

# Data Transformation of the Histogram Feature in Object Detection

Rongguo Zhang, Baihua Xiao and Chunheng Wang  
*Key Laboratory of Complex System and Intelligence Science  
 Institute of Automation, Chinese Academy of Sciences  
 {rongguo.zhang,baihua.xiao,chunheng.wang}@ia.ac.cn*

## Abstract

*Detecting objects in images is very important for several application domains in computer vision. This paper presents an experimental study on data transformation of the feature vector in object detection. We use the modified Pyramid of Histograms of Orientation Gradients descriptor and the SVM classifier to form an object detection model. We apply a simple transformation to the histogram features before training and testing. This transformation equals a small change in the kernel function for Support Vector Machines. This change is much quicker than the  $\chi^2$  kernel, but obtains better results. Experimental evaluations on the UIUC Image Database and TU Darmstadt Database show that the transformed features perform better than the raw features, and this transformation improves the linear separability of the histogram feature.*

## 1. Introduction

Object detection is an important task for image understanding and computer vision. The sliding window method [6, 7, 15] is commonly used to accomplish this. Though this sliding method usually scans over many locations in the image at multiple scales, the approach has been proved to be very effective in many situations. The sliding window approach consists of training a classifier and evaluating the trained classifier at various locations in the test image. Windows with higher confidence output are classified as object localizations in the image. Then, a postprocessing step generally uses a non-maximal suppression to avoid multiple detections of the same object.

The feature vectors of the training samples are used to train the classifier. Then the trained classifier is used to classify the candidate detection windows that are represented by feature vectors of the test samples. So the feature vectors are very important in the whole pro-

cess. The data distribution of the feature vector influences the discrimination ability of the features and then the final detection performance. In the field of computer vision, the histogram feature is commonly used [7, 16, 3, 12, 13]. Data transformations are a remedy for outliers, failures of normality, linearity, and homoscedasticity. The Box-Cox [4] power transformation on the dependent variable is a useful method to alleviate heteroscedasticity when the distribution of the dependent variable is not known. The book [10] mentioned that when variables are causal, the distribution of each variable may be approximated by a gamma density. In this case, it is advantageous to convert the distribution to a normal-like one by applying a power transformation.

In this paper, we used the PHOG [3] features to describe the image samples from the UIUC Image Database and TU Darmstadt Database. We apply a similar power transformation to the histogram feature vectors. The experimental results demonstrate that this simple data transformation is very useful for the final detection outcome. Especially for the linear kernel SVM classifier, the result is drastically improved.

The remainder of this paper is structured as follows. In Section 2 we will talk about some previous works. Section 3 reviews the feature we used in the experiments. Section 4 introduces our data transformation of the histogram feature in detail. Our experiments and the results analysis are in Section 5. The last section has the conclusion.

## 2. The Histogram Feature

Lazebnik *et al.* [12] matched two images each consisting of a 2D point set based on spatial pyramid matching [11]. Based on the image pyramid representation and the HOG [7], Bosch *et al.* [3] proposed a new descriptor called PHOG. The descriptor is to represent an image by its local shape and the spatial layout of the shape. Local shape is captured by the distribution over edge orientations and spatial layout is obtained by di-

viding the image into subregions at multiple resolution levels. Each image is divided into a sequence of much finer spatial grids by repeatedly increasing the number of divisions. This is a pyramid representation because each region at one level is divided into some regions at the next level. Bosch *et al.* [3] limit the number of levels to  $L=3$  and their HOG vectors are computed on the regions got by dividing an image by power of 2, that is  $2^0 \times 2^0, 2^1 \times 2^1, 2^2 \times 2^2$  regions.

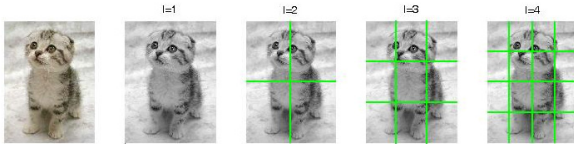


Figure 1: **An image from the PASCAL VOC 2008 [5] dataset. We divide the image into  $1 \times 1, 2 \times 2, 3 \times 3, 4 \times 4$  regions. The HOG features are computed on these regions.**

We provide some modifications to the descriptor. First, we compute the features on subdivisions got by dividing an image by consecutive integers. The idea is illustrated in Figure 1. We set the number of orientation bins  $K=20$  with the range  $[0, 360]$  using all orientations as stated in the original SIFT descriptor [13] and set the levels  $L=4$ . So the dimensionality of the PHOG vectors can be computed as:

$$Dimensionality = K \sum_{l=1}^L l^2 \quad (1)$$

Second, we normalized the features to sum to unity at each level. This normalization ensures that some small details will not be neglected directly by the whole normalization over all levels. So the 600-dimension histogram features we used in this paper are based on the PHOG with some slight modifications. Then we used the raw features and the transformed features to train the LIBSVM [5] classifier respectively. The experimental results show that the performance of the features after a simple transformation compares favorably with the standard histogram features. This expresses the effectiveness of our method.

### 3. Data transformation of the histogram feature

A Box-Cox power transformation [4] on the dependent variable is a useful method to alleviate heteroscedasticity when the distribution of the dependent

variable is not known. It is a particularly useful transformation that can often significantly improve the linear fit and approximately normalize the data. It is defined as:

$$T(X) = \begin{cases} \frac{X^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(X) & \text{if } \lambda = 0 \end{cases} \quad (2)$$

As with many statistical techniques, transformation is an iterative process which requires post calculation evaluation. In statistics, the power transform is a family of transformations that map data from one space to another using power functions. In the pages 76-77 of the book [10], the power transformation

$$Y = X^v \quad (0 < v < 1) \quad (3)$$

was applied. This is a useful method of data processing to reduce data variation and to make the data more normal distribution-like.

In computer vision, the distribution of histogram features can be approximated by a gamma density function [10]. So the simple power transformation we applied to the histogram feature is very effective [10]. In previous work [12, 11], the distance function measuring the similarity between the images was histogram intersection. The paper [3] showed that a  $\chi^2$  distance has superior performance to histogram intersection. In our paper, we just apply a simple power transformation to the histogram feature as follows:

$$new\_vector = (raw\_vector)^r \quad (0 < r < 1) \quad (4)$$

Its effectiveness is similar to using  $\chi^2$  kernel on the raw feature. However, this power transformation is much simpler than using  $\chi^2$ . Using this transformation with the RBF kernel can be expressed in the following equation:

$$K(x, y) = \exp(-\rho \|x^r - y^r\|^2) \quad (\rho > 0) \quad (5)$$

Using much less time than  $\chi^2$ , our transformation can obtain better results than using  $\chi^2$  kernel. The comparison is shown in Table 1 and Figure 2.

To demonstrate the influence of different values of the parameter  $r$ , we use some values of  $r=0.1, 0.3, 0.5, 0.7, 0.9$  and  $1.0$  (that's without transformation.). We use these data transformation in our detection model for the detection of the single-scale cars. The test results show that the results based on  $0 < r < 0.5$  are better than based on  $0.5 < r < 1.0$ . When  $0.3 \leq r \leq 0.5$ , the results are best. So in the following experiments, we just assign  $r=0.5$  for ease of calculation.

Table 1: The time cost on the UIUC single-scale test set and multi-scale test set using Linear, Rbf,  $\chi^2$  and the Linear and Rbf kernel with our transformation. We can see that our method is much quicker than  $\chi^2$  kernel.

method(kernel)	Linear	Rbf	$\chi^2$	the transformed Linear	the transformed Rbf
Time on single-scale set(minutes)	0.64	0.56	14.9	0.68	0.73
Time on multi-scale set(minutes)	4.48	4.23	98.56	4.82	5.81

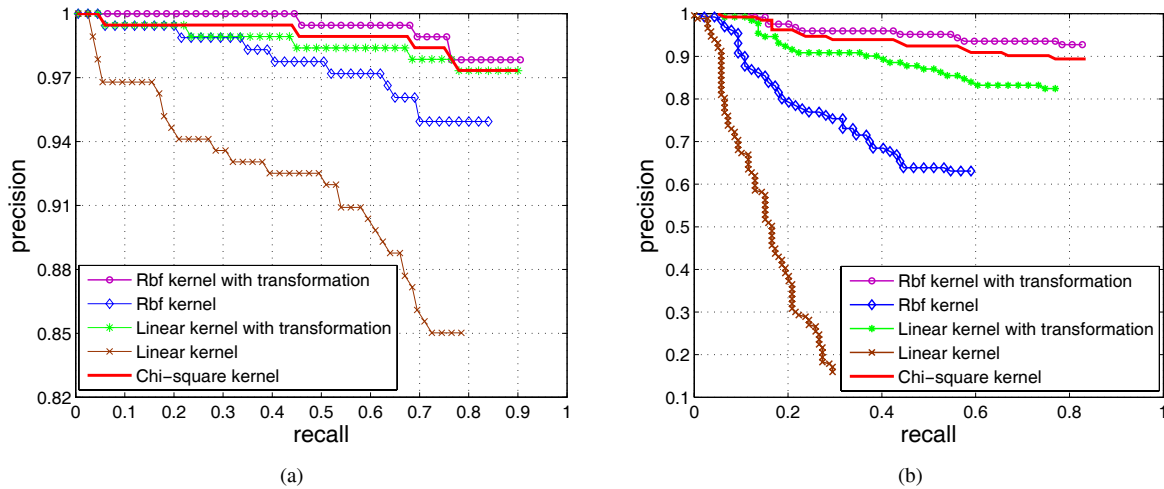


Figure 2: The precision-recall curves of the detection results' comparison. (a) The result of UIUC single-scale test set. (b) The result of UIUC multi-scale test set. We can see that our transformation with the RBF kernel has the best performance.

Table 2: Performance on TU Darmstadt cows dataset with the best previously reported *implicit shape model (ISM)*, *local kernels (LK)*, their combination [9] and the structured training method [2]. The last two columns are the results got by the raw features and the transformed features.

method	ISM	LK	LK+ISM	structured training	the raw Rbf	the transformed Rbf
performance	96.1%	95.3%	97.1%	98.2%	<b>94.7%</b>	<b>98.0%</b>

## 4. Experiments

### 4.1. Dataset and evaluation criteria

We use the UIUC Image Database for Car Detection. This dataset contains 550 training cars, 500 negative training examples, a single-scale test set and a multi-scale test set. The original ground truth data was provided in terms of bounding boxes for the cars. The single-scale test set consists of 170 images containing 200 cars; the cars in this set are all roughly the same size as in the training images. The multi-scale test set consists of 108 images containing 139 cars; the cars in this set are of different sizes, ranging from roughly 0.8 to 2 times [1] the size of cars in the training images. The TU Darmstadt cow dataset consists of 111 training and 557 test images of side views of cows in front of

different backgrounds [14].

The performance is measured using the evaluation codes provided in the PASCAL VOC 2008 [8] challenges. Detection is considered correct when the result overlaps more than 50% with the corresponding ground-truth bounding box. The precision-recall curves and the average precision (AP) are used to measure the results.

### 4.2. Experimental results

We used the UIUC training set of 1050 labeled images to train a linear SVM classifier and a RBF SVM classifier. The classifiers are used to detect the single-scale test set and multi-scale test set respectively. First, we use the raw feature vectors to represent the training and the testing samples to get a detection result. Then, we apply the data transformation to the feature

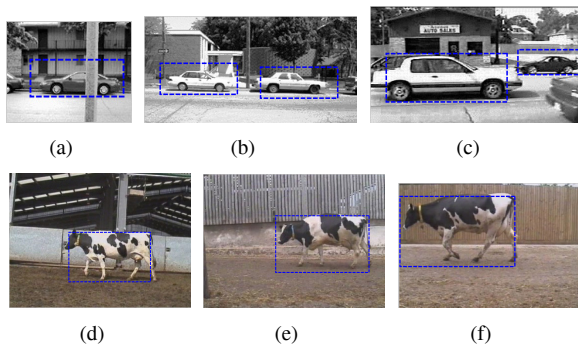


Figure 3: **Some detection results. (a) An image containing one car with some occlusion. (b) An image containing two cars. (c) An image containing two different scale cars. (d)-(f) Some images from TU Darmstadt cows dataset.**

vectors to get another result. We compare the results with precision-recall curves in Figure 2. We can see that the results with the data transformation are all better than the raw feature vectors. Especially for the linear SVM, there are obvious improvements for both the single-scale test set and the multi-scale test set. In Table 1 and Figure 2, we can see that the Rbf kernel with our transformation performs best. Our method obtains better results than  $\chi^2$  kernel, in much less time.

Using the data transformation in our detection model, we achieve the best average precision 99.7% for the single-scale car test set and 98.2% for the multi-scale car test set. These both improve the detection performance in [1]. And our precision-recall curves are much better.

We also achieved a comparable detection result with the TU Darmstadt cows dataset. Our method with the transformed features improves over the most previous methods as shown in Table 2. So in general, the data transformation of the histogram features helps to improve the final detection result. Some detection results in our experiments are illustrated in Figure 3.

## 5. Conclusion

In the field of computer vision, the histogram feature is the most common feature in applications. Our main contribution is demonstrating a data transformation of the histogram feature to improve the performance in object detection. We just apply a simple power transformation to the feature vector. This transformation improves the linear separability of the histogram feature. The experimental results show that the features after transformation perform better than the raw features.

The result also outperforms some previous methods. In the future, we will add the appearance feature combining the existing shape feature to further improve the detection performance.

## References

- [1] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. In *IEEE Transactions on PAMI*, number 2004, pages 26(11):1475–1490.
- [2] M. B. Blaschko and C. H. Lampert. Learning to localize objects with structured output regression. In *ECCV,2008*.
- [3] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *CIVR,2007*.
- [4] G. E. P. Box and D. R. Cox. An analysis of transformations. In *Journal of the Royal Statistical Society,1964*, pages Series B 26: 211–246 <http://www.jstor.org/stable/2984418>.
- [5] C.-C. Chang and C.-J. Lin. Libsvm : a library for support vector machines. In *Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm,2001*.
- [6] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *Computer Vision and Pattern Recognition, 2007*, pages 1–8.
- [7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005*, pages 886–893 vol. 1.
- [8] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge 2008 results. In *2008*, pages <http://www.pascal-network.org/challenges/VOC/voc2008/workshop>.
- [9] M. Fritz, B. Leibe, B. Caputo, and B. Schiele. Integrating representative and discriminative models for object category detection. In *ICCV,2005*, pages 1363–1370.
- [10] K. Fukunaga. Introduction to statistical pattern recognition. In *Academic Press, New York,1990*.
- [11] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV,2005*.
- [12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR,2006*.
- [13] D. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV,2004*, pages 60(2):91–110.
- [14] D. Magee and R. Boyle. Detecting lameness using 'resampling condensation' and 'multi-stream cyclic hidden markov models'. In *Image and Vision Computing,2002*, pages 20, 581–594.
- [15] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR, 2001*, pages 511–518 vol.1.
- [16] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR,2006*.