

## Transfer Learning Dog Breed Classification

---

Connor Kordes - JCK160130  
Nathan Tompkins - NAT180002  
University of Texas at Dallas  
CS 4372

|   |           |
|---|-----------|
| <b>Transfer Learning Dog Breed Classification</b> | <b>1</b>  |
| <b>History Plots and Models</b>                   | <b>2</b>  |
| Model 1:  | 2         |
| Model 2   | 3         |
| Model 3   | 4         |
| Model 4   | 5         |
| Model 5   | 6         |
| Model 6   | 7         |
| <b>Sample Predictions of Model 1</b>              | <b>8</b>  |
| <b>Hyperparameter Tuning</b>                      | <b>10</b> |

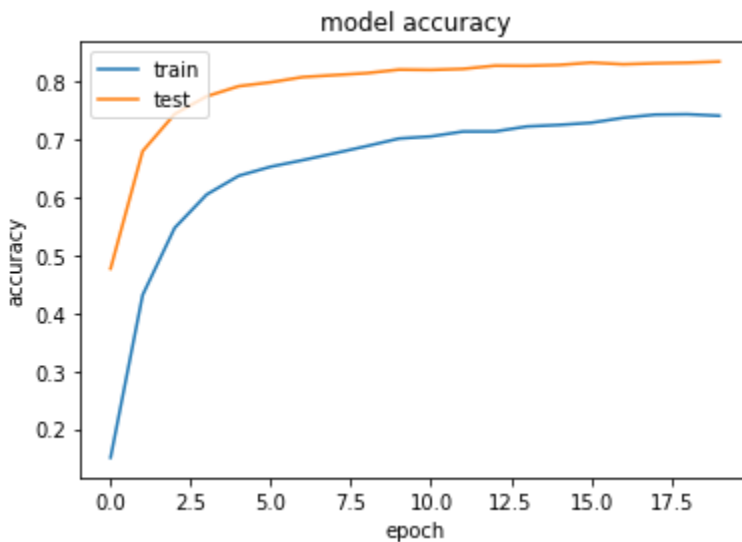
## History Plots and Models

In all of the below imagenet2 is the base\_model with include\_top set to false.

### Model 1:

```
model = tf.keras.Sequential()  
model.add(tf.keras.layers.RandomFlip('horizontal'))  
model.add(tf.keras.layers.RandomRotation(0.2))  
model.add(rescale)  
model.add(base_model)  
model.add(global_average_layer)  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(120, activation='softmax'))
```

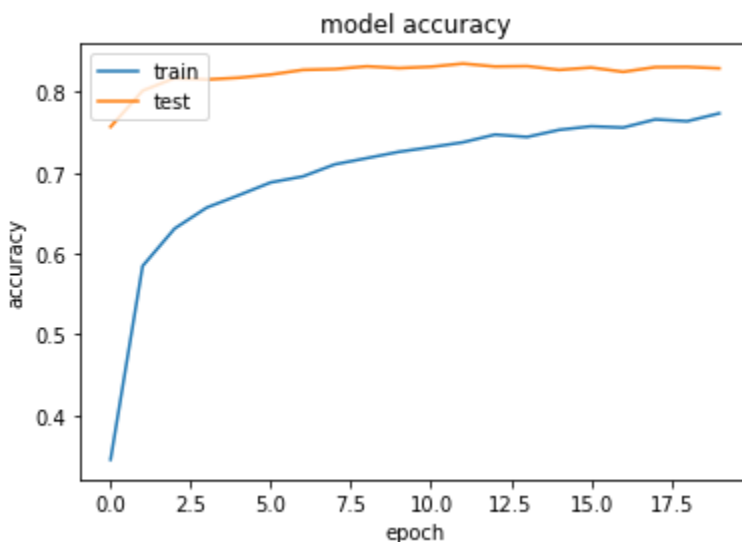
```
base_learning_rate = 0.0001  
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),  
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
              metrics=['accuracy'])
```



## Model 2

```
model = tf.keras.Sequential()  
model.add(tf.keras.layers.RandomFlip('horizontal'))  
model.add(tf.keras.layers.RandomRotation(0.2))  
model.add(rescale)  
model.add(base_model)  
model.add(global_average_layer)  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(1000, activation='relu'))  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(120, activation='softmax'))
```

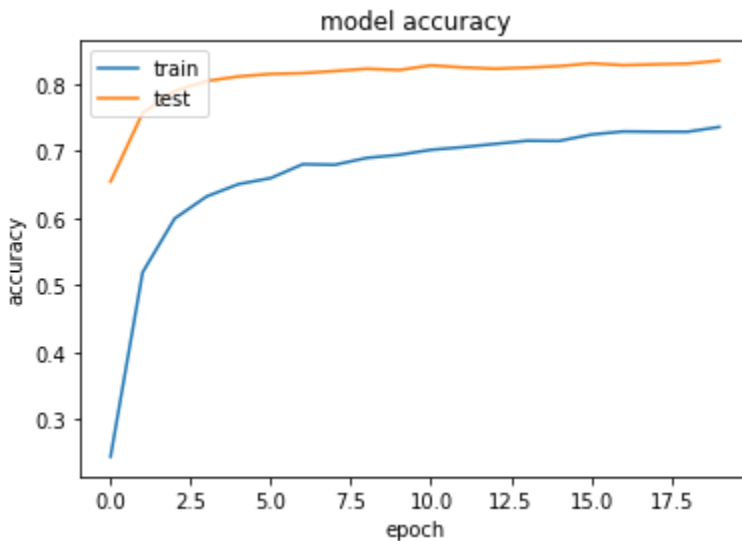
```
base_learning_rate = 0.0001  
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),  
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
              metrics=['accuracy'])
```



### Model 3

```
model = tf.keras.Sequential()  
model.add(tf.keras.layers.RandomFlip('horizontal'))  
model.add(tf.keras.layers.RandomRotation(0.2))  
model.add(rescale)  
model.add(base_model)  
model.add(global_average_layer)  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(1000, activation='sigmoid'))  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(120, activation='softmax'))
```

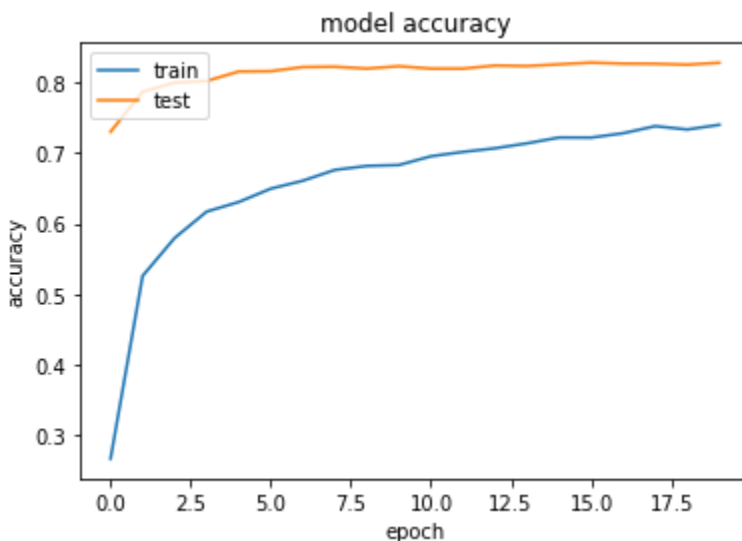
```
base_learning_rate = 0.0001  
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),  
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
              metrics=['accuracy'])
```



## Model 4

```
model = tf.keras.Sequential()  
model.add(tf.keras.layers.RandomFlip('horizontal'))  
model.add(tf.keras.layers.RandomRotation(0.2))  
model.add(rescale)  
model.add(base_model)  
model.add(global_average_layer)  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(1000, activation='relu'))  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(500, activation='relu'))  
model.add(tf.keras.layers.Dropout(0.2))  
model.add(tf.keras.layers.Dense(120, activation='softmax'))
```

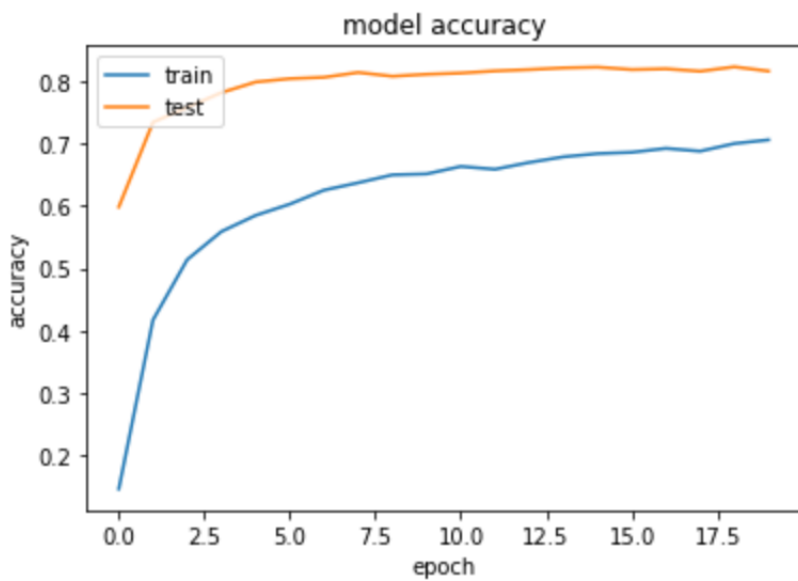
```
base_learning_rate = 0.0001  
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),  
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),  
              metrics=['accuracy'])
```



## Model 5

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.RandomFlip('horizontal'))
model.add(tf.keras.layers.RandomRotation(0.2))
model.add(rescale)
model.add(base_model)
model.add(global_average_layer)
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(1000, activation='sigmoid'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(500, activation='relu'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(120, activation='softmax'))
```

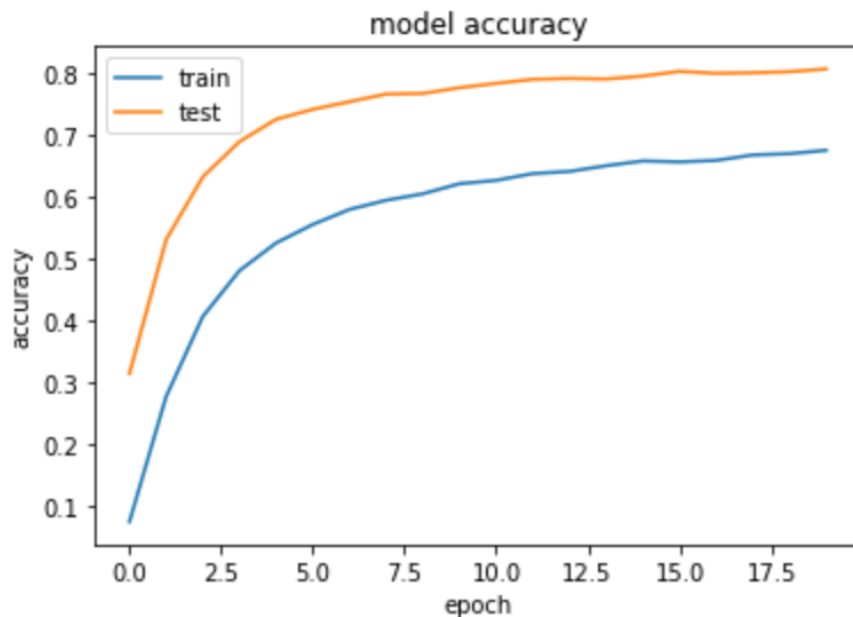
```
base_learning_rate = 0.0001
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
              metrics=['accuracy'])
```



## Model 6

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.RandomFlip('horizontal'))
model.add(tf.keras.layers.RandomRotation(0.2))
model.add(rescale)
model.add(base_model)
model.add(global_average_layer)
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(1000, activation='sigmoid'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(1000, activation='sigmoid'))
model.add(tf.keras.layers.Dropout(0.2))
model.add(tf.keras.layers.Dense(120, activation='softmax'))
```

```
base_learning_rate = 0.0001
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
              metrics=['accuracy'])
```



# Sample Predictions of Model 1

n02088238-basset 36% (n02088238-basset)



n02116738-African\_hunting\_dog 99% (n02116738-African\_hunting\_dog)



n02098105-soft-coated\_wheaten\_terrier 86% (n02098105-soft-coated\_wheaten\_terrier)



n02109525-Saint\_Bernard 99% (n02109525-Saint\_Bernard)



n02088632-bluetick 87% (n02088632-bluetick)



n02091244-ibizan\_hound 98% (n02091244-ibizan\_hound)



n02108915-French\_bulldog 98% (n02108915-French\_bulldog)



n02096177-cairn 59% (n02110627-affenpinscher)



n02112137-chow 90% (n02112137-chow)



n02094114-Norfolk\_terrier 77% (n02094114-Norfolk\_terrier)



n02115913-dhole 79% (n02115913-dhole)



n02087046-toy\_terrier 80% (n02087046-toy\_terrier)



n02089973-English\_foxhound 29% (n02089867-Walker\_hound)



n02113712-miniature\_poodle 51% (n02113712-miniature\_poodle)



n02092002-Scottish\_deerhound 52% (n02092002-Scottish\_deerhound)



n02091244-ibizan\_hound 98% (n02091244-ibizan\_hound)



n02086079-Pekinese 90% (n02086079-Pekinese)



n02110806-basenji 61% (n02110806-basenji)



n02108915-French\_bulldog 55% (n02096585-Boston\_bull)



n02110627-affenpinscher 99% (n02110627-affenpinscher)



n02100236-German\_short-haired\_pointer 75% (n02100236-German\_short-haired\_pointe)



n02093647-Bedlington\_terrier 38% (n02093647-Bedlington\_terrier)



n02112706-Brabancon\_griffon 98% (n02112706-Brabancon\_griffon)



n02113978-Mexican\_hairless 95% (n02113978-Mexican\_hairless)





n02092002-Scottish\_deerhound 52% (n02092002-Scottish\_deerhound)



n02091244-ibizan\_hound 98% (n02091244-ibizan\_hound)



n02086079-Pekinese 90% (n02086079-Pekinese)



n02107908-Appenzeller 88% (n02107908-Appenzeller)



n02093647-Bedlington\_terrier 38% (n02093647-Bedlington\_terrier)



n02112706-Brabancon\_griffon 98% (n02112706-Brabancon\_griffon)



n02113978-Mexican\_hairless 95% (n02113978-Mexican\_hairless)



n02097130-giant\_schnauzer 71% (n02097130-giant\_schnauzer)



## Hyperparameter Tuning

| Model  | Accuracy                      |
|--|-------------------------------|
| Model 1: <ul style="list-style-type: none"><li>● ImageNet</li><li>● Global Average Layer</li><li>● Dropout .2</li><li>● Dense 120 Softmax</li></ul>  | Train: 0.7414<br>Test: 0.8350 |
| Model 2: <ul style="list-style-type: none"><li>● ImageNet</li><li>● Global Average Layer</li><li>● Dropout .2</li><li>● Dense 1000 RELU</li><li>● Dropout .2</li><li>● Dense 120 Softmax</li></ul>   | Train: 0.7735<br>Test: 0.8292 |
| Model 3: <ul style="list-style-type: none"><li>● ImageNet</li><li>● Global Average Layer</li><li>● Dropout .2</li><li>● Dense 1000 Sigmoid</li><li>● Dropout .2</li><li>● Dense 120 Softmax</li></ul>  | Train: 0.7359<br>Test: 0.8348 |
| Model 4: <ul style="list-style-type: none"><li>● ImageNet</li><li>● Global Average Layer</li><li>● Dropout .2</li><li>● Dense 1000 RELU</li><li>● Dropout .2</li><li>● Dense 500 RELU</li><li>● Dropout .2</li><li>● Dense 120 Softmax</li></ul>   | Train: 0.7397<br>Test: 0.8275 |
| Model 5 <ul style="list-style-type: none"><li>● ImageNet</li><li>● Global Average Layer</li><li>● Dropout .2</li><li>● Dense 1000 Sigmoid</li><li>● Dropout .2</li><li>● Dense 500 RELU</li><li>● Dropout .2</li><li>● Dense 120 Softmax</li></ul> | Train: .7059<br>Test: .8161   |

|  |                             |
|--|-----------------------------|
| Model 6 <ul style="list-style-type: none"><li>• ImageNet</li><li>• GlobalAverageLayer</li><li>• Dropout .2</li><li>• Dense 1000 Sigmoid</li><li>• Dropout .2</li><li>• Dense 1000 Sigmoid</li><li>• Dropout .2</li><li>• Dense 120 Softmax</li></ul> | Train: .6856<br>Test: .8088 |
|--|-----------------------------|