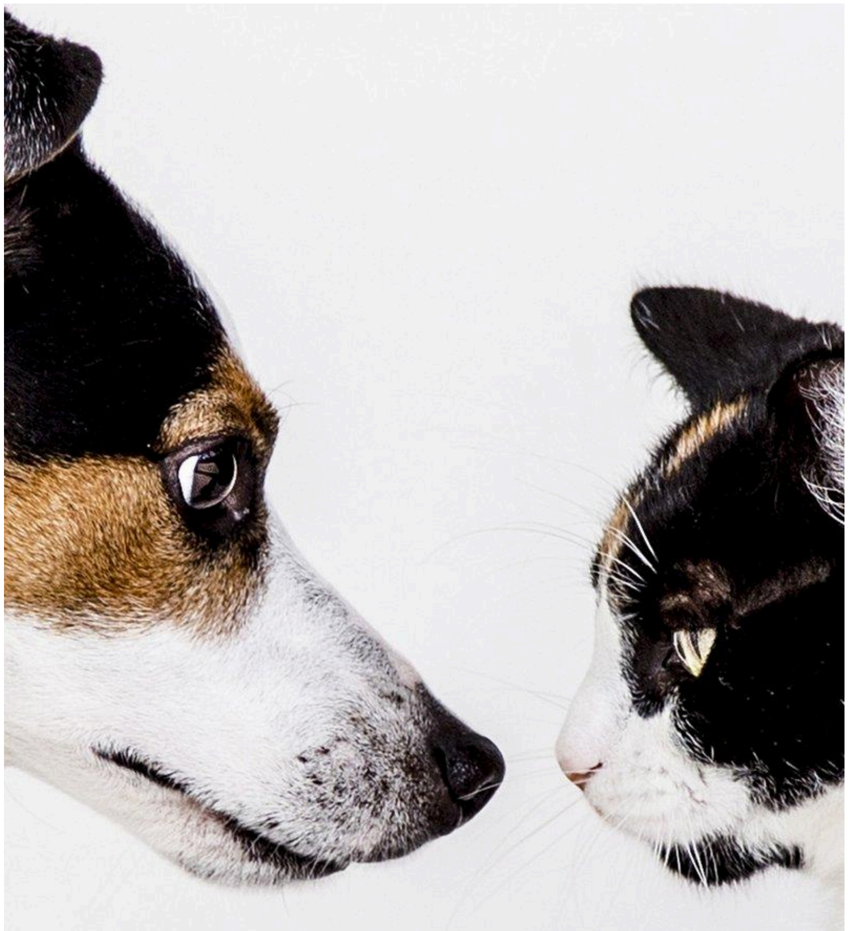
Cat vs. Dog Image Classification Model

Savannah Sky Katona



Introduction

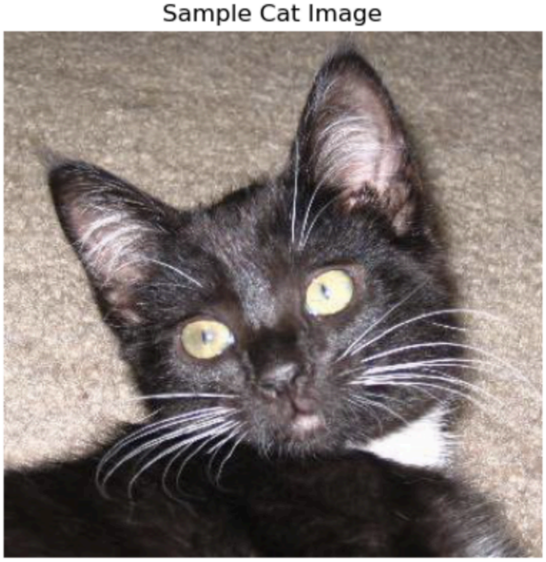
As humans, we have a knack for distinguishing between cats and dogs based on subtle physical characteristics. Cats are generally recognized by their small pointed ears, short snouts, and round or angular faces. While dogs typically have floppy ears, elongated snouts, and oval faces. Although not evident in every case, cats are generally smaller and more furry than dogs. While it may seem like a simple task to differentiate between these two creatures, it's not quite so easy for a computer to do so.

The goal is to develop a deep-learning model that can distinguish between images of cats and dogs with precision. This classification process will provide predictions based on images. I will construct a convolutional neural network utilizing Keras and Tensorflow. Once completed, the model can then be used in an application, enabling users to submit an image of their beloved pet and receive an accurate prediction.

The Method

Utilizing Tensrflow, the following steps were taken to classify cats and dogs:

Sample Cat Image



- Datasets consisting of 8,000 training images were loaded for cats and dogs.

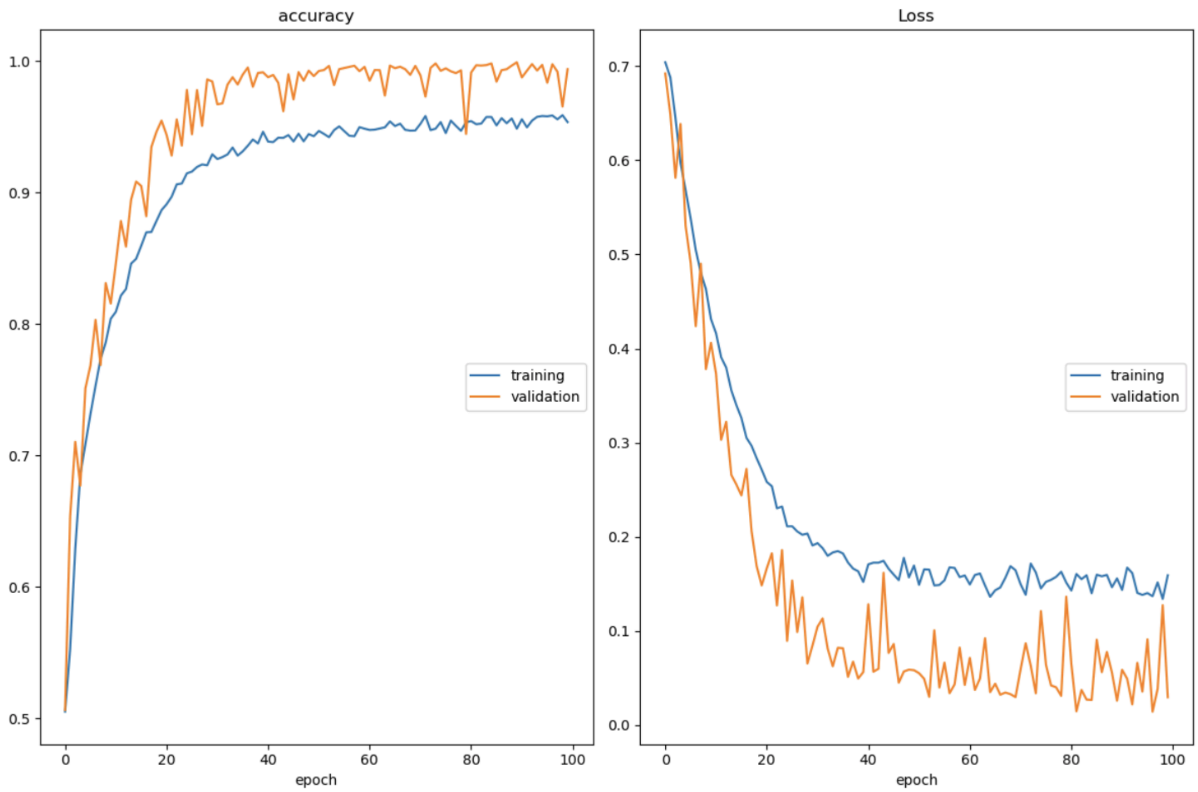
- I used a testing dataset consisting of 2,000 alternate images of cats and dogs, as well, to obtain a testing basis.

Sample Dog Image

- Each test image, was projected onto the training model for cats and dogs.

- If the cat-basis norm was greater than the dog-basis norm, the test image was classified as a cat; otherwise, it was classified as a dog.

- Success rates were computed. As you can see below, the accuracy rates of this model are high and consistent.




```
accuracy
        training                  (min:    0.505, max:    0.959, cur:    0.954)
        validation                (min:    0.506, max:    0.999, cur:    0.994)
Loss
        training                  (min:    0.134, max:    0.704, cur:    0.159)
        validation                (min:    0.014, max:    0.692, cur:    0.029)
250/250 [==============================] - 171s 683ms/step - loss: 0.1590 - accuracy: 0.9538 - val_loss: 0.02
92 - val_accuracy: 0.9940
```

CONCLUSION

In conclusion, the Image Classification model for Cat and Dog Images using CNN architecture has been successfully implemented. The Sequential model API has been utilized to create the model layer-by-layer, starting with a Conv2D layer, followed by batch normalization, activation, and max pooling layers. The model has been trained for a total of 100 epochs.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 124, 124, 64)      4864

 max_pooling2d (MaxPooling2  (None, 62, 62, 64)        0
 D)

 conv2d_1 (Conv2D)           (None, 58, 58, 32)        51232

 max_pooling2d_1 (MaxPoolin  (None, 29, 29, 32)        0
 g2D)

 dropout (Dropout)           (None, 29, 29, 32)        0

 conv2d_2 (Conv2D)           (None, 25, 25, 16)        12816

 max_pooling2d_2 (MaxPoolin  (None, 12, 12, 16)        0
 g2D)

 dropout_1 (Dropout)         (None, 12, 12, 16)        0

 flatten (Flatten)           (None, 2304)              0

 dense (Dense)               (None, 256)               590080

 dropout_2 (Dropout)         (None, 256)               0

 output (Dense)              (None, 1)                 257

=================================================================
Total params: 659249 (2.51 MB)
Trainable params: 659249 (2.51 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Same model ran with different parameters.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_3 (MaxPoolin  (None, 31, 31, 32)        0
 g2D)

 conv2d_4 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_4 (MaxPoolin  (None, 14, 14, 32)        0
 g2D)

 dropout_3 (Dropout)         (None, 14, 14, 32)        0

 conv2d_5 (Conv2D)           (None, 12, 12, 16)        4624

 max_pooling2d_5 (MaxPoolin  (None, 6, 6, 16)          0
 g2D)

 dropout_4 (Dropout)         (None, 6, 6, 16)          0

 flatten_1 (Flatten)         (None, 576)               0

 dense_1 (Dense)             (None, 128)               73856

 dropout_5 (Dropout)         (None, 128)               0

 output (Dense)              (None, 1)                 129

=================================================================
Total params: 88753 (346.69 KB)
Trainable params: 88753 (346.69 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

The final results show a training accuracy of around 95% and a test accuracy of around 99%, indicating that the model has achieved good performance on both the training and testing datasets. The total number of parameters in the model is 659,249; out of which all were trainable parameters and 0 were non-trainable parameters. Overall, the Image Classification model for Cat and Dog Images using CNN architecture has been successfully implemented with good performance on both the training and testing datasets.