Cats Vs Dogs – Classification

Bone Jovi

Isaac Berman (260616501, isaac.berman2@mail.mcgill.ca) and

Dennis Liu (260581270,dennis.liu@mail.mcgill.ca)

Introduction

The problem for this assignment is to create a program that when given a photo of a dog or a cat, classify it as one or the other using methods learned in class. The approach we took was to extract the dog/cat from the background and obtain features from this foreground image as to not have the background interfere with our results. Then extract and compute features, and categorize those features into 2 classes based on the given label in the training file. Then we would train the SVC classifier with the set of extracted features and the 2 corresponding labels. Lastly, the test image prediction would be based on the classifier.

Problem Representation

We decided to extract SIFT features for this assignment. The reason we chose SIFT features is because they are invariant of changes to scale, size, and intensity. This is especially important in this assignment given that all the images are of different sizes, and taken under different lighting and camera positioning. Other options, such as the Harris corner detector are not scale invariant and therefore would not be suitable for this assignment. The reason we decided to implement SIFT feature extraction, as opposed to HOG is that HOG has no compensation for scale, and is best suited to contexts where there is little change in scale, such as surveillance. Since there is variable scale in our test images that we will be given, HOG shouldn't be used in this context. Originally, we also contemplated using LBP for feature extraction, however decided against it since LBP is best used to extract *texture* features. Since dogs and cat’s both have very similar texture in the many different breeds, and many of the textures overlap between the two species, we felt the two labels would not have distinct enough differences.

Algorithm Selection and Implementation

Based on our research, we chose to use SVM because it uses a smaller training set and therefore can save memory and time to train the classifier. It also seemed that for this problem, SVM with suitable parameters have the ability to prevent overfitting, a problem that is present with other classifiers. [1] So we decided to implement an algorithm that trains our SVM classifier on images that have been represented by the bag of words model, created with extracted SIFT features.

We settled on extracting the 10 best features from the training images from our testing. We found that when to many features were extracted the feature-space that was obtained was too over-saturated and tended to weigh heavily towards the “dog” label, and hence all of our images were classified as dogs when the classifier was applied. However, when too few key points were extracted, these surface-level, “best” features would tend to predict all images as “cat”. After testing we found 10 features to provide the most accurate predictions.

We decided to use LinearSVC for our implementation of SVM for image classification. This is a wrapper provided by sci-kit learn that is a wrapper around libsvm. We decided to use LinearSVC to implement our linear SVM classification because it is fast and scales well, both of which are beneficial to making a prediction on many images in this large data-set.

Testing

Our Kaggle results were not ideal as we only obtained a 59% accuracy, well below some of our classmates. We feel this could be due to a variety of reasons. Unfortunately, we never found a method of extracting the foreground that was reliable. Utilizing GrabCut sometimes would cut out the foreground instead of the background which obviously was a huge detriment to our results. Since we couldn’t manually choose the foreground for each image, we had to assume that the foreground was in the middle of image, which was not always the case and greatly skewed our results. In addition to this, we didn’t train our set on enough images to obtain the proper spectrum of the complete breeds of dogs and cats, as there are many breeds that look similar between the two, and if one is not included in training set our data could be extremely inaccurate. Modifying the number of keypoints extracted, and the size of the training set were the main alterations we made when not having satisfying prediction results. Although these changes did lead us to around 60% accuracy, further fine-tuning and added complexity to the algorithm would allow it to perform even better. A possible improvement would be that when the classifier makes a prediction, if there is an “edge-case” where the image could be a toss-up, perform LBP extraction to differentiate the texture, and make a prediction based on LBP as well. This way any toss-up predictions are not just arbitrarily guessed.

Discussion

Pros: Relatively fast training, simple and easy to follow process. Easy to check where the problems exist in the algorithm. Can be given any size of set and performance does not change dramatically. Performance is invariant to size and brightness of images provided.

Cons: Not extremely robust, sometimes grabcut blacks out entire images making them not usable in training set. Classifier could have more depth. No further classification method used for "toss-up" predictions.

References

Y. Bang Liu, K. Zhou, *Image Classification for Cats and Dogs,* Available: https://sites.ualberta.ca/~bang3/files/DogCat_report.pdf