Objects Recognization Based on HOG with SVM
Final Report for CIS 519

Shangyi Cheng SHANGYI@SEAS.UPENN.EDU
Yao Chu CHUYAO@SEAS.UPENN.EDU
Chenyang Zhao CHZHAO@SEAS.UPENN.EDU

Abstract

We studied the question of object recognization 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 recognization. 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, AD. 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 4, 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 condition, the model can give the label for the class the detected object belongs to.

2. Methodology

2.1. Overview of the Method

Inspired by the paper (Dalal & Triggs, 2005) presented by Navneet Dalal and Bill Triggs, we tried the HOG method to extract the features vector for each individual image and apply SVM learner to train the dataset. An overview of the whole process is shown in Figure 1.

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 extraction. Then, HOG is applied on those processes images and get a vector of length 900 as features for one instance. Thus, we reconstruct the training dataset by standardizing the dataset and apply PCA method to reduce the number of features. 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.

While detecting ball in an brand-new image, 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 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. 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.

University of Pennsylvania, CIS 419/519 Course Project.

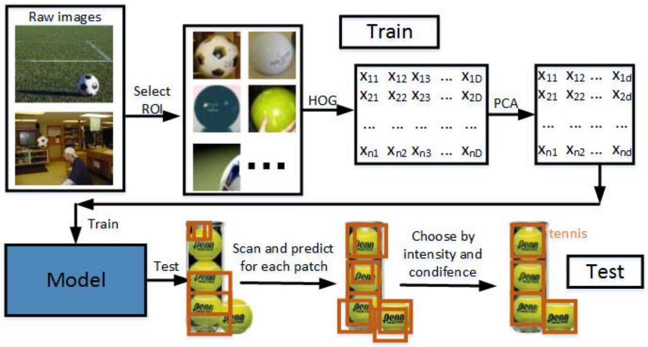

Figure 1. An illustration of HOG with SVM process.

At the beginning, we considered using corner detection directly on the original image to select ROI and at the same time extract features. However, as shown in Figure 2, 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.

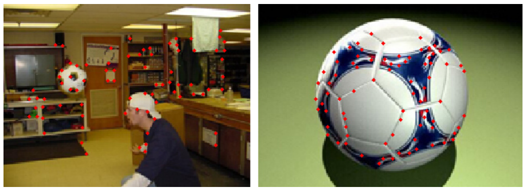

Figure 2. Corner Detection Results

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 3, this method preforms good for the standard images with a single ball occupyting 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.

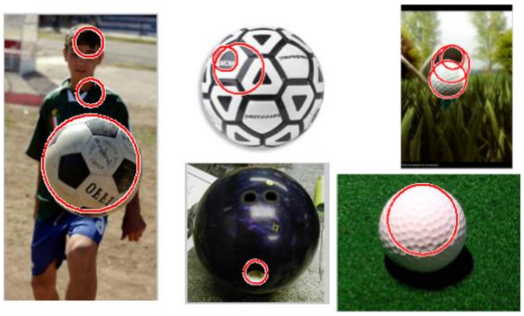

Figure 3. Unseccessful CHT Detection Results

Thus, we finally decided to select ROI for each image in 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 pixal 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 4 shows the processed images after manual selection and resize. First two rows are images for balls and the third row is for "Unknown".

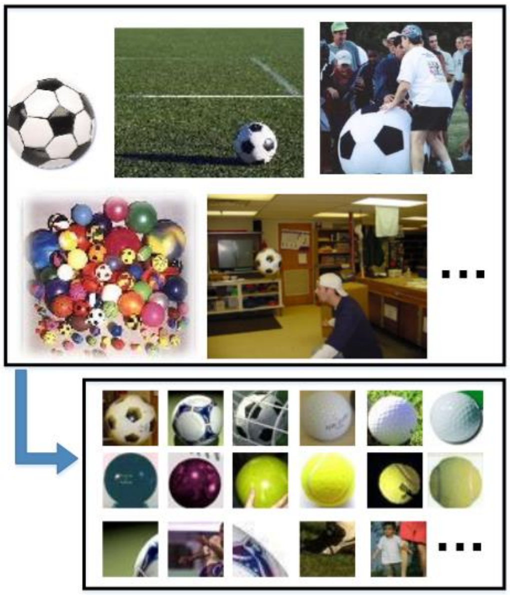

Figure 4. Resized Images Before & After ROI Selection

2.3. Histogram of Oriented Gradients Discriptors

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 discriptors.

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(centered [-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 choosen for R-HOG.

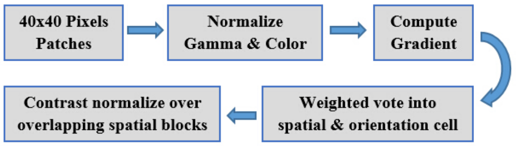
40x40 Pixels Patches
Normalize Gamma & Color
Compute Gradient
Weighted vote into spatial & orientation cell
Contrast normalize over overlapping spatial blocks

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.

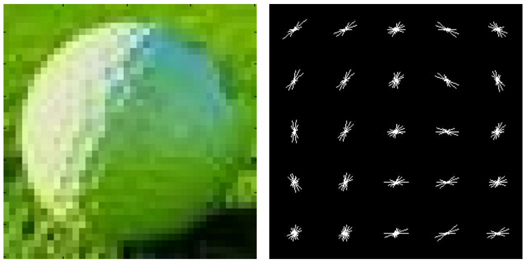

Figure 6. HOG Feature Extraction Result

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, it's necessary to reduce the dimension first. On the other hand, applying PCA would improve efficiency and decrease the time for training. On the training dataset, we firstly standardize the dataset X and record covariance matrix Σ and mean values μ. Next, we choose the d most important PCA basis vectors by calculating the eigenvalues and eigenvectors of X^TX. Then we reconstruct the new training dataset by multiply the PCA basis vectors to the origin dataset X. Support vector machine(SVM) is one of basic learning algorithm that most widely used to solve object recognition problems (Navneet D., 2005),(Massimiliano P., 1998). In our case, we applied one-class SVM first and extend it to multi-class to train the dataset with several different labels.

2.5. One-Class SVM

Given a set of instances which belongs to either of two classes, a SVM classifier finds the optimal hyperplane leaving the largest possible fraction of instances of the same class on the same side, while maximize the distance of either class from hyperplane. The objective function of SV learner is

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{d} \theta_i^2 \tag{1}$$

s.t. $y_j(\theta \mathbf{x_j}) \geq 1 \, \forall j$.
The problem of minimizing the cost function could be simplified as maximizing the $J(\alpha)$ in 2.

$$J(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{2}$$

s.t. $\alpha_i \leq 0 \forall i$, $\sum_i \alpha_i y_i = 0$, where α_i is the constraints weight scaler, and $\langle x_i, x_j \rangle$ is a scaler given by the kernel function.
Take linear kernel as an example, the parameter of the hyperplane follows that

$$\theta = \sum_{i=1}^{N} \alpha_i y_i x_i \tag{3}$$

while b can be determined from α solution of the dual problem, and from the Khn-Tucker conditions

$$\alpha_i(y_i(\theta x_i + b) - 1) = 0, i = 1, 2, ..., N \tag{4}$$

The problem of classifying a new data instance could be simply solved by looking at the sign of the score function

$$P(\mathbf{x}) = \theta \mathbf{x} + \mathbf{b} \tag{5}$$

While applying SVM algorithm, choosing proper kernel is essential to train the classifier. We used two different kernels to train the dataset, which are Gaussian kernel, linear kernel, and drew the Receiver Operating Characteristic(ROC) curve respectively. The result are shown in the figure (7,8). As shown in the figure, the two kernels give similar result while Gaussian kernel provides a faster speed to converge.

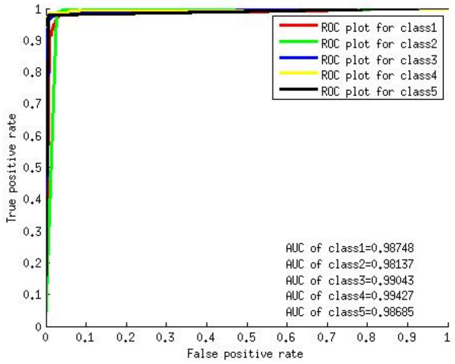

Figure 7. ROC curve of SVM with Gaussian Kernel

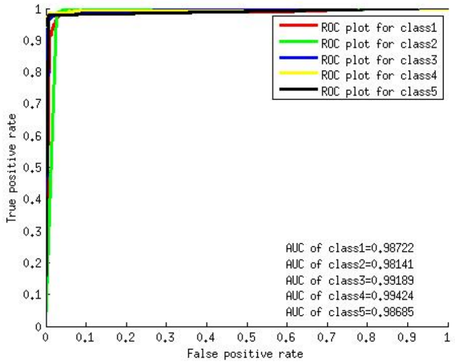

Figure 8. ROC curve of SVM with Linear Kernel

2.6. Multi-Class SVM

Traditional SVM algorithm only solves 2-classes problems. However, detecting ball object in a image should be able to tell whether there is a ball and figure out the type of the ball object, which is a multi-class classification problem. Therefore, one-vs-rest is introduced for solving multi-class problem.

Given a dataset with k different labels, k different one-class SVM classifiers is trained for each class. When training for individual class, the instances that belongs to this class are labeled as 1, and all the others are labeled as 0. A new instance should be predicted as in the class which gives the largest value from score function 5.

2.7. 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 neighbour positive detection threshold is set to improve the detection result.

3. Results and Performance Study

3.1. Learning Curve

To evaluate our SVM classifier, we run our classifier through 10-fold cross validation and drew the learning curve in figure (9) to show the performance versus the number of training examples. As shown in the figure, the testing accuracy increases as the set of training instances grows larger, and the variance of accuracies decreases.

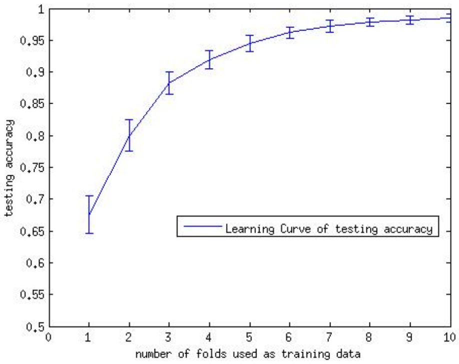

Figure 9. Learning Curve of testing accuracy

3.2. Results on Raw Images

Using the model after learning, we test it on several raw images, including the ones that we select ROI for training and new images which the model never see before. In Figure 10, we show some of the results that our model detect and label the target successfully. Note: we use different colors of rectangle to frame the area of the detected target, red for soccer, cyan for tennis, purple for bowling-ball and green for golf. In theory, our model can detect all balls with the same label on an image, such as the upper middle and right ones with several tennis balls. The lower middle image with golf is a brand-new image from website and our model can detect and label it correctly.

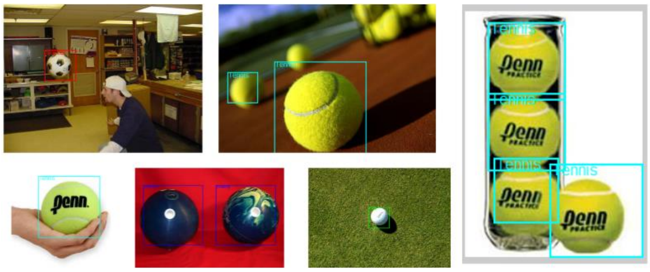

Figure 10. Successful Results for Detection and Label

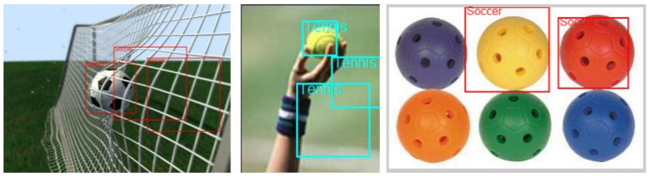

Figure 11. False Positive and True Negative Results

However, there are also some cases in which our model either fails to detect all the targets (as the right one with several bowling-balls in Figure 11), or gives false positive prediction (label some irrelevant areas as a specific kind of ball, such as the left and middle ones). We analyzed the reasons for the failure. One is that the target is sheltered by other objects have regular patterns, such as the soccer in the net shown Figure 11. Another reason is related to the size of the raw images. The image with a lifting hand holding a tennis has 96×96 pixels. Since the background of it is also green, it's more likely to detect and label the background mistakenly as tennis.

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.

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. Distante A. A new algorithm for ball recognition using circle hough transform and neural classifier. pp. 393–408, 2004.

Massimiliano P., Alessandro V. Support vector machines for 3d object recogintion. pp. 637–646, 1998.

Navneet D., Bill T. Histograms of oriented gradients for human detection. pp. 886–893, 2005.

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