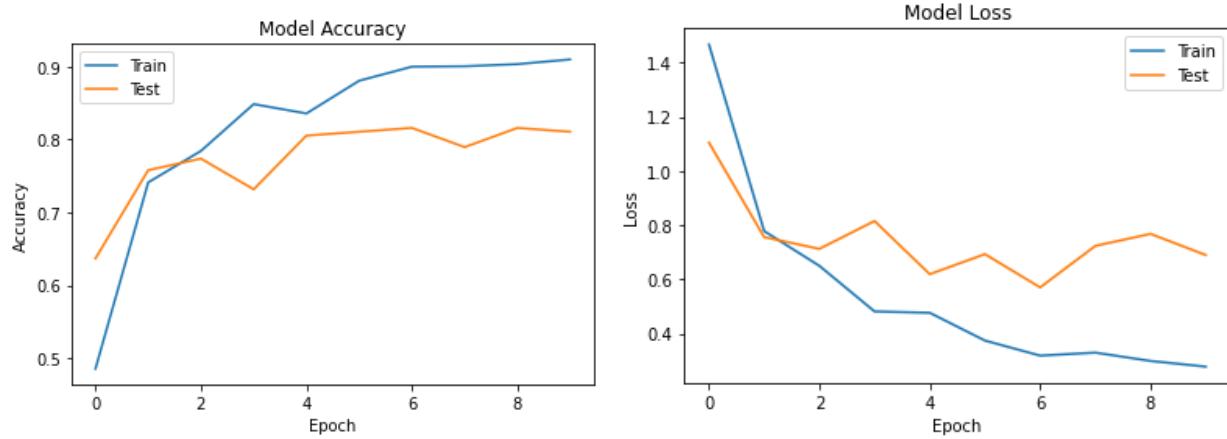


`dropout_rate = 0.5`

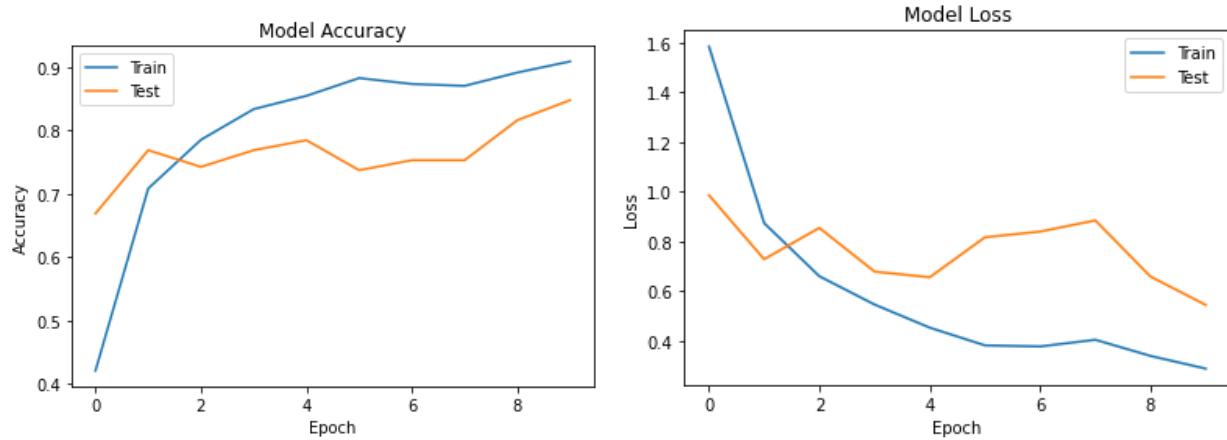


3.6.2 ResNetV2

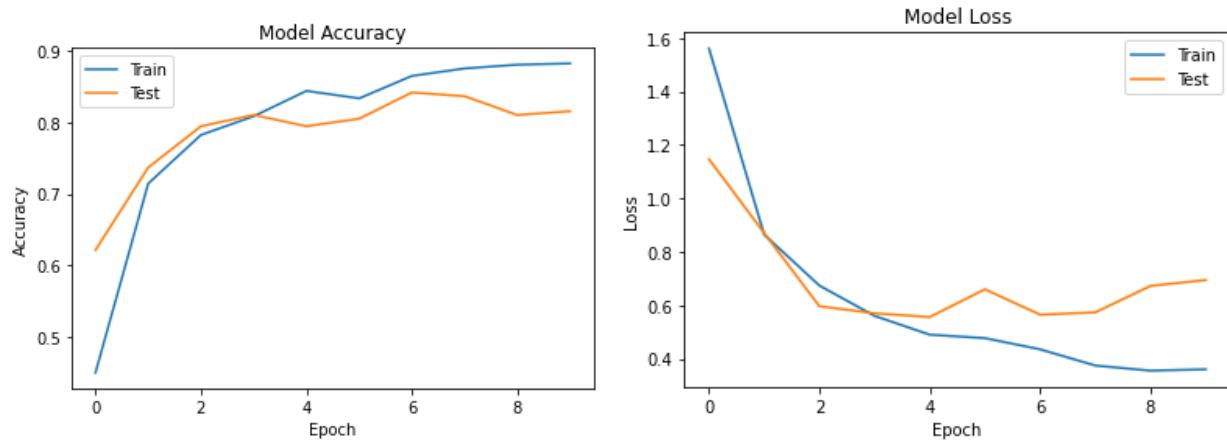
`dropout_rate = 0.1`



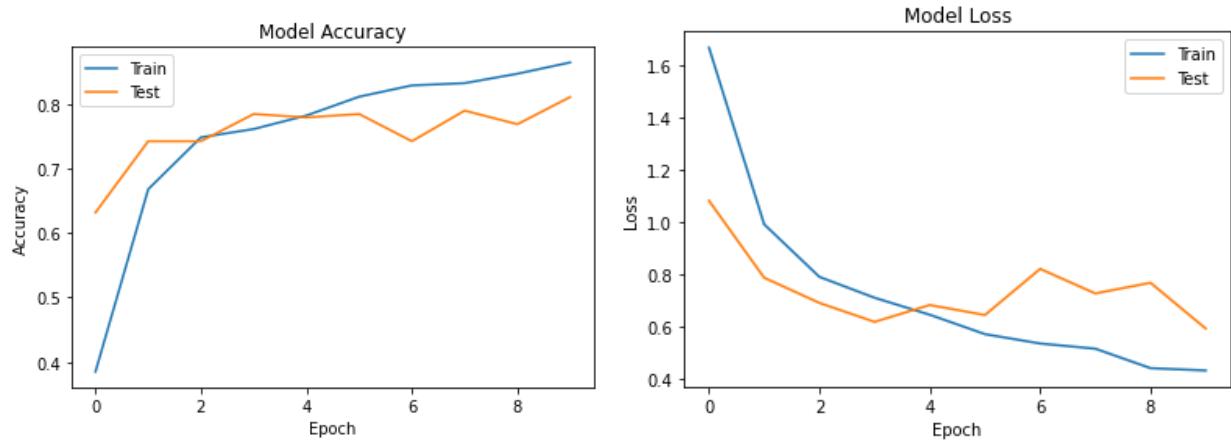
`dropout_rate = 0.2`



`dropout_rate = 0.33`

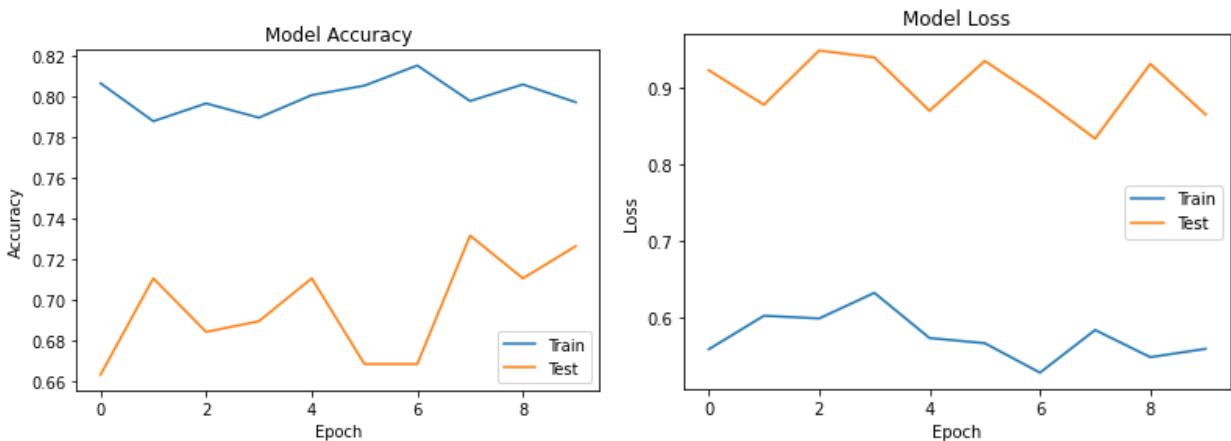


`dropout_rate = 0.5`



3.7 Fine Tuning with Optimal Hyperparameters

3.7.1 MobileNetV2



3.7.2 ResNetV2

