CIS442D Advanced Business Analytics

Analytics project

Image Classification of Dogs and Cats

by

Qingyuan Zhang

Lu Xia

Jiaqi Xiong

Delilah Huynh

Image Classification of Dogs and Cats

**Introduction**

In this project, our objective is to develop models to classify images of dogs and cats. We investigated several classification algorithms to train classifiers that accurately classify new unknown images as either “dogs” or “cats”, and then examined their performance.

The data comes from the world’s largest site that finds homes for homeless pets, and the images are now classified manually by people. By developing classification models, we hope to distinguish the images automatically by computer, so that the huge amount of images of dogs and cats can be classified more efficiently. This will speed up the process for distributing the images of dogs and cats to individuals who are willing to adopt dogs and cats, respectively. Precise advertising may increase the probability that the homeless pets successfully find their new owners.

**Challenges**

There are many challenges that's supposed to be easy for people to solve, but difficult for computers. Such a challenge is often called a CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) or HIP (Human Interactive Proof). Our team’s final project, which is a HIP, is called Asirra (Animal Species Image Recognition for Restricting Access). We build a model to classify whether images contain either a dog or a cat. This task is difficult for computers, but studies have shown that people can accomplish it quickly and accurately.

While random guessing is the easiest form of attack, various forms of image recognition can allow an attacker to make guesses that are better than random. There is enormous diversity in the photo database (different backgrounds, angles, poses, lighting, etc.), making accurate automatic classification difficult. In an informal poll conducted many years ago, computer vision experts posited that a classifier with better than 60% accuracy would be difficult without a major advance in the state of the art. For reference, a 60% classifier improves the guessing probability of a 12-image HIP from 1/4096 to 1/459.

**Data Source**

Our data source comes from Petfinder.com, the world's largest site devoted to finding homes for homeless pets. They've provided Microsoft Research with over three million images of cats and dogs, manually classified by people at thousands of animal shelters across the United States. Our dataset contains 25000 photos of dogs and cats. It is equally divided with same amount of dog photos and cat photos. All the data are jpg files with different size and file name indicates whether the photo is a cat or a dog. Some images have human and some images contains more than one cat. All these uncertain factors make the classification process much harder.

**Methods**

Our classifier takes images as input and outputs what the image contains. In other words, the output is a class label cat or dog. We train the algorithm to learn the differences between cat and dog. To make sure the image has been successfully converted to data, we choose to use Opencv library to load the photo and pre-process the image. OpenCV (Open Source Computer Vision) is a [library of programming functions](https://en.wikipedia.org/wiki/Library_(computing)) mainly aimed at real-time [computer vision](https://en.wikipedia.org/wiki/Computer_vision). It contains many useful functions to handle and process the images.

After we store image data into an array, which consists of color values in each pixel, we pre-process the data to standardize all the images in the dataset. As part of pre-processing, an input image or patch of an image is also cropped and resized to a fixed size. In addition, we simplify our calculation by converting all the photos to black and white. These two steps are very essential because the next step, feature extraction, is performed on standardized images.

In most cases, the input image has too much extra information that is not necessary for classification. Therefore, an essential step in image classification is to simplify the image by extracting the important information contained in the image and leaving out the rest. We compute a PCA (eigenfaces) on the face dataset with unsupervised feature extraction and dimensionality reduction. For all the eigenfaces we get from image, we only pick the top 30 faces as the extracted feature and project the input data on the eigenfaces orthonormal basis. We use the feature vector as independent variable and class label as dependent variable to run our prediction model. Like analyzing table data, we try different classification algorithm to see which one gives us the best result.

The classification algorithm takes feature vector as input and outputs a class label we need to train by showing thousands of examples of cats and dogs. Different learning algorithms learn differently, but the general principle is that learning algorithms treat feature vectors as points in higher dimensional space, and try to plane way to that partition the higher dimensional space in such a way that all examples belonging to the same class are on one side of the plane.

We investigated the following six classification algorithms and compare their performance with different parameters :

**1. KNN:** The first method used by our team is the K-Nearest Neighbor algorithm, also known as the knn() function. Consider the sequence (X1, Y1),…(Xi,Yj) where X’s represents a set of pixel predictor variables specifics to different facial features of either a cat or a dog, and Y is a vector consisting of 0’s or 1’s values corresponding to cat image or dog image, respectively. To recognize a face, a similarity measure based on the texture, color intensity, and location of each facial feature is used to identify the identity of the face. The algorithm will classify each element of X according to a majority vote of the k nearest points in the data set, using the training data.

**2. SVM:** The second classification that we considered was support vector machines (SVM). The objective of SVM is to separate different facial features of cats or dogs according to boundary conditions with a margin around the boundary and then classify regions in the feature space according to the regions formed by the boundary. Since we are working with images, SVM uses the pixels of each image to classify the conditions which will help form boundaries for classifications. This methods allows a certain amount of violations in the margin for misclassification, which is controlled by a cost parameter C to allocate a certain amount of violations in the margin.

**3. Decision Tree:** The main idea of the tree is to split the feature space of the pixels to regions with facial varieties, then assign a value to each region based on the texture, color intensity, and location of each pixel. The fitted image will then be determined as a dog image or a cat image. We first used the decision tree with a max depth of 3 to classify the photos. Then use the cross validation method to test the accuracy of the model. We also tried a decision tree with different max depth from 1 to 8, then cross validate the accuracy. The accuracy did not change much with different depths.

**4. Random Forest:** The next classifier we choose to use is the random forest. Random forest is an algorithm that fits a number of decision tree classifiers on various sub-samples of the dataset. For each dataset, random forest randomly selects features and branches if needed. We used random forest classifier to predict the image, then use cross validation to get the accuracy of the model.

**5. Boosting:** We also tried to boost with decision trees. Boosting produces a series of classifiers, training weak learners to strong ones. The training set used for each member of the series is chosen based on the performance of previous classifiers in the series, so that new classifiers can have better performance. We boosted decision trees using the AdaBoost algorithm. AdaBoost is short for Adaptive Boosting, which can be used in conjunction with many other types of learning algorithms to improve their performance.

**6. K-mean Clustering:** The last algorithm we used is K-means clustering, which divides data into k clusters based on similarities and find cluster centers in a set of unlabeled data. To segment the images of dogs and cats, the algorithm assigns a label of either 0 or 1 to every image so that pixels in the image with the same label share similar visual characteristics, such as location, color, intensity, and texture.

**Results**

|  |  |
| --- | --- |
| **Classifier** | **Accuracy & Variance** |
| KNN | 0.59 ± 0.01 |
| SVM (linear) | 0.58 ± 0.00 |
| SVM (RBF) | 0.65 ± 0.02 |
| Decision Tree | 0.58 ± 0.01 |
| Random Forest | 0.59 ± 0.01 |
| Boosting | 0.60 ± 0.01 |
| K-mean Clustering | 0.51 ± 0.00 |

From the above table, the standard error for all the models are below 2%, so we use the mean accuracy as the point estimator. SVM using RBF classifier performed the best with mean accuracy of 66%. However, SVM using linear classifier resulted in an accuracy of 58%, which is lower than SVM (RBF). This is the case because SVM (linear) only linearly separates the image features into two different regions, reducing the number of facial varieties for the predicted images. As for SVM (RBF), it could classify images based on small differences within the image dataset, increasing facial varieties to better distinguish between images of cats and dogs. Furthermore, KNN resulted in an accuracy of 59%. The decision tree model did not show a high accuracy of 58%, even though we tried different depth of the tree. Because the image has too many pixels and the tree needs many splits to capture features of pixels. The random forest model gave a relatively low mean accuracy of 59%. We think this is because random forest captured features randomly, and it is irrelevant to pick a feature twice. The accuracy of AdaBoost method is slightly higher than that of the simple decision tree method and random forest method, as Boosting improves the performance. The K-mean Clustering gave the lowest accuracy among six classification algorithms. We suppose the reason is that clustering is unsupervised rather than supervised model, and it may not be suitable for this analysis.

The highest accuracy we achieve is **66%** using the SVM (RBF) model, which we choose as the best classifier for our project. The 66% accuracy, which is significantly more than 50%, indicates that the model we developed is much better than a random guess.

**Conclusion**

Our team had essentially introduced the issue of image classification, illustrated its challenges, and briefly discussed the data background of our project. We then proceeded with the methodologies and results of the analysis. First, we used Opencv to load the image and computed a PCA (eigenfaces) on the face dataset with unsupervised feature extraction and dimensionality reduction. We then utilized six different predictive models and used 5-fold cross validation. SVM (RBF) classifier returned the best accuracy of 66%, where the K-Mean Clustering resulted in the lowest accuracy of 51%. According to the accuracy result, we chose SVM (RBF) as our best classification model for images of cats and dogs. For further analysis, our team could possibly increase the size of out of sample test dataset and use other predictive models to improve performance.

**Further Application**

The image classification for cats and dogs can be developed into more complex algorithm that is able to identify the specific faces for cats and dogs. This face recognition for pets can be applied to finding the lost pets with public CCTVs.