Building a Linear Classifier to Identify Cats and Dogs

Kyle Doerr
Department of Electrical and Computer Engineering
The University of Western Ontario
kdoerr@uwo.ca

Abstract: This paper discusses the results of building a classifier that can distinguish between cats and dogs using the bag-of-words model. The effects of normalizing feature vectors on classification accuracy show that data scaling is superior over L^1 and L^2 normalization for this particular problem. The findings also show that image cropping an object from its background fails to improve the accuracy of the bag of words model. It was also found that more distinguishable features can be found in the face of the animal when there is a large variance of phenotypes of animals belonging to the same family. The final results were that using the generic bag-of-words model on a dataset containing 21 breeds of cats and dogs with 4198 images the classifier was able to obtain an accuracy of 83.7%.

1 Introduction

Object recognition is an important task in computer vision and has a wide variety of real world applications. Object recognition in computer vision allows for the classification of different objects by grouping those that share similar traits. The goal of this work is to create a classifier that can determine in any standard image containing a cat or dog which pet family the animal belongs to. The classifier should be able to determine the family of cat or dog regardless of the animal breed, location of the animal, the pose, and the background behind the animal. This can be a challenging problem because of the wide variety of breeds of these animals and the fact that animals in particular cats are very deformable [1]. That is, there are many different poses that the animal can be in see Figure 1.

Another challenge in this task is that different breeds of the same family can have different fur colours and textures. The observable physical features that the animal shows are known as phenotypes which can make classification more challenging.

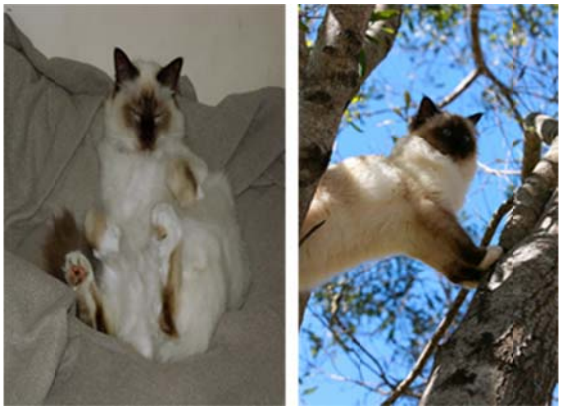

Figure 1: Different poses of the Birman cat [1]

Previous work on this task was conducted by Parkhi et al. [1] in which they created an annotated dataset of pets containing 12 different breeds of cats and 25 breeds of dogs with roughly 200 images per breed for a total of 7349 images. This dataset was compiled into what is known as the *Oxford-IIIT Pet Dataset* this dataset is managed by the Visual Geometry Group at the University of Oxford. The dataset was created by gathering images from various social networking sites in which each image was analyzed to ensure correctness of pet breed. They ended up building a classifier using shape and appearance models and were able to correctly classify the pet family with an accuracy rate of 95.37% and they were also able to distinguish among pet breed with an accuracy of 59%. It should be noted that the 59% accuracy they obtained is impressive given the difficulty of the problem.

This work uses a more simplified approach at tackling the problem of distinguishing pet family by using a generic bag-of-words model. The model uses SIFT descriptors to compute visual words that can be used on new images to identify similar features; see section 2 Methods for more information.

2 Methods

The classifier was built in MATLAB with similar ideas and concepts presented by Vedaldi and Zisserman in their guide *Practical Image Classification* [2]. The *VLFeat* open source library which contains many popular computer vision algorithms was used to generate the features [3]. The code for using a support vector machine was created by Chang and Lin and is called *LIBSVM* [4]. Due to some of the limitations on the functionality of *LIBSVM* by itself, *Weka LibSVM (WLSVM)* [5] was used. This is simply the *LIBSVM* library integrated into the popular tool Weka which is used for data mining and machine learning.

The first test in conducted in 3.1 involves finding out which normalization method performs the best on a small subset of the dataset. The first method used is the L^1 norm also known as the Manhattan distance. The second was to use the L^2 norm which is commonly known as the Euclidean norm [6]. The third method while not technically a norm involved scaling all vectors between 0-1 inclusive.

The second test conducted in 3.2 involves taking one cat breed and one dog breed and comparing the accuracy of the classifier by building the visual word vocabulary and analyzing the results when:

i) The original image is intact
ii) The animal is manually cropped
iii) The head of the animal is cropped

The third test in 3.3 involves taking 21 animal breeds with roughly 200 images per breed and determining the accuracy of the classifier on non-cropped images. See Table 1 for pet breeds used in the dataset.

2.1 Preprocessing

The images were first split up into cat and dog groups. From there the data was partitioned into two distinct datasets one for building the vocabulary and one for computing the histograms. Each image from the dataset used was read in one by one and was resized to have at most 480 pixels in height. Each image is then converted to grayscale.

Cat	Dog
Abyssinian	American Bulldog
Bengal	American Pit Bull Terrier
Birman	Basset Houng
Bombay	Beagle
British Shorthair	Boxer
Egyptian Mau	Chihuahua
Maine Coon	English Cocker Spaniel
Persian	English Setter
Ragdoll	German Shorthaired
Russian Blue	Great Pyrenees
Siamese	

Table 1: List of dog and cat breeds used

2.2 Visual Word Vocabulary

The first step in the training process was to compute a visual word vocabulary. Taking a set of training images, features were detected using a dense SIFT descriptor. The Descriptors were then extracted using a 128 x k matrix with key points that contained information regarding the location, scale, and contrast of the feature. Each descriptor was then quantized using the k-means algorithm to form a ‘word’ that would then build up to form a ‘vocabulary’. The number of words created was 1000 and each feature contained 10 times as many words see figure 2.

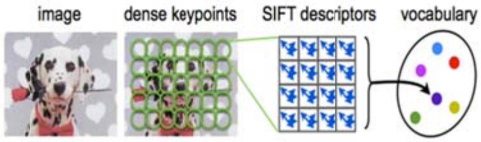

Figure 2: Process for creating a vocabulary [2]

Each feature is extracting by smoothing the image with a Gaussian kernel at scales of sizes 4, 6, 8, and 10 and then the sift descriptors are taken with a step of 6 pixels. The vocabulary was then stored into a k-dimensional tree structure to index the descriptors for faster processing. The images used in this process are then discarded as not to be used in the next stage.

2.3 Creating a Histogram of Visual Words

The next stage was to create a histogram of visual words on a set of unseen images. First, each image in the set has its features extracted using the same process as in 2.2. From there the descriptors are quantized in which the nearest visual word and the distance to the word in the previously computed vocabulary was determined, see figure 3.

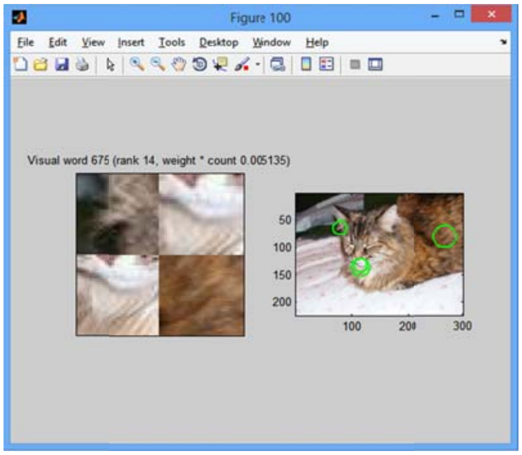
Figure 100
File Edit View Insert Tools Desktop Window Help
Visual word 675 (rank 14, weight * count 0.005135)
50
100
150
200
100
200
300

Figure 3: Most relevant visual words that appear in the vocabulary

A histogram is then built containing the frequency of each visual word identified in the image. The image is partitioned into four equal parts with a spatial histogram for each, see Figure 4. The histograms are then concatenated to form the feature vector that is used for testing. The resulting feature vector has 1000 entries per histogram for a total of 4000 components for each image.

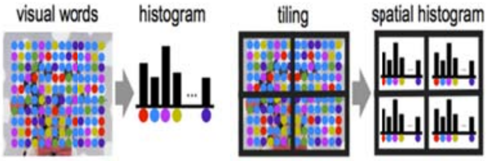

Figure 4: The spatial histogram for each partition [2]

2.4 Testing Process

After all the spatial histograms have been created for each image, the feature vectors are then entered into a support vector machine (SVM) [4] which then is used to classify the data into positive examples (cats) belonging to class +1 and negative examples (dogs) belonging to class -1. A variety of kernels were tested and the optimal meta-parameters for the SVM were determined by employing a grid search which was also cross validated.

3 Results

The entire vocabulary was built using roughly 100 distinct images per pet breed. The testing was then done also using roughly 100 distinct images per pet breed. The different breeds of cats and dogs used in this experiment can be seen in Table 1. Refer to Table 2 to see the total number of images per pet family used for the vocabulary building and testing stages.

	Cats	Dogs
Vocabulary	1087	1000
Testing	1111	1000
Total	**2198**	**2000**

Table 2: Images used for building vocabulary and testing stages

3.1 Comparing Normalization Methods of the Abyssinian and American Bulldog with 200 images per breed:

Linear Kernel – with 10-fold cross validation
Regular Images

Normaliziation Method	Accuracy (%)
L1	90.547
L2	85.074
Scaled [0,1]	91.045

Table 3: Comparing normalization methods for images with backgrounds present

3.2 Abyssinian and American Bulldog with 200 images per breed:

Linear Kernel – with 10-fold cross validation

Image Type	Accuracy (%)
Original Images	90.498
Manually Cropped Images	91.048
Cropping Animal Head	89.552

Table 4: Comparing effects of cropping images on the classifier accuracy

3.3 Using cat and dog breed from Table 1 with 200 images per breed:

Linear Kernel – with 10-fold cross validation

Image Type	Accuracy (%)
Original Images	83.704
Cropping Animal Head	84.226

Table 5: Accuracy of classifier for unmodified images and images where the animal head was cropped

		Predicted Class	
		cat	dog
Actual Class	cat	911	200
	dog	144	856

Table 6: Confusion Matrix for 3.3 original images

4 Discussion

From the first test conducted in 3.1 the classifier performed well. This was an easier task as there were only two pet breeds used and the classifier does not have to deal with variance of phenotypes of animals of the same class. For example all Abyssinian cats have similar colours, body types, and fur textures. The same goes for the American Bulldog. Interestingly enough during this test there was also an increase in accuracy of the classifier if the data was scaled between 0 and 1 inclusive as opposed to L^1 or L^2 normalization. Since this scaling performed the best it was used in all subsequent tests.

In the second test in 3.2 the results show there was little difference in the accuracy of the classifier if the animal's body or head was cropped or if the original image including the background was used. This was probably because the cropped images still contained some of the background. For this particular example the linear kernel showed to perform the best.

In the final test in 3.3 the entire dataset of 2198 images of cats and 2000 images of dogs was used. The vocabulary was built on exactly half the images. The classifier was trained and tested on the original images as well as images where the head of the animal was cropped. As expected the accuracy of this classifier was less than that found in 3.1 given that the task was more difficult because of the increase in breeds of pets among the same class; however, the classifier still performed reasonably well, see Table 5. There was also a slight increase in accuracy of the classifier from cropping the head but nothing too considerable. This might mean that for a larger variance of pet breed the head of the animal might contain more information about the pet family. The confusion matrix as seen in Table 6 showed that for the misclassifications that took place there was a tendency to predict that the image contained a dog when it in fact contained a cat. Still there did not appear to be any real significant occurrence of the classifier favouring a particular pet family.

It is also worth noting that the implementation was done using parallel processing wherever possible. By using the MATLAB parallel computing toolbox the running time of building the vocabulary and creating the histogram of visual words was considerably decreased; although this depends on the number of processes that can be run on a cluster.

5 Future Work

Future work on this classifier would include a comparison between normalization and kernel selection. For this paper the best method was found to be a simple data scaling and using a linear kernel. It would also be interesting in the future to see the effect that this has on performance among varying kernels.

In the future it would also be a good idea to use the full *Oxford-IIIT Pet Dataset* containing all 37 pet breeds of cats and dogs as this would improve the validity of this experiment.

There is a good chance that based on the work done by [1] that if image segmentation of the animal was used as opposed to image cropping the accuracy of the classifier could be increased. This might happen because currently when the animal is cropped from the image there is still pixels that contain the background remain present. Segmentation might overcome this and improve the accuracy.

Since this classifier can be used for object recognition it would be interesting to see the performance of this classifier on different categories of objects, say for example distinguishing between other species of animals.

6 Conclusion

This paper has discussed the method of object recognition by the use of the bag-of-words model. The goal was to use this model to build a classifier that had the ability to distinguish between cats and dogs. It was found that simple data scaling could be superior to vector normalization and that image classification of pet family can achieve a good accuracy of 83.7% over the pet dataset used. While this shows quite good performance given the problem, further work could be done to even further improve the accuracy.

7 References

[1] O. M. Parkhi, A. Vedaldi, A. Zisserman and C. V. Jawahar, “Cats and Dogs,” Presented at IEEE Conference on Computer Vision and Pattern Recognition. 2012.

[2] A. Vedaldi and A. Zisserman, "Image Classification Practical, 2011," Available: http://www.robots.ox.ac.uk/~vgg/share/practical-image-classification.htm [January 7, 2013].

[3] A. Vedaldi and B. Fulkerson, “VLFeat - an open and portable library of computer vision algorithms,” Presented at ACM International Conference on Multimedia, 2010.

[4] C. Chang and C. Lin, “LIBSVM: A library for support vector machines,” *ACM Transactions on Intelligent Systems and Technology* 2(3), pp. 27:1-27:27, 2011.

[5] Y. EL-Manzalawy and V. Honavar, “WLSVM: Integrating LibSVM into Weka Environment,” 2005.

[6] E. Weisstein, “Vector Norm,” MathWorld, Available: http://mathworld.wolfram.com/VectorNorm.html [January 7, 2013].