PyBark

A neural network model for classifying dog breed images

CWRU Bootcamp Final Project

Dena Wetmore
Steven Anthony
Kaleb Decker
Jack Ritz
Em Needles

The Dataset

9,600 dog images

120 dog breed categories

https://www.kaggle.com/datasets/miljan/stanford-dogs-dataset-traintest

Data Pre-processing

- Renamed dog image folders to remove hyphens
- Created a database for the dog breed images

Database

- SQLite database
- Flexible
- Simple
- Breed (*text)*, dog picture (*blob*)

Model Training

- Models were trained on a sample size of 9,600 images
- 2 different models trained
- First Model (Pre-trained) - 86% accuracy

```
: Found 12000 images belonging to 120 classes.
Found 8580 images belonging to 120 classes.
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/vgg16/vgg16_weights_tf_dim_ordering_tf_kernels_notop.h5
58889256/58889256 [==============================] - 1s 0us/step
Epoch 1/5
375/375 [==============================] - 1923s 5s/step - loss: 1.742 - accuracy: 0.8272 - val_loss: 0.6712 - val_accuracy: 0.7459
Epoch 2/5
375/375 [==============================] - 1896s 5s/step - loss: 1.7896 - accuracy: 0.8452 - val_loss: 0.7078 - val_accuracy: 0.7442
Epoch 3/5
375/375 [==============================] - 4621s 12s/step - loss: 1.7608 - accuracy: 0.8483 - val_loss: 0.6957 - val_accuracy: 0.8439
Epoch 4/5
375/375 [==============================] - 3630s 10s/step - loss: 1.822 - accuracy: 0.8806 - val_loss: 0.6671 - val_accuracy: 0.7360
Epoch 5/5
375/375 [==============================] - 16573s 44s/step - loss: 1.632 - accuracy: 0.9434 - val_loss: 0.6868 - val_accuracy: 0.7268
269/269 [==============================] - 822s 3s/step - loss: 4.6868 - accuracy: 0.8568
[4.686814308166504, 0.026806525886058807]
```

Model Deployment & Testing

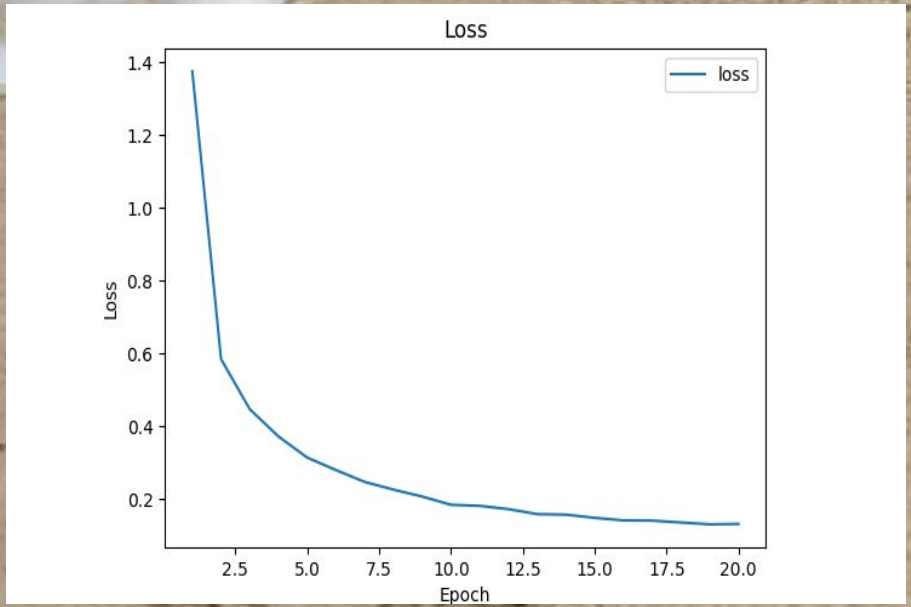
- 96.2 % accuracy model
- 1.35% loss

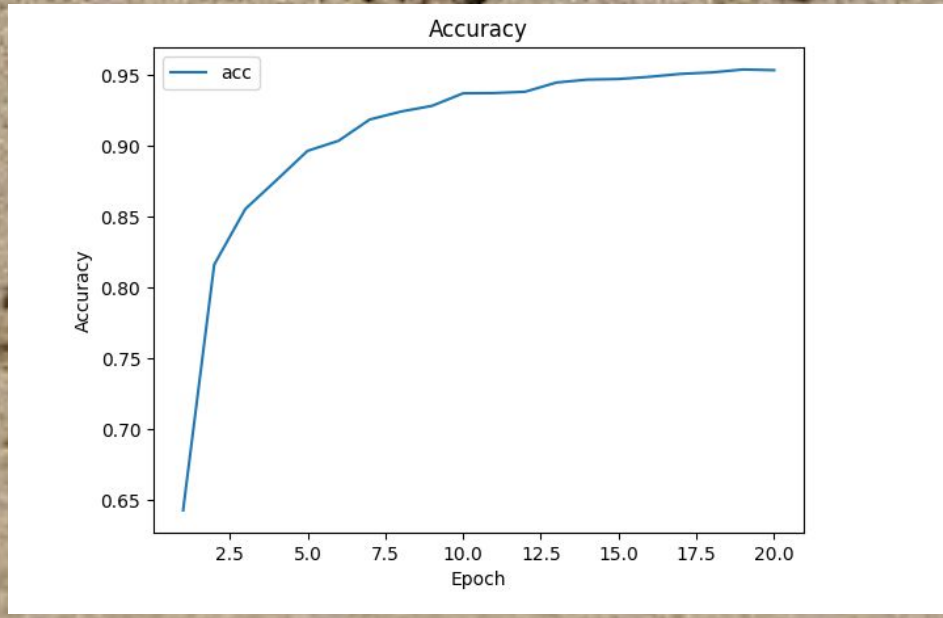
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 keras_layer (KerasLayer)    (None, 1280)              2257984

 dropout (Dropout)           (None, 1280)              0

 dense (Dense)               (None, 120)               153720

=================================================================
Total params: 2411704 (9.20 MB)
Trainable params: 153720 (600.47 KB)
Non-trainable params: 2257984 (8.61 MB)
_________________________________________________________________
```

```
Epoch 1/10
300/300 [==============================] - 470s 2s/step - loss: 1.6255 - acc: 0.6020 - val_loss: 0.6830 - val_acc: 0.8079
Epoch 2/10
300/300 [==============================] - 466s 2s/step - loss: 0.5804 - acc: 0.8283 - val_loss: 0.6000 - val_acc: 0.8146
Epoch 3/10
300/300 [==============================] - 466s 2s/step - loss: 0.4369 - acc: 0.8658 - val_loss: 0.5614 - val_acc: 0.8154
Epoch 4/10
300/300 [==============================] - 514s 2s/step - loss: 0.3437 - acc: 0.8938 - val_loss: 0.5503 - val_acc: 0.8229
Epoch 5/10
300/300 [==============================] - 504s 2s/step - loss: 0.2794 - acc: 0.9146 - val_loss: 0.5515 - val_acc: 0.8246
Epoch 6/10
300/300 [==============================] - 511s 2s/step - loss: 0.2334 - acc: 0.9317 - val_loss: 0.5502 - val_acc: 0.8279
Epoch 7/10
300/300 [==============================] - 523s 2s/step - loss: 0.2021 - acc: 0.9389 - val_loss: 0.5638 - val_acc: 0.8267
Epoch 8/10
300/300 [==============================] - 523s 2s/step - loss: 0.1809 - acc: 0.9467 - val_loss: 0.5620 - val_acc: 0.8333
Epoch 9/10
300/300 [==============================] - 472s 2s/step - loss: 0.1541 - acc: 0.9551 - val_loss: 0.5686 - val_acc: 0.8258
Epoch 10/10
300/300 [==============================] - 514s 2s/step - loss: 0.1358 - acc: 0.9623 - val_loss: 0.5641 - val_acc: 0.8296
```

Data Model Optimization

- Re-ran original model at 20 epochs instead of 10
- Model trained at 100% accuracy leading us to conclude that the model was over-trained
- Loss: 1.8%
- Accuracy: 100





Model predictions (green: correct, red: incorrect)

Conclusion and Final Thoughts


THE END