
Objects Recognition Based on SOG with SVM

Final Report for CIS 519

Shangyi Cheng
Yao Chu
Chenyang Zhao

SHANGYI@SEAS.UPENN.EDU
CHUYAO@SEAS.UPENN.EDU
CHZHAO@SEAS.UPENN.EDU

Abstract

We studied the question of object recognition using histograms of oriented gradients (HOG) and support vector machine (SVM) with Gaussian Kernel on ball detection as a test case. After reviewing several existing methods for feature extraction, our project verified that HOG performs well in ball recognition. The whole process is shown in this report, from choosing regions of interest manually, extracting features from these regions using HOG, using principal component analysis (PCA) to reduce dimension of the dataset and then applying SVM with Gaussian kernel to classify.

1. Introduction

The topic for this project is to detect the object of our interest on the given images and predict which class it belongs to. The dataset we use is from Caltech 256 (Griffin, G. Holub, A.D. Perona, P.), which contain images of four kinds of balls, including 98 images of golf in the folder “088.golf-ball”, 174 images of soccer ball in “193.soccer-ball”, 104 images of bowling-ball in “017.bowling-ball” and 98 images of tennis ball in “216.tennis-ball”. As shown in Figure (?), among images of the same kind (take soccer-ball as example), some of the images have a whole contour of soccer in the center and fill the image, while in other images (such as a photo of a soccer game), the soccer may cover a small portion in the corner, or only part of a soccer or a group of soccers appear on the image. We decide to use the image with a single target occupying most of the image as the training set. So the first thing we need to do before training is to create such standard images for training from the raw dataset. And our final goal is, given a new image, the model can detect whether this image contains any ball of our interest or just unrelated to balls and in the first condi-

tion, the model can give the label for the class the detected object belongs to.

2. Model Details

2.1. Overview of the Method

Inspired by the paper (Dalal & Triggs, 2005) presented by Navneet Dalal and Bill Triggs, we tried the method described in this paper. The summary figure for our approach is shown in Figure .

An overview of the whole process is shown in Figure (?). For the training process, we first preprocess the images by selecting the region containing the ball manually, resize and restore such images as the dataset for feature extraaction. Then, HOG is applied on those processes images and get a vector of length 900 as features for one instance. Thus, we build the training dataset, a .dat file. Then, SVM with Gaussian kernel is used to learn from the dataset. The metrics of accuracy, ROC and learning curve along with cross-validation are used to ensure that the model has learned to classify four kinds of balls with an acceptable accuracy without overfitting. The training is finished.

For the test part, given an brand-new image, no matter whether it contains any ball of our interest, what the portion of the ball covers the image and whether it appears in the center or in the corner, individually or in group, the model scan the image using mask with increasing size, doing the corresponding manipulation with the same parameters as those for training set, and then make prediction for each block. Among all predictions, the model will choose the ones with high confidence and decide whether they point to the same object. Finally, the target of our interest will be framed in a rectangle along with its label.

2.2. Region of Interest (ROI) Selection

Before we use HOG to extract features from labeled images, we need to select the ROI which contains an object we would like our model to recognize. We tried several ways to select such region.

At the beginning, we considered using corner detection

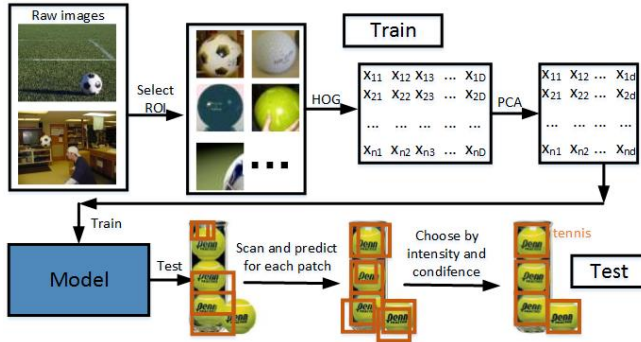


Figure 1. An illustration of HOG with SVM process.

directly on the original image to select ROI and at the same time extract features. However, as shown in Figure (?), extracting corner features based on the Harris Corner Detection and Adaptive Non-maximal Suppression method (Brown et al., 2005) is not so helpful for following reasons: firstly, different pattern can be found in a single kind of balls so that geometric corner feature doesn't provide enough information for classification; secondly, the complicated background may generate large amount of features which are irrelevant to the ball.



Figure 2. Corner Detection Results

Then we tried several methods for circle detection including (Atherton T.J., 1999), (Victor A., 2006), (D'Orazio T., 2004) to pick up the circular region and build the training set. One of the robustest approach is Circle Hough Transform (CHT), which takes the image M and desired radius range $[r_{min}, r_{max}]$ as inputs and return the positions of circle centers. As shown in Figure (?), this method performs good for the standard images with a single ball occupying the whole image but has a poor performance when there are other ball-like distractions such as heads and the three holes on a bowling-ball, or only partial of the object, rather than a whole circle, appear on the image. It's also possible for CHT to return some position with high confidence for some circular patterns inside the ball. It's really hard to pick up the ROI just by the circular shape and without any referring to the patterns and features inside the region.

Thus, we finally decided to select ROI for each image in



Figure 3. Unsuccessful CHT Detection Results

dataset manually. To increase the number of instances in our training data, for each image, let the user chooses the ROI with a movable, resizable rectangle and position it interactively using the mouse. And then shift the area a little in one of the eight directions, thus a raw image will create at most nine images and at least one image for training. Finally, the picked area is resized to a 40×40 pixel square. Apart from the four kinds of balls, we also generate some squares with the label "Unknown" to represent the unrelated features to ball. The images in this class are mostly selected from the background, fragments of the ball and other unrelated area. Figure (?) shows the processed images after manual selection and resize. First two rows are images for balls and the third row is for "Unknown".

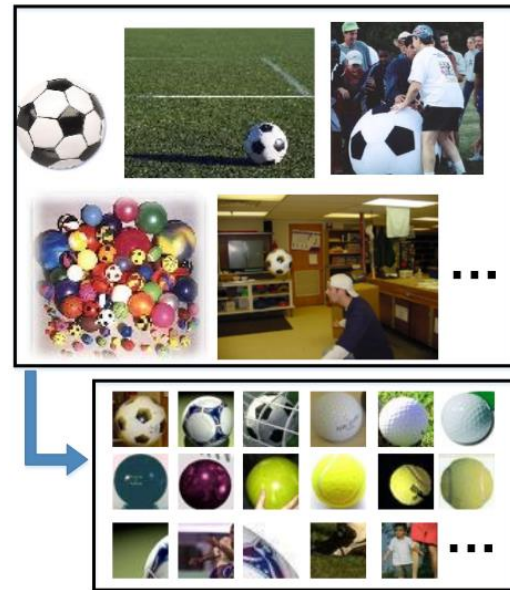


Figure 4. Resized Images Before & After ROI Selection

2.3. Histogram of Oriented Gradients Descriptors

As described in ROI selection section, the patches cropped out manually were resized to 40x40 pixels. Each patch

works as an dataset instance, however, there are two primary ways to extract features from these patches. First, each pixels value (grayscale, RGB or LAB color spaces) can be extracted directly as a feature. Second, apply HOG or SIFT to extract features. Considering the HOG/SIFT representation can capture edges and gradient structure which is characteristic of local shape as well as easily maintain the invariance to local geometric and photometric transformation, we finally chose second method. Also, HOG method can be implemented easily with fewer features compared to SIFT, so we chose to use HOG descriptors.

Implementing HOG to extract features from patches followed procedure shown in figure 6. For balls detection and classifier, following coefficient were chosen: All patches were converted into grayscale color space to normalize gamma and color; 1-D point derivative(entered [-1, 0, 1]) mask with no smoothing was applied to patches; 9 orientation bins evenly spaced over 0 180 degrees (unsigned gradient) was chosen; Cell size 6x6 and block size 2x2 were chosen for R-HOG.

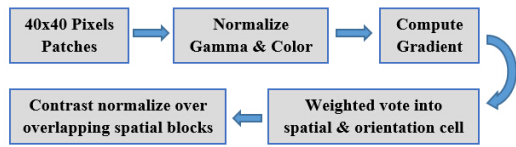


Figure 5. HOG Procedure

HOG features is shown in figure ?? . After extract HOG features from a patch, reshape the feature matrix into a vector represent the patch. Eventually we generated a 3660x900 datasets with 3660 instances and 900 features.

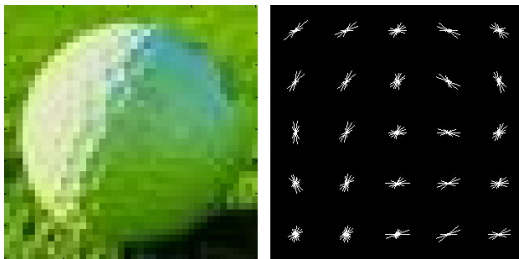


Figure 6. HOG Feature Extraction Result

2.4. PCA

The number of features we get using HOG is 900. Since the number of instances we use is about 4000, which is not large enough for 900 features and easy to result in overfit. On the other hand, to improve efficiency and decrease

the time for training, it's necessary to reduce the dimension first. On the training dataset, we compute covariance matrix Σ and mean (average over all rows of X). Next, we choose the d most important PCA basis vectors as the new training dataset. Then Σ and X got previously on training dataset should be used on the test data.

2.5. Detect Balls from New Image

Once classifier was trained, balls could be detected from the new input image based on the predict results. In order to detect the balls, the algorithm should search every possible detect window in the image. So square detect window with changeable size move around in the image to extract test patches. Window size changes from minimum value (80 pixels or 10 percent of maximum side) to maximum value (small one between image width and height). And overlapping ratio between two consecutive detect windows is 85 percent. Every patch is resized to 40x40 pixels, extracted HOG features, and normalized in the same way with training data. Following this, apply classifier to predict labels for test instances. Based on the predict results, overlap test instances can be eliminated by adaptive non-maximum suppression. A neighbor positive detection threshold is set to improve the detection result.

3. Further Work

Although our current Gaussian SVM detector is reasonably accurate for the cropped patches (test part of manually cropped ROI), there is still a problem when test thousands of test instances extracted from new input image (downloaded from other sources). False positive results were the main problem we encountered. The most likely explanation for this problem we believe is the insufficiency of the datasets, especially in the negative sets ('Unknown'). Also the original datasets for each kind ball is limited, the cropped patches generated based on the original datasets may be not representative. It would be useful to use a larger datasets to improve detector accuracy and implement cascade training method to improve the training efficiency.

1.

Acknowledgments

This project is supported by Uenn CIS 519. Thanks Eric Eaton and Xiaoxiang Hu for their instruction. Thanks MATLAB for providing so many useful toolboxes.

References

Atherton T.J., Kerbyson D.J. Size invariant circle detection. pp. 795–803, 1999.

Table 1. Confusion Matrix: B for Bowling, G for Golf, S for Soccer, T for Tennis and U for Unknown

TRUE LABEL	B	G	S	T	U	PRECISION
PREDICTION						
BOWLING	234	3	4	2	0	0.96
GOLF	21	357	10	7	10	0.88
SOCCER	12	1	393	0	1	0.97
TENNIS	6	3	4	285	1	0.96
UNKNOWN	1	1	0	0	474	0.96
RECALL	0.86	0.98	0.96	0.97	0.97	
ACCURACY	0.95					

Brown, Matthew, Szeliski, Richard, and Winder, Simon. Multi-image matching using multi-scale oriented patches. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 510–517. IEEE, 2005.

Dalal, Navneet and Triggs, Bill. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pp. 886–893. IEEE, 2005.

D’Orazio T., Guaragnella C., Leo M. Distant A. A new algorithm for ball recognition using circle hough transform and neural classifier. pp. 393–408, 2004.

Victor A., Carlos H.G., Arturo P. Raul E.S. Circle detection on images using genetic algorithms. pp. 652–657, 2006.