Image Classifier: Dog Breed Classification

By Ari Francisco

Purpose of An Image Classifier

- Identify specific individuals within a picture
  - Social media companies use this to connect users
  - Security and law enforcement use this to find a person of interest

- Develop self-driving vehicles
  - classification for their vehicles to recognize pedestrians, traffic signals, signs, etc

- This image classifier will predict the breed of a dog in the image

The Dataset

- Used Kaggle’s Dog Breed Identification Dataset
- There are 10,222 images
- These images are grouped into 120 different dog breeds

Exploratory Data Analysis

- The histogram in Figure 1 shows the distribution of the breeds
  - There are about 70 to 120 images per dog breed
- Figure 2 shows that most of the images have less than 1 million pixels
  - The minimum number of pixels per image is 12,240 and the maximum is 7,990,272
  - The mean is 184,176.35 with a median of 183,875
- Figure 3 illustrates the distribution of the image sizes
  - Most of the images seem to be around 500x500 pixels

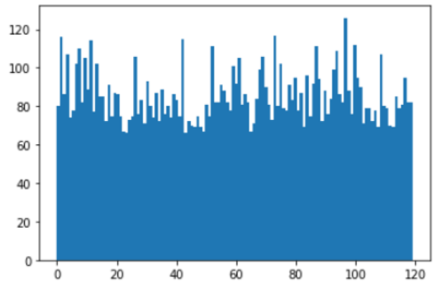

Fig 1. Histogram of Dog Breed Images

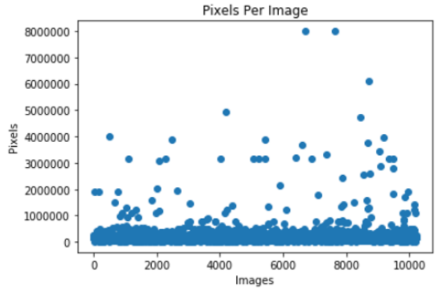

Fig 2. Pixels Per Image Scatterplot

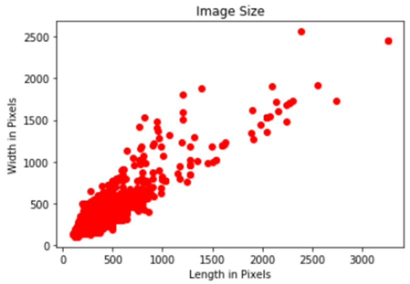

Fig 3. Image Size Distribution

Resizing the Images

After trial and error, a 200x200 pixel image maintained the integrity of the image without too much distortion of both the largest image in figure 4 and the smallest in figure 6

```
image shape: (2448, 3264)
number of pixels: 7990272
```

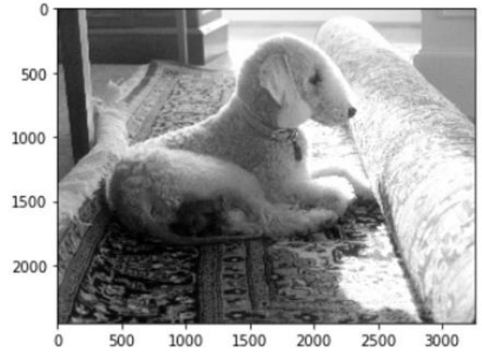

Fig 4. Largest Image at Original Size

```
image shape: (200, 200)
number of pixels: 40000
```

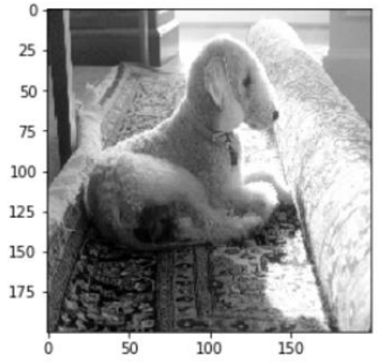

Fig 5. Largest Image Resized

```
image shape: (134, 97)
number of pixels: 12998
```

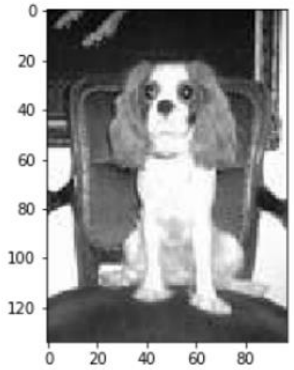

Fig 6. Small Image at Original Size

```
image shape: (200, 200)
number of pixels: 40000
```

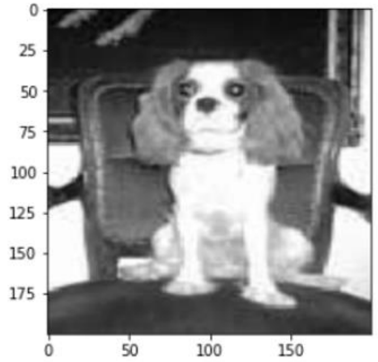

Fig 7. Small Image Resized

Hypothesis Testing

Breed	Average Pixels Per Breed
affenpinscher	126256.812500
afghan_hound	187398.060345
african_hunting_dog	174261.453488
airedale	177532.841121
american_staffordshire_terrier	165741.337838
appenzeller	175589.282051
australian_terrier	153140.058824
basenji	214494.936364
basset	219222.048780
beagle	189387.819048

- This table shows that there are differences in the average number of pixels per breed
- Need to determine whether the difference between the average number of pixels per breed is statistically significant
- The test used for this analysis is a One-Way ANOVA

Hypothesis Testing Cont.

	df	sum_sq	mean_sq	F	PR(>F)
breed	119.0	1.187827e+13	9.981743e+10	1.901349	1.525382e-08
Residual	10102.0	5.303371e+14	5.249822e+10	NaN	NaN

- Null-Hypothesis: There is no difference between the means of pixels of each breed
- Alt-Hypothesis: The means of some of the breeds are different
- Alpha= .05

- The p-value on our F-statistic is much smaller than the significance level of .05
- Therefore, the null hypothesis is rejected
- It is concluded that some of the differences between the pixel means of the breeds are statistically significant.

In-Depth Analysis

- The method used to classify the dog breed images is a convolutional neural network (CNN)
- Used Keras and Tensorflow in Python
- simplest model to use in Keras for a CNN is Sequential()
- The activation functions used were rectified linear unit (relu) on the first four layers and softmax used for the last layer
- The learning rate was set at .1 as this was the smallest learning rate possible with the computational power available
- Testing the different hyperparameters also resulted in Adam being the best optimizer

```
model = Sequential()
model.add(Conv2D(64, (3,3), input_shape = feature_norm.shape[1:]))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Conv2D(64, (3, 3)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))

model.add(Flatten())
model.add(Dense(120))
model.add(Activation('softmax'))

adam = optimizers.Adam(lr=0.1)
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

model.fit(feature_norm, label, batch_size=32, epochs=3, validation_split=0.2)
```

Results

```
Epoch 1/3
256/256 [==============================] - 375s 1s/step - loss: 284.7138 - accuracy: 0.0097 - val_loss: 280.5924 - val_accur
acy: 0.0073
Epoch 2/3
256/256 [==============================] - 371s 1s/step - loss: 284.7130 - accuracy: 0.0075 - val_loss: 280.5924 - val_accur
acy: 0.0117
Epoch 3/3
256/256 [==============================] - 371s 1s/step - loss: 284.7131 - accuracy: 0.0086 - val_loss: 280.5924 - val_accur
acy: 0.0103
```

- To measure the accuracy of the model, the validation accuracy is used
- The training set was split into 80% training data and 20% validation
- The accuracy used to assess the model is the validation accuracy
- The best validation accuracy of the three epochs was 1.17%

Conclusion & Improvements to The Image Classifier

- Clearly a validation accuracy of 1.17% is less accurate than randomly guessing
- Multiple hyperparameters were used, however, the best ones were not able to overcome the main limiting factors
  - Computing power
  - Low number of images per breed
- To improve on this model without having to invest in a more powerful computer transfer learning should be used
- There are a few transfer learning models available to try on the dog breed classification dataset, VGG-19 and ResNet-50, amongst others
- On a more positive note, many entries into the Kaggle competition for dog breed classification were not able to reach 1% accuracy without transfer learning.
- Once transfer learning was used, results can range from 60% accuracy and higher