Jason Dean
Methods for Data Analysis
Final Project
June 8th, 2017

Introduction

Humans are excellent at image classification. Show us a packed grocery store produce aisle and we can instantly pick out lettuce and bananas from hundreds of other options. Even though apples and pomegranates are both red, spherical, and contain a stem we can easily distinguish them visually. In fact, image classification is so simple for our brains that we take it for granted. Computers, on the other hand, have had more difficult time with image classification. A picture is simply a multi-dimensional array of numbers to a classification algorithm, and extracting features and information from millions (or billions) of pixels is a formidable challenge. Advances in machine learning and computing power are enabling computers to approach human image classification accuracies. These technologies will have a powerful impact on the medical imaging field, however here I turn to a much more important problem – dog breed classification. For this project I built logistic regression and convoluted neural network (CNN) models to classify dog breeds based on an image.

Image Processing

Over 20,000 images of dogs from 120 breeds is publicly available from the Stanford Vision website at the following location: http://vision.stanford.edu/aditya86/ImageNetDogs/. The goal of this project was to build a dog breed classifier based on an image, and the first decision I faced was what images and breeds to consider. Many dog breeds, such as the Siberian Husky and the Alaskan Malamute, look remarkably similar, and building a classifier to distinguish these edge cases was out of the scope of this project. Rather, I selected three dog breeds that were visually distinct: West Highland White Terrier, Irish Wolfhound, and the Black Tan Coonhound (Figure 1).

Figure 1. Representative images of dog breed images used for classification.

As shown in Figure 1, these three dog breeds have substantially different physiques and facial features. A significant fraction of the photos were found to contain humans, cats, and multiple dogs. Also, the images were different sizes (to be discussed below) and the dogs were not centered in the middle of the image.

There were 169 West Highland White Terrier,s 218 Irish Wolfhounds, and 159 Black Tan Coonhound color photos in this data set and the images were not all identically sized (Figure 2).

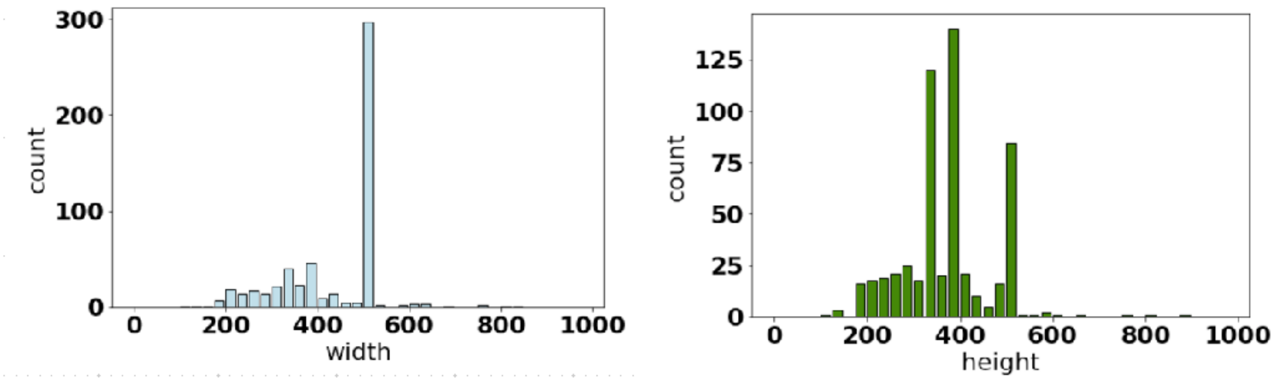

Figure 2. Pixel size distributions for all images.

Furthermore, we found that the mean image width was greater than the height for all of the dog breeds (Table 1).

	width	height
West Highland White Terrier	468.018	385.361
Irish Wolfhound	455.417	387.725
Black Tan Coonhound	351.126	322

Table 1. Mean pixel width and height for images used for dog breed classification.

For logistic regression I chose to scale images to a width of 450 and a height of 400 and images were converted to greyscale. For CNN images were scaled to a width of 96 and a height of 96, as it was found that CNN model fitting speed was significantly enhanced by decreasing image size and by resizing height and width to the same number of pixels, and color information was retained.

Machine learning classification algorithms make class predictions based on features, and for image classification problems features are represented as pixels. An image is essentially a matrix containing pixel information at each element. In order to convert images to a suitable format for machine learning each image is converted to a 1x(total pixel number) array, and this is further illustrated in Figure 3. The pixel feature matrix will have the number of images as columns and the number of pixel rows x the number of pixels in a column features.

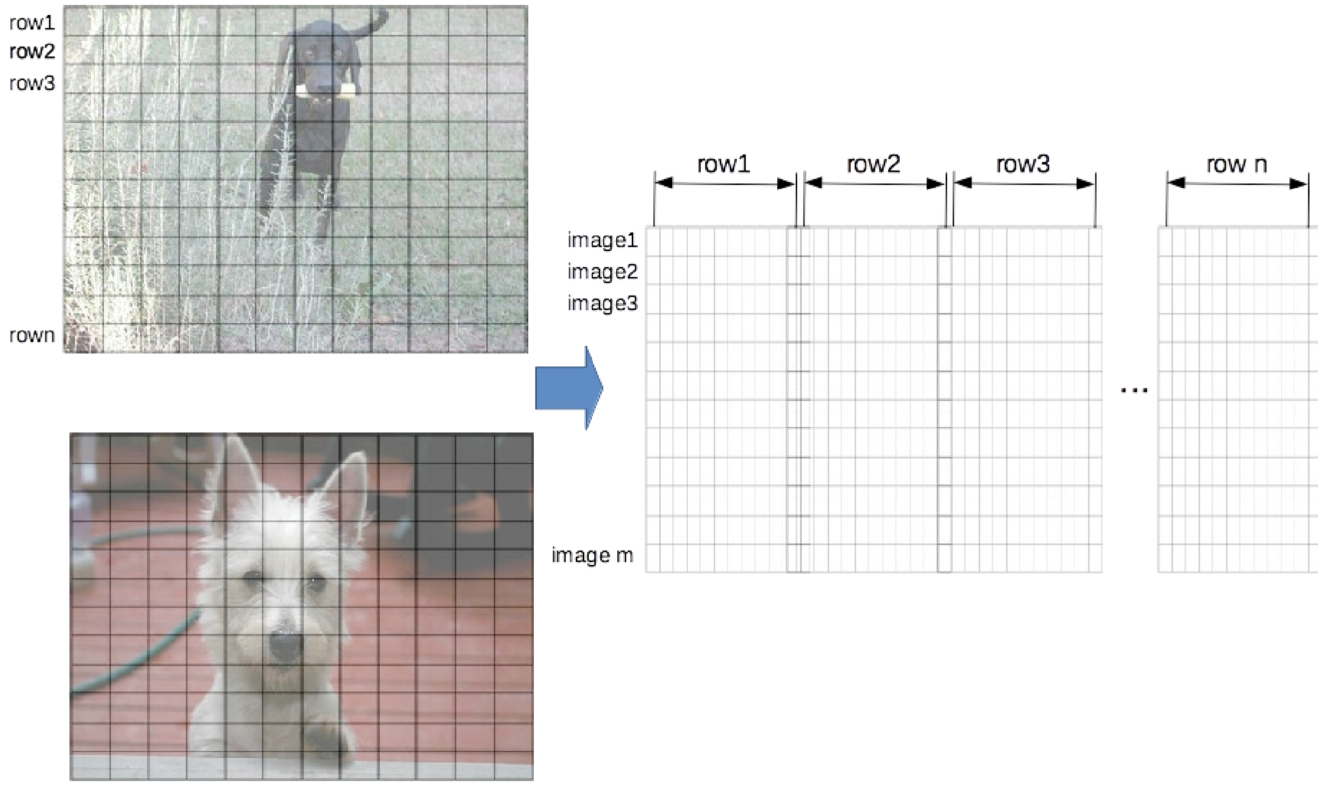

Figure 3. Strategy for decomposing an image into a feature matrix for classification. After resizing the images to the same size the pixel data was converted to a feature matrix consisting of images in rows and row pixel data in columns. The dimension of the pixel matrix will be : (image number)x(row number x number of pixels in a column)

Logistic Regression Model

There were in total 546 dogs in this data set and for logistic regression the images were scaled to a dimension of 450x400. This means that the pixel feature matrix contained 546 rows and 180,000 features, and it was found that hyperparameter tuning a logistic regression model with this number of features was prohibitively slow. To reduce the number of features but still preserve the variance in the data I choose to reduce the feature space dimensionality via Principal Component Analysis (PCA). PCA is a matrix decomposition method that determines the eigenvectors, or principal components (PC), of a covariance matrix, and by projecting our original data set on to this Principal Component space we can reduce dimensionality by creating a new smaller set of features that capture the majority of the variance in the data.

Before performing PCA on the pixel matrix I split the dataset into training (70%) and test (30%) for model fitting and validation. Next, I performed PCA separately on both the test and training set. The eigenvalue associated with each PC indicates how mucstill h variance is associated with that particular PC, and by plotting the cumulative proportion of each PC we can visualize how much of the pixel variance is captured as we include PCs. This plot is shown for the training data below in Figure 4.

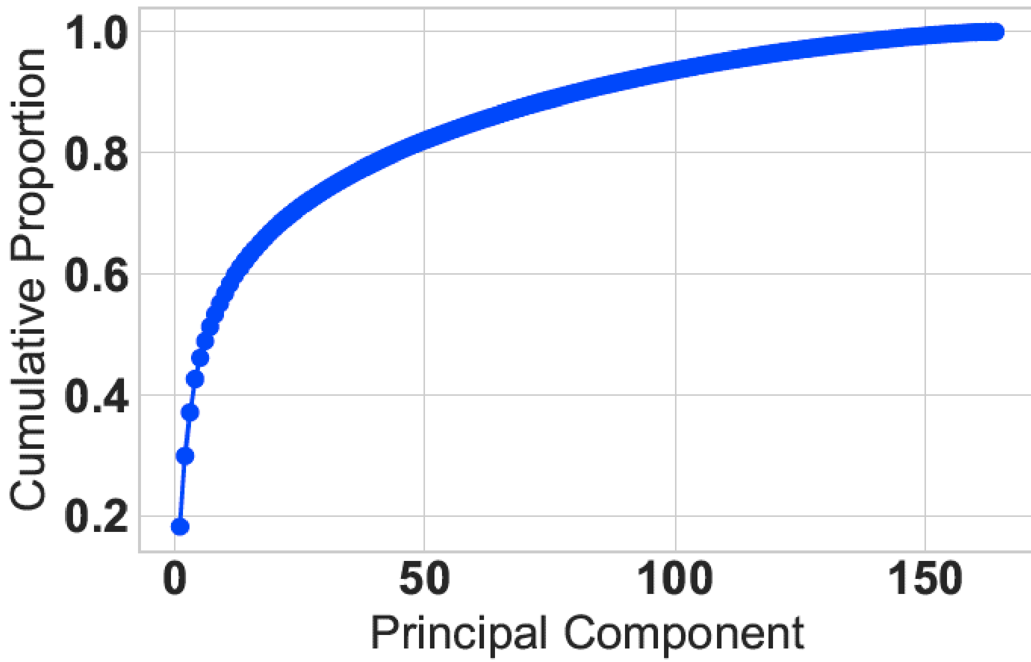

Figure 4. Cumulative proportion of each PC following PC decomposition of the test pixel feature matrix.

I found that the first PC captured 18.2% of the variance in the pixel feature data, and this is quite significant considering there are 180,000 pixels. Based on the results from Figure 4, I projected the original pixel feature data on to the first 400 PCs, and this new coordinate system captured >99% of the original variance while greatly speeding up model training.

Once the number of features was reduced and projected on to the PC space a multinomial logistic classifier was trained to predict dog breed based on pixel features. First, the model was hyperparameter tuned on the training data (70% of the data set) with 10-fold cross validation with GridSearch to determine the optimal 'C' parameter and the best solver. The C parameter controls the regularization strength, and the magnitude of C is inversely proportional to the strength of regularization. For this model I choose L2 regularization. L2 regularization prevents overfitting by adding an additional term to the loss function: the sum of the squares of the model coefficients. Thus, models with large coefficients will be heavily penalized. The optimal C value for this model was found to be 1e-9, which is quite small considering that the default model value is 1. This indicates that the optimal model identified by cross validation is relying on heavy regularization. Additionally, the best solver was found to be 'lbfgs'. A logistic regression model was next constructed and the test data was used to evaluate the predictive power. The accuracy was found to be 63.1% and the resulting confusion matrix is shown below in Table 2.

	Predicted West_Highland_white_terrier	Predicted Irish_wolfhound	Predicted black_tan_coonhound
Actual West_Highland_white_terrier	30	22	2
Actual Irish_wolfhound	9	37	12
Actual black_tan_coonhound	0	15	37

Table 2. LR model confusion matrix for predictions on test data.

The model performance was further quantified by calculating the precision and recall (Table 3).

	precision	recall
West_Highland_white_terrier	0.77	0.56
Irish_wolfhound	0.50	0.64
black_tan_coonhound	0.73	0.71
avg / total	0.66	0.63

Table 3. LR model performance metrics.

As shown in Table 2, the highest values in the confusion matrix are found on the diagonal, indicating that when all predictions for a given breed are grouped against the actual breed the highest value will be for the correct breed. This can be further quantified by analyzing the metrics in Table 3. Precision is defined as the following ratio: true positive / (true positive + false positive). In other words, when a breed prediction is made, how often is it correct? We found that the precision for the west highland white terrier was 0.77 and 0.73 for the black tan coonhound, indicating that when a model prediction is made for one of these two breeds it is right 77% and 73% of the time. The precision of the Irish wolf hound prediction, however, was 0.5, indicating that the prediction was no better than a guess. The recall is defined as the following ratio: true positive /(true positive + false negative). In other words, for a given breed, what fraction of the total observations of that breed did the model predict correctly? The highest recall, 0.71, was found for the black tan coonhound predictions and the lowest was for the west highland white terrier class. Taken together, the model had the best predictive power for classifying black tan coonhounds.

<u>Convoluted Neural Network</u>

As mentioned above, many of the images were not centered on the dog to be classified and the images were taken from different distances, and I speculated that a Convoluted Neural Network may be able to overcome these issues. I quickly learned that a deep understanding of CNNs was not feasible for me in the timeframe of this project, however I was curious to take my first (misguided?) steps into this technology. I opted to use keras, a high level neural networks API that uses the Tensor Flow framework, for CNN model construction because implementation of a CNN for image classification using keras is relatively straightforward after reading documentation (https://keras.io/).

The network architecture for this model was constructed following these excellent tutorials:

-http://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/

-https://elitedatascience.com/keras-tutorial-deep-learning-in-python

Once the model architecture was decided upon (the maximum amount of layers my MacBook Air could handle) the model was compiled using the ‘Adam’ optimizer and categorical cross-entropy loss. It was found that the ‘Adam’ optimizer resulted in faster training and better accuracy on the training data than the popular sgd optimizer. I was able to obtain 100% accuracy on the training data using 100 epochs, but I found that this model did not generalize well to the test data. Instead, I determined with trial and error that 40 epochs generated a model that fit the training data with > 90% accuracy and generalized reasonably well on the test data (to be discussed below). The log loss and accuracy over training epochs is shown below in Figure 4.

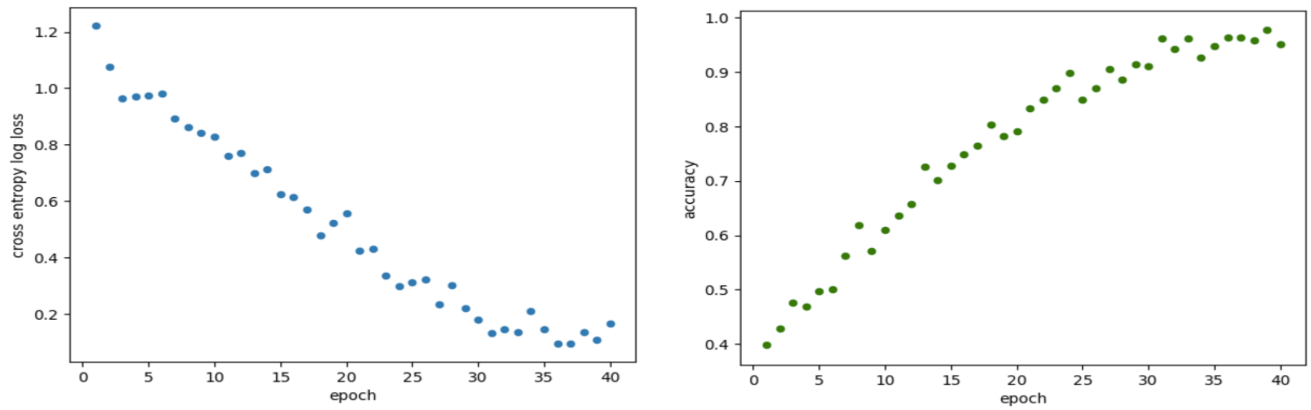

Figure 4. CNN cross entropy log loss and model accuracy on training data over 40 epochs.

The trained model was found to have 56% accuracy on the test data, 7.1% lower than the LR model, and the results are shown in Figure 5.

	Predicted West_Highland_white_terrier	Predicted Irish_wolfhound	Predicted black_tan_coonhound
Actual West_Highland_white_terrier	32	19	3
Actual Irish_wolfhound	10	31	17
Actual black_tan_coonhound	0	24	28

Similarly to the LR model, the Irish Wolf hound classification was problematic. Relative to other CNN projects in the literature the number of images in this data set is small. In an effort to improve model performance image augmentation, the process of generating slightly altered images from existing ones and thereby increasing the size of the training data was attempted, however I found that models that were generated in this fashion generalized poorly and provided accuracies no better than a random guess.

<u>Conclusions</u>

In conclusion, I found that a LR model was able to classify the three dog breeds evaluated in this data set with 63.1% accuracy and that a CNN achieved 56% accuracy despite achieving nearly perfect classification of the training data. The Irish Wolfhound proved to be problematic for both models, as this breed was associated with both the lowest precision and recall. It is possible that the CNN framework was not suitable for such a small data set, or that my limited experience with CNN hindered it's performance. Regardless, this project was really fun for me and I learned a huge amount doing it. Thanks!