Image processing techniques to create distinguishing features for bird classification

Ben Larson

May 6, 2015

1 Introduction

Bird classification is the process of identifying a bird species through analysis of its physical traits. Given an image of a bird where most features are visible we want to be able to predict what type of bird it is. This is an interesting and quite challenging problem. Birds have subtle features that differentiate them from each other: beak length, color of feathers, etc. Images of birds are also all from different angles therefore showing relevant features in different aspects for example, flying vs perched.

Attempts to find what important bird features has been done by Berg and Belhumeur. They created an experiment that would find similar birds and then use important distinguishing features to weight a feature vector that would be mapped in a similarity matrix. Features included things like beak shape, chest color, and other manually analyzed features for each bird [3]. A technique and game that incorporated human interaction and a initial computer vision weight vector was developed by Branson et al[4]. Increasing information gain per question from a human user, they refined important features for each bird species. Similar experiments have been done on this data set. Marini et al. used color histograms on the bird images after segmenting out the background and trained with support vector machines to achieve a range of 8% and 90% accuracy of classification [2].

In this experiment I used image processing techniques to extract features from the bird images based on histogram data. After extracting these features I used machine learning techniques to experiment if the features would separate bird examples. To test if the features were usable I used PCA and clustering methods on dictionary learned features that came with the data as well as my own created feature sets.

2 Methods

My goal was to use an image's histogram as a identifying feature array. I also was given a feature file for 10 different birds that were found using dictionary learning methods and used this as a comparison for my own created features (figure 1.c).

2.1 Data

The data, "Caltech-UCSD birds 200", is a collection of 200 birds available to the public. Each image includes a bounding box, a rough segmentation (mask), and binary attribute annotation (features). The features given had 1024 different features for each image [1]. It is important to note that the data is labeled. I know what each bird is because it is named in a file with the bird name.

2.2 Image Processing

Several, common image processing techniques were used to identify and analyze important parts of the image. As I was using histograms as features for the data my first step was to extract the actual bird pixels from the background pixels. This eliminates a lot of noise from the data set, but not all. The segmented masks for the birds came with the data set. Using indexing techniques the actual bird pixels were lifted from the images. The next step was using the gradient magnitude of the images. This is accomplished by taking the derivative of the image in the x and y directions then finding their magnitude. This is an image processing technique to help highlight detail in an image (pixels with high differences are indicative of detail/border) as well as make the data less distinguishable because of illumination differences(different lighting luminance, but same bird problem). Because of the nature of the data we wanted to pick up small details in the bird physical characteristics.

With the masked bird image the image histogram was found using 16 bins. 16 was chosen as a reasonable number of features that would be able to distinguish several different bird types. Now a distribution of pixel intensities that represent the bird image are ready to be processed by our machine learning techniques.

2.3 Machine Learning

First I would like to explain the feature set that I was given and tested my methods against. The feature set was derived from dictionary learning methods. Using dictionary learning techniques namely: finding a sparse linear model and decomposition of that model (see below), image representations can be approximated using a combination of pixels that were decomposed from the image. Essentially we are looking for small pixel patches (D_i) that are repetitive in a image. Using a sparse matrix(α_i) to map the pixels to respective locations in the image it would reconstruct our image. [5]

$$l(x,d) = min_{\alpha \in \Re^k} \frac{1}{2} \|\mathbf{x} - \mathbf{D}\alpha\|_2^2 + \lambda \|\alpha\|_1$$

Where x is the input image, D is the dictionary, α is the sparse matrix and λ is a hyper parameter. By minimizing differences between the model and the image they can find the best features D that would represent x, the image. The part of this model that is relevant to my project is the matrix D. This is a list of distinctive features of an image, and previous efforts have been made to use these as features for classification. This method of feature extraction may produce exact replicas of the image it was taken from, but how it will compare to similar objects, in different backgrounds, and with different scale and angles is to be analyzed.

For this project I used KMeans and PCA for analysis of the data. The machine learning software package sklearn (python) was used [6]. I will briefly explain KMeans and PCA as well as how they were used in my program.

Principal Component Analysis is a method for reducing the dimensions of a vector by preserving important information. This information is the minimization of the reconstruction error (class slides). So in my research I first used PCA to reduce my feature vectors (size 16, 48, and 1024) down to 2 dimensions. Which I then send to the clustering algorithm KMeans.

Kmeans is a clustering algorithm that finds groups of values that intuitively would belong together (at least in an image processing setting). Given a number of clusters it randomly assigns the cluster centers, or centroids, and then will iteratively assign points to a mean and then update the mean to better represent a grouping of the points. Formally it seeks to minimize:

$$\sum_{i}^{n} min_{\mu_j \in C} \left(\|x_j - \mu_i\|^2 \right)$$

where x is a point and μ is the mean of the points for each cluster/group of points for each iteration. For my algorithm I used this to cluster together the reduced feature data so that I could visualize the groups in a x,y plot using a verroni diagram.

As a sanity check that PCA could reduce a high dimensional feature set to 2 dimensions while still preserving seperability and also that KMeans would cluster data into several clusters given relevant features, I created a random set of 3 classes that were 3 different normally distributed clusters with different means. I wanted to be able to reduce the the input data from 3 dimensions (x,y,z) to 2 dimensions (x,y) and see the 3 different clusters in 2 dimensions. This simple example correctly reduced the dimensions while preserving the separability of the 3 classes. This is how my experiment was framed.

After the image processing I have a list of histograms for each image. The histogram returns an array of 16 entries for gray scale and 48 (3 for each channel) for color images. An entry is defined as the number of pixel intensities in a bin in the histogram. $\mathbf{x} = \{x_1, x_2, ...x_{16}\}$ and $\mathbf{x} = \{x_1, x_2, ...x_{1024}\}$ These entries are interpreted as the features for each bird example. I was given 1024 entries for the dictionary learning features. I now reduced the dimensionality using PCA so I could get the dimensions/features into 2 dimensions which makes the data easy to visualize.

With the dataset reduced to 2 dimensions I used the KMeans clustering algorithm. First the data was normalized to be in the range (0,1), this helps with plotting and the verroni diagrams. The KMeans algorithm was given a cluster size(3 or 10), random initialization of the means, and ran for 300 iterations. I didn't notice any difference in running more iterations than this in convergence (for toy example and for my own data). I didn't have any nonlinear shape of data I was trying to separate that would require KMeans to start at an assigned centroid so the random method was assumed sufficient. The KMeans could be fit to a dataset and then used to predict given an input of features. I utilized this to train a KMeans learner that could predict and help me test my feature set after splitting the data into training and testing data.

Plots were created that would allow us to see how our features were being clustered (see figures). The data in the plots are the location of the normalized PCA results in 2 dimensions (x,y) with a color that represents what cluster/class they belong to. We then plotted an image corresponding to a cluster group so that we could visually inspect if the reduced features were separating the data into clusters. I then used the kmeans prediction that was fit to our data to predict a color/class/cluster for a meshgrid(0:1, 0:1, for 10,000 entries) of values[6]. This visually pleasant result shows us the decision boundaries as a verroni diagram.

The KMeans prediction needed some sort of testing, more than me just visually inspecting if the images on the plots were the same bird. The data is labeled. The KMeans function allows you to first fit a function on a dataset, then you can predict using this. I split my data 50/50 to train and to test on. This split allowed me to test the accuracy of the KMeans prediction.

Several runs of the algorithm were done. Tests were done using 3 and 10 different bird classes. The same birds for the set of 3 and 10 were used. I tested the gray scale, color, and dictionary features for each set of 3 and 10 birds. See table and figures below.

3 Results

I had two metrics for evaluation of my methods. The first was a simple plot with images attached to cluster points. This is a helpful tool for image learning because it allows you to see images that were clustered together and evaluate what types of features might be useful, or what might be learned. The second method was a comparison of the cluster to the label, or bird. For each bird example I checked how many times there was a cluster that was different from the most common cluster in all examples of that specific bird. We would expect a value of correct per total examples = 1 for a perfect cluster.

Our results (see plot, and table) indicate that our methods did not do well in creating features for the birds. The classifiers seem to do better with just 3 classes, however our features do not distinguish well when we look at 10 classes. What I wanted was features that would be different enough from bird to bird that I would be able to distinguish each bird into a cluster. We do better than chance in all cases (over 30% for 3 classes, 10% for 10 classes), but not well enough that I would tell the community that I've found a perfect method for classifying birds.

4 Discussion

In this project I attempted to create a distinguishing feature set for birds based on their gray scale and color histograms and compare this feature set to a dictionary learning feature set. The resulting feature set was reduced using PCA methods and clustered using KMeans. I then plotted the images and measured the accuracy of the predicted cluster.

Features	Figure	Number Images	Accuracy (as percent correct)
Color Histogram 3 Classes	Fig 2.a	81	44
Dictionary Learning 3 Classes	Fig 2.b	64	51
Gray Scale Histogram 3 Classes	Fig 2.c	81	56
Dictionary Learning 10 Classes	Fig 2.d	215	22
Gray Scale Histogram 10 Classes	Fig 2.e	288	24
Color Histogram 10 Classes, Image gradient not used	Fig 2.f	288	14
Color Histogram 10 Classes	Fig 2.g	288	16

The reason I split my data 50/50 was because I was testing features. With the lack of similarities between the images as well as lack of outliers of the bird features I figured this testing and training ratio would keep me from over-fitting and under-fitting my KMeans learner. Increasing my training data would change the decision boundaries by an amount that wouldn't be of interest(because I'm not sure which of the features are relevant). And increasing my test set would make the variance in my data increase.

One of the first observations one will notice when looking at the results is the decrease in accuracy when including more classes to the training and testing. I speculate the reason for this is the features that I'm getting from the images start to have more in common with other birds as I increase the variety of birds in the classification. The more classes I include in a test increases the number of images analyzed significantly and decreases the chance that an image's histogram will create a unique detail for a bird. For example, compare the image 1.e and you see the clusters are aggregated closer to one another than the same gray scale histogram of features in 1.c.

An interesting part of this project is the form that the clusters take. Especially relevant are the plots 1.a color histogram and 1.c gray histogram. We see there is a circular shape the color histogram, most likely following the order of colors, and the gray scale is sort of bunched together in the center. Something we can take away from this is that the grayscale is looking more at the detail differences in the birds while the color histogram is taking the detail and the color into consideration. It is then surprising that it doesn't produce results that are more accurate. My guess is that there are irrelevant features or colors that are not as good at classifying a bird. My conclusion is that distance in clustering (separability) in this case doesn't imply accuracy of classification.

Looking at the actual features we may need to change the histograms in future work. The histogram bins are one of the few parameters that we might be interested in changing. The histograms were chosen as a reasonable set of features that would show a distinguishing set of pixels for each bird. To truly find the best number of bins a different numbers bins should be tested and compared without changing this data set so as to maximize the accuracy. I would doubt any fewer bins would increase accuracy, but increasing the bins might bring subtle enough changes so that the PCA method will have more features to consider

In this project we see that the bird dataset is a difficult machine learning problem in that it is hard to extract features from the images. We saw that dictionary learning, while being a great tool for reconstruction of images, struggles to provide a vector of features that can be applied across all the images. We also saw that by just taking the histograms of the images we get results that are similar to advanced methods such as dictionary learning.

References

[1] http://www.vision.caltech.edu/visipedia/CUB-200.html

[2] Marini Adreia, Bird Species Classification Based on Color Features. et al. PUCPR Brazil.

[3] Thomas Berg and Peter N. Belhumeur. How Do You Tell a Blackbird from a Crow?. IEEE International Conference on Computer Vision. 2013. Columbia University.

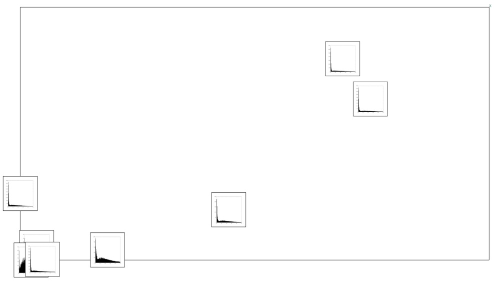
[4] S Branson et al. Visual Recognition with Humans in the Loop. University of California, San Diego.

[5] Mairal, Back. *Online Dictionary Learning for Sparse Coding.* ICML '09 Proceedings of the 26th Annual International Conference on Machine Learning. Pages 689-696.

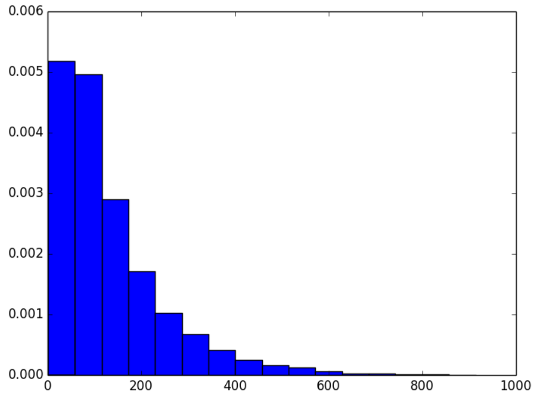
[6] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

(a) Example, Bird Image

(b) Segmented image(mask)

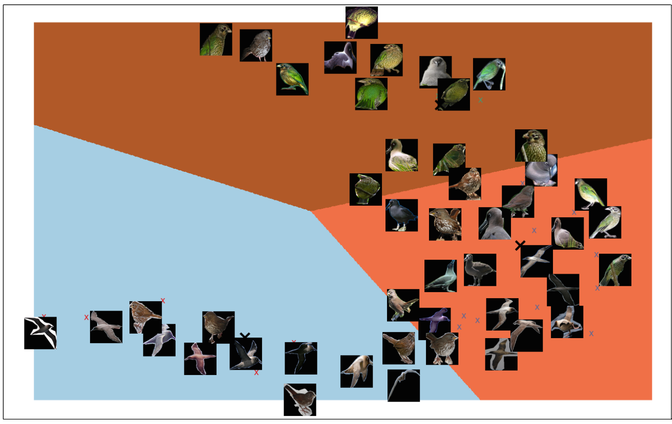


(c) Early attempt using KMeans clustering with histogram as features. Images histograms plotted to aid seeing similarities. This is of just 1 bird class.

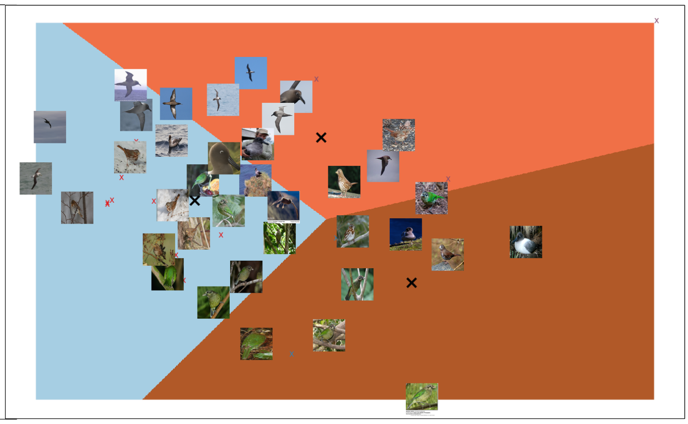


(d) Example histogram of bird image

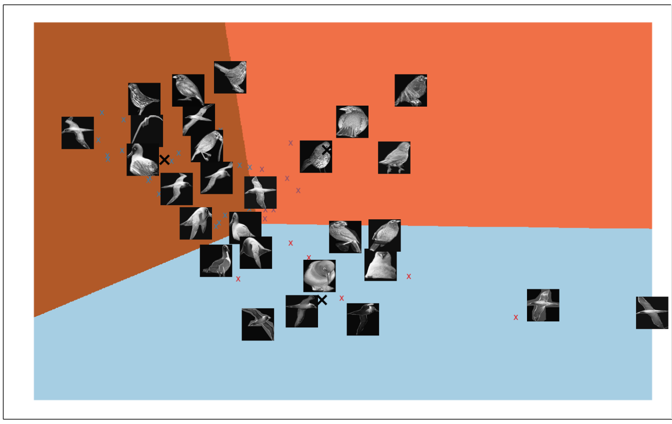
Figure 1: Image results.



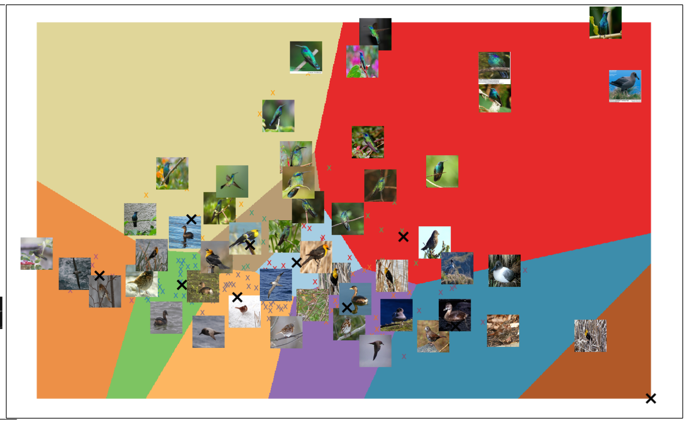
(a) 3 bird classes. Features from color histogram.



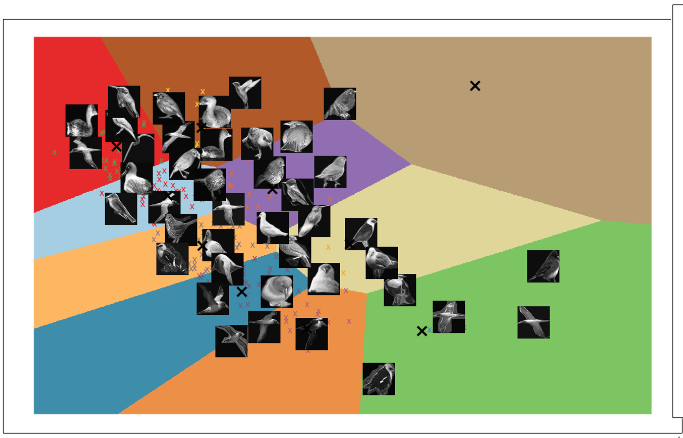
(b) 3 bird classes. Features from dictionary learning.



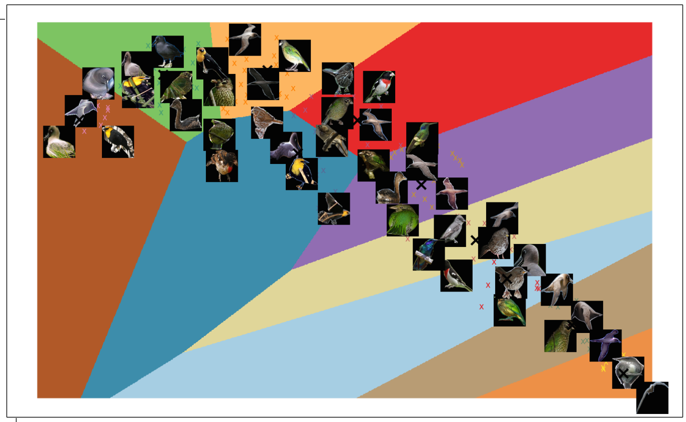
(c) 3 bird classes. Features from gray scale histogram.



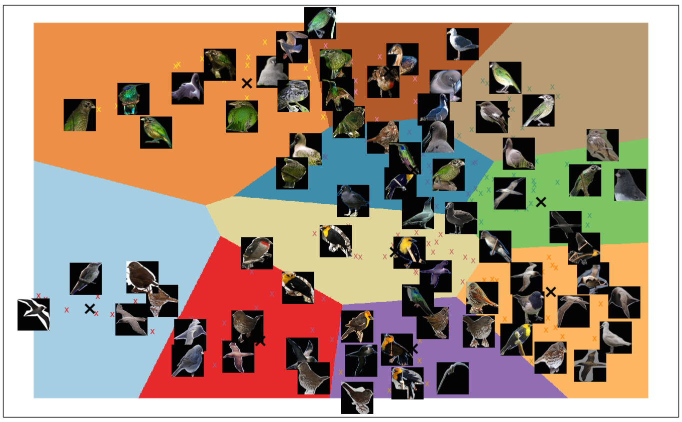
(d) 10 bird classes. Features from dictionary learning.



(e) 10 bird classes. Features from color histogram



(f) 10 bird classes. Features from color histogram NOT using image gradient.



(g) 10 bird classes. Features from color histogram

Figure 2: Veronni diagrams showing kmeans prediction decision boundaries. Note the features from dictionary learning do not have the images masked. Each cluster is represented as a separate color(actual bird class is a different color from it's background class. This keeps the marks visible). Centroids for the cluster is a black X.