Feature Context for Image Classification and Object Detection

Xinggang Wang*[1] Xiang Bai[1] Wenyu Liu[1] Longin Jan Latecki[2]
[1] Dept. of Electronics and Information Engineering, Huazhong Univ. of Science and Technology, China
[2] Dept. of Computer and Information Sciences, Temple Univ., Philadelphia
wxghust@gmail.com, xiang.bai@gmail.com, liuwy@hust.edu.cn, latecki@temple.edu

Abstract

In this paper, we presents a new method to encode the spatial information of local image features, which is a natural extension of Shape Context (SC), so we call it Feature Context (FC). Given a position in a image, SC computes histogram of other points belonging to the target binary shape based on their distances and angles to the position. The value of each histogram bin of SC is the number of the shape points in the region assigned to the bin. Thus, SC requires knowing the location of the points of the target shape. In other words, an image point can have only two labels, it belongs to the shape or not. In contrast, FC can be applied to the whole image without knowing the location of the target shape in the image. Each image point can have multiple labels depending on its local features. The value of each histogram bin of FC is a histogram of various features assigned to points in the bin region. We also introduce an efficient coding method to encode the local image features, call Radial Basis Coding (RBC). Combining RBC and FC together, and using a linear SVM classifier, our method is suitable for both image classification and object detection.

1. Introduction

Current methods for image classification and object detection are based on the information collected from local image descriptors. Traditionally, vector quantization (VQ) is used for coding the local image descriptors. Firstly, a codebook is generated by a clustering method *e.g.* k-means algorithm. Then each descriptor is assigned to its nearest neighbor(s). As demonstrated by recent results of Sparse Coding [27] and Local Coordinate Coding [29], a better coding can significantly improve image classification results. In order to speed up the Local Coordinate Coding, Wang *et al.* proposed the Local-constrained Linear Coding (LLC) [26], which encodes a local descriptor by solving a small scale least square problem. All there methods consider coding as either ℓ_1 or ℓ_2 approximation problem, i.e., of finding a set of codewords that base approximate a given local descriptor. We consider the coding problem from another view. We use codewords with highest activation potentials to encode a given local descriptor. Thus, we treat each codeword as a Radial Basis Function (RBF) whose activation potential is proportional to its similarity to the local descriptor. Consequently, in our framework each codeword is not only characterized by its location in the feature space but also by its RBF activation kernel, which gives us more flexibility in adjusting to high variability of the density of local descriptors. In other words, our coding takes place in the kernel space, and we represent each codeword by its kernel coordinates. From the point of view of the density estimation, the codewords (RBFs with their activation kernels) provide a nonparametric density estimation the probability density of the local descriptors.

To have a compact representation of thousands of local descriptors in an image, the *bag-of-features* (BoF) method [22, 7], which represents an image as an orderless collection of local features has been widely used. However, the BoF method disregards all information about the spatial layout of local features, hence it is incapable of capturing or locating an object. To overcome this problem, the most efficient and effective extension of BoF is the *Spatial Pyramid Matching* (SPM) method [12]. Motivated by [11], the SPM method partitions image into increasingly finer spatial sub-regions, and computes histograms of local features for each sub-region. Typically, $2^l \times 2^l$ sub-regions, $l = 0, 1, 2$ are used.

On the other hand, shape based object detection community, e.g., [32, 23], often relays on a semi local shape descriptor, called Shape Context (SC) [3]. SC is a robust shape descriptor, where a binary shape is represented as a discrete set of points usually sampled from its contour. Given a reference point, these points are mapped into a log-polar coordinate system centered at the reference point. Each bin of the log-polar space is determined by angle and distance intervals. SC counts the number of the sample points within

[*] Part of this work was done while the author was at Temple University.

each bin and represents the counts as a 2D histogram. The proposed descriptor is motivated by this idea. The key difference lies in the fact that we do not know the set of sample point from the target object. Therefore, we replace the sample points with codewords representing local image descriptors. We choose max-pooling [27] method to extract the most relevant local descriptors in each bin together. Since we have many different possible codewords in each bin, we arrive at a 3D histogram of codewords for a given reference point. We call this histogram a Feature Context (FC). FC contains not only the information of each local descriptor, but also the spatial distribution of the local descriptors. As the number of reference points gets large, the representation of the image content becomes exact so that we can exactly identify the locations of target objects.

The remainder of the paper is organized as follows: Section 2 gives the related work; Section 3 presents the *Radial Basis Coding* and Section 4 presents the *Feature Context*; Section 5 gives the implementation details; Section 6 carries out experiments using our method for both image classification and object detection; Section 7 conclusions are made, and future research further research issues are discussed.

2. Related Work

A few recent papers propose to extend bag of features taking in consideration spatial information about features. Marszalek and Schmid [15] used spatial weighting to reduce the influence of background clutter in object detection. Cao *et al*. [6] designed different kinds of bag of features by projecting local features into different lines and circles, then using boosting to select representative features to do image retrieval. Bronstein *et al*. [5] constructed spatial-sensitive bag of features based on a vocabulary of pairs features, but it is not efficient enough for large scale computer vision task. The most successful approach is the the *Spatial Pyramid Matching* (SPM) method [12]. Our method also divides an image into regions, but our regions have circular shape in log polar coordinate system. The models designed to describe the contexture information of objects usually tend to utilize the information surrounding the objects, *e.g*. [24, 31], while our FC model focus on encoding the spatial information inside the objects.

Recently proposed, new coding approaches [27, 28, 29, 26] combined with linear SVM classifier have get great success in image classification. Yang *et al*. [27] propose the ScSPM method in which sparse coding is applied to local SIFT descriptors densely extracted from the image and SPM over the sparse codes is used to obtain the final image representation. In [29] sparse coding is approximated by a locally linear model. More efficient coding algorithms than [27] and [29] have been developed in [28] and [26]. The motivation of the proposed RBC is very different from these methods. Moreover, RBC can achieve a very high accuracy while coding local features in a linear time.

3. Radial Basis Coding

Let X be a set of D-dimensional local descriptors for training the codebook, they can be randomly extracted from training images, *i.e.* $X = [x_1, x_2, ...x_M] \in \mathbb{R}^{D\times M}$. The popular k-means clustering aims to partition the M training vectors into K sets ($K < M$) $S = \{S_1, S_2, ..., S_K\}$ by minimizing the within-cluster sum of squares:

$$\arg\min_S \sum_{i=1}^{K} \sum_{x_j \in S_i} \|x_j - \mu_i\|^2 \tag{1}$$

where μ_i is the center of the cluster S_i. For each cluster S_i, we also compute the standard deviation σ_i of the distances of points in S_i to μ_i.

The proposed coding is inspired by the classical perceptron and the Radial Basis Functions (RBF) network. A given point $x \in \mathbb{R}^D$ is encoded according to its activations of neurons placed at the cluster centers $\mathcal{C} = \{\mu_1, \mu_2, ..., \mu_K\}$. The activation strength is measured with a (truncated) Gaussian

$$w_i = G(\frac{x-\mu_i}{\sigma_i}) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_i} exp(\frac{-\|x-\mu_i\|^2}{2\sigma_i^2}), & \mu_i \in \mathtt{NN}(x) \\ 0, & otherwise \end{cases} \tag{2}$$

where NN(x) denotes a set nearest neighbors of x. We restrict the activation to NN(x) for efficiency reasons. The number of nearest neighbors N is usually between 3 to 10.

Let f represents the unknown density of the local feature descriptors of images in $\mathbb{R}^D$. We obtain that the density at point x is proportional to

$$f(x) \propto \sum_{i=1}^{K} w_i = \sum_{i=1}^{K} G(\frac{x-\mu_i}{\sigma_i}). \tag{3}$$

Consequently, the proposed coding is related to nonparametric density estimation, also called kernel density estimation.

However, if point x is far from all codewords, all w_i will be close to zero. To avoid this problem, the activation potentials w_i are normalized so that

$$\sum_{i=1}^{K} w_i = 1. \tag{4}$$

Finally, point x is encoded as vector

$$\mathcal{C}(x) = (w_1, w_2, ..., w_K),$$

i.e., $\mathcal{C} : \mathbb{R}^D \to \mathbb{R}^K$. Therefore, we call $\mathcal{C}$ the **codebook** and each μ_i is a **codeword**.

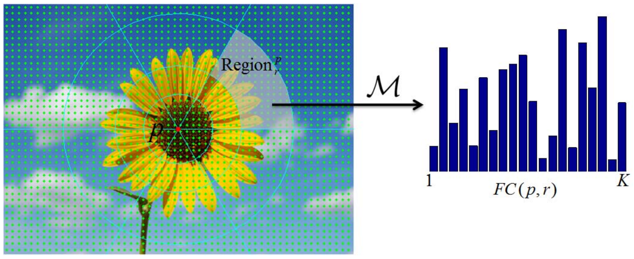


Figure 1. Illustration of Feature Context, the green dots on in the image are positions of local descriptors. The histogram is constructed by pooling codewords located in the highlighted region.

In contrast to recent coding approaches [27, 29, 26], we neither minimize L_2 nor any other L_p norm between x and the linear combination $\sum_{i=1}^{K} w_i \mu_i$. In our framework, we simply map x to new coordinates in the kernel space determined by the codewords.

4. Feature Context

In a given image I, we have a set $Z = \{z_1, \ldots, z_L\}$ of feature point locations. Each feature point $z \in Z$ is encoded as vector $C(z) = (w_1^z, \ldots, w_K^z)$. Let p be a location of an reference point in I. Following SC, the area around p is divided into regions $Region_r^p$ in log polar coordinate system for $r = 1, ..., R$, e.g., see Fig. 1. The **feature context (FC)** around point p is defined as a matrix

$$FC(p, r, i) = \mathcal{M}\{w_i^z \mid (z - p) \in \Delta_s(Region_r^p)\}, \quad (5)$$

where $i = 1, \ldots, K$ indexes the codewords and $\mathcal{M}$ is a **pooling function**, which extracts the most relevant codewords present in the region $Region_r^p$. The function $\mathcal{M}$ can be max, sum, $mean$ or some other functions. We selected max pooling function as $\mathcal{M}$ in our experimental results, since the max pooling method is more robust to local transformation than mean statistics in histogram [27]. $\Delta_s(Region_r^p)$ denotes a neighborhood of region $Region_r^p$ of radius s. It allows us to compensate for spatial uncertainty of local descriptor. As the illustration in Fig. 2, the local descriptors near the boundaries of regions may belong to multiple regions. By using $\Delta_s(Region_r^p)$ in Eq. (5), they are assigned to those regions, which increases the robustness of our Feature Context descriptor. In image I, we usually have a set of reference points as $P = \{p_1, p_2, ..., p_L\}$. Therefore, FC of an image I is a tensor of dimension $L \times R \times K$ given by

$$FC(I) = (FC(p, r, i))_{p,r,i} \in \mathbb{R}^{L \times R \times K}.$$

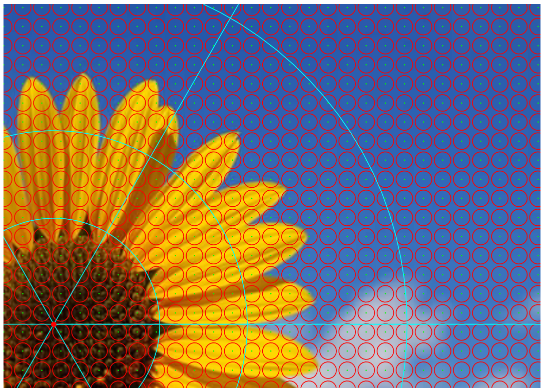


Figure 2. Illustration of spatial uncertainty. The green dots in the image are positions of local descriptors. The local descriptors near the boundaries of regions may belong to multiple regions as illustrated by the red circles.

5. Implementation Details

5.1. Image classification

Each image is represented by a set of local descriptors with their spatial coordinates. We use the densely extracted SIFT feature [14] implemented by [13], but other descriptors computed from image patches could also be used. The image classification framework consists of three steps:

1. *Local descriptor coding using RBC*

   As described in Section 3, we randomly select millions of patches in the training images to generate a codebook with k-means clustering algorithm. Then, encode all dense SIFT features using Eq. (2), and normalize the code using Eq. (4).

2. *Spatial information coding with FC*

   For image classification, we use nine reference points positioned as shown by the red dots in Fig 3. Similar to Shape Context [3], the radius of the circle of a reference point is the mean distance of all local descriptors to the reference point. Denote the radius as R_l, where l is the index of reference point, the parameter of spatial uncertainty s is set to be 0.1 of R_l, for object detection is the same. Different reference points can be seen as different perspectives to describe the image content. The final representation of an image is a concatenation of the feature vectors of all reference points.

3. *Classification using linear SVM*

   Each feature vector as the final representation of image is normalized and fed into a linear SVM. Here we use ℓ_2 normalization and LibLinear SVM [9], LibLinear is a efficient linear SVM software for large scale classification problems.

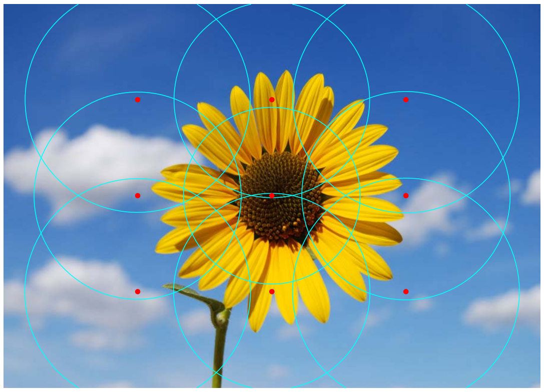


Figure 3. For image classification, we use nine reference points positioned as shown by the red dots. The circles illustrate the extend of the log polar coordinates of each reference point.

5.2. Object detection

In this part we present a framework for detecting objects using RBC and FC, which is based on the sliding window mechanism. As in the image classification framework, we are still using dense SIFT as the local descriptor, and encode it using RBC.

The training data includes a set of images containing instances of objects in target classes annotated by their bounding-boxes. Positive training examples are collected from the bounding-boxes while negative training examples are randomly collected from the background. At the same time, the system computes the mean dimension of positive training windows (width M_w and height M_h). To speedup the detection process, we design a two-stage object detector.

Stage 1. In this stage, we aim to reject many negative testing examples, and give some candidates in the testing windows that may contain target objects. Here we use the traditional BoF representation with sum pooling. Taking advantage of the integral histogram method [19], we can perform very fast object detection. A threshold is set to obtain candidate regions from the probability maps obtained by the linear SVM classifier.

Stage 2. In this stage, we use our FC method with max pooling to describe each window, and linear SVM is also used for classification. However, now the windows are restricted to the regions obtained in Stage 1. For a given window of size $M_w \times M_h$, the position of the reference points is determined the following way. We assume that $M_w > M_h$, otherwise the calculation is performed with M_w and M_h interchanged. The number of reference points is $N_r = round(M_w/M_h)$, and the coordinates of the reference points are $(\frac{1}{2\times N_r}M_w, \frac{1}{2}M_h), \ldots, (\frac{2\times N_r - 1}{2\times N_r}M_w, \frac{1}{2}M_h)$. The radius of the reference points is the same and it is set to $\sqrt{(\frac{1}{2\times N_r}M_w)^2 + (\frac{1}{2}M_h)^2}$.

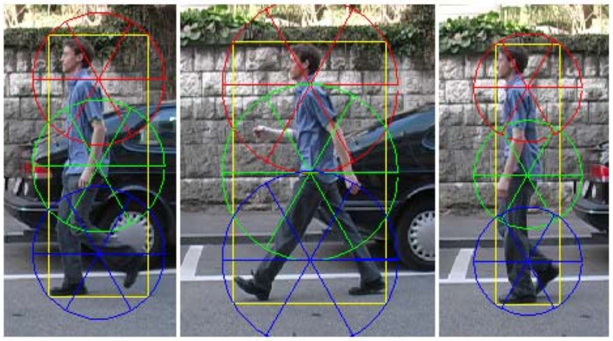
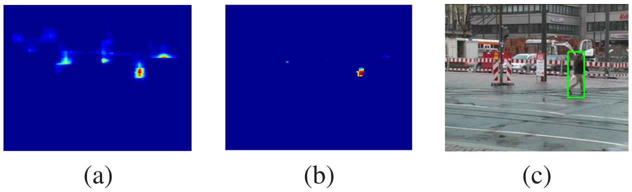


Figure 4. Examples of training images for pedestrian detection. We use three reference points and scale the Feature Context regions according to the training bounding boxes.



(a) (b) (c)

Figure 5. Two-stage pedestrian detector. (a) the probability map output from stage 1; (b) the probability map output from stage 2; (c) the final detection result (at equal error rate).

For example, there is a pedestrian with three different poses in Fig. 4. In this case $N_r = 3$, the there circles plotted in red, green and blue represent their log polar coordinate systems. Though the poses of pedestrians are different, same parts are likely to be located in the same bins (with spatial uncertainty), *e.g.*, the faces and legs. By this kind of accurate representation, the detector can reject more hard false positive examples from the first stage and affirm the true positive examples.

In order to illustrate the benefits of the two stage detection we consider the example in Fig. 5. There, (a) is the probability map output from stage 1; there are several positions with high probability due to the pillar and cluster. Then in stage 2, as shown in (b), the pedestrian probability at false positive locations decreases a lot, while the probability of true positive location increases. The final detection result in (c) is obtained via non-maximum suppression.

Usually, to search a pedestrian with 9 scales in a 1500×1200 image takes 8 to 15 seconds with the step size of sliding window of 5 pixels. We use the C++ and Matlab mixed implementation and our CPU is Core i7 @ 2.80GHz.

6. Experimental Evaluation

We have evaluated our classification performance on two standard datasets Caltech 101 [10] and 15 Scene Categories [12]. In order to demonstrate the quality and variably of our

Algorithms	15 training	30 training
Zhang[30]	59.10±0.60	66.20±0.50
Boiman[4]	65.00±1.14	70.04
Gemert[25]	-	64.14±1.18
Lazebnik[12]	56.40	64.40±0.80
SC+SPM[27]	67.0±0.45	73.2±0.54
LLC+SPM[26]	65.43	73.44
RBC+SPM	68.81±0.48	75.58±0.82
RBC+FC	**69.63±0.84**	**77.09±0.74**

Table 1. Classification rate (%) comparison on Caltech-101 dataset.

detection approach, we evaluate detection performance on the Graz 17 dataset [18], TUD-Pedestrians dataset [1]. In both cases we compare to the state-of-the-art methods. All the classification experiments for each dataset are repeated ten times with different randomly selected training images, and the average of the pre-class classification rates is recalled for each run. The final result is reported as the mean and standard deviation of the results from individual runs. In the object detection experiments, to identify the correct detections, we use the PASCAL criterion, to evaluate the detection performance.

6.1. Caltech 101 dataset

The Caltech-101 dataset contains 9144 images in 101 classes including animals, vehicles, flowers, etc, with significant variance in shape. The number of images per category varies from 31 to 800. We follow the common experiment setup for Caltech-101, training on 15 and 30 images per category and testing on the rest, and measure the performance using average accuracy over 102 classes, (101 object classes and a ”background” class). The images are resized to be no larger than 300 × 300 pixels with preserved aspect ratios. Following the experiment setup of Wang *et al*. in [26], the SIFT features are extracted from patches densely located by every 8 pixels on the image, under three scales, 16×16, 24×24, 32×32 respectively. We train a codebook with 2048 codewords, use 9 reference points for FC, and the number of nearest neighbors for RBC is 5. For comparison purpose, we also run SPM with 4×4, 2×2 and 1×1 sub-regions.

In Table 1, we compare our method to the state-of-the-art methods, which can be divided into two categories, one is using either nearest neighbors or non-linear classifiers to do the image classification, and the other category is using coding to encode local descriptors and then linear SVM for classification. We use a linear SVM classifier. The proposed feature descriptor FC and the coding method RBC outperform the other methods. Moreover, our results clearly demonstrate that RBC is superior to Sparse coding [27] and LLC [26], and FC works better than SPM on this dataset.

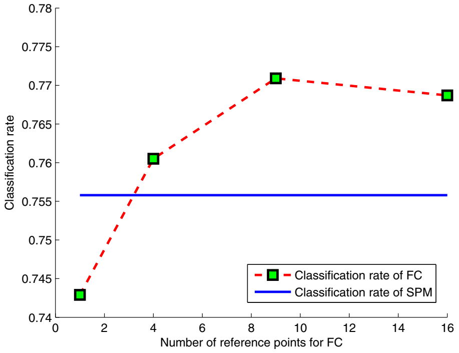


Figure 6. The classification rate of FC for different numbers of reference points. The blue curve is the classification rate of SPM encoded with RBC coding. It is shown as a baseline.

The time complexity of RBC also compares favorably to LLC. First both methods must find nearest neighbors, resulting in the same time complexity. Then encoding procedure of RBC takes $\mathcal{O}(N)$, while encoding procedure of LLC takes $\mathcal{O}(N^2)$, where N is the number of nearest neighbors.

In Fig. 6, we evaluate the classification rate of FC under different numbers of references points. The numbers of reference points are 1, 4, 9 and 16, and they are placed at equidistant gird points in the image as illustrated in Fig. 3. For comparison, we also include the classification rate of SPM encoded with RBC coding. It is shown as the blue line (its value does not change, since it is not related to the number of our reference points). As number of reference points increases from 1 to 9 the classification rate increases, but the classification rate decreases for 16 reference points. This is because the representation of image become too specific, making it too sensitive to the in-class variance. In fact, as the number of reference points increases, the length of feature vector also increases. Thus, 9 points is the best choice for both effectiveness and efficiency.

6.2. 15 Scene Categorization

We also evaluate our algorithm on the 15 scene dataset, which was collected by several researchers [16, 10, 12]. It contains the total of 4485 images divided into 15 categories, each category has 200 to 400 images, and average image size is 300 × 250 pixels.

Following the same experiment setup as in Lazebnik *et al*. [12], we take 100 images per class for training and use the remaining images for testing. We use single scale dense SIFT features computed over a grid with spacing of 8 pixels, the patch size is 16 × 16 pixels, and the size of codebook is 400. The number of FC reference points is 9 and the number of nearest neighbors used in RBC is 5.

The results are shown in Table 2. Again the proposed

Algorithms	Classification rate
VQ+SPM+IKSVM[12]	81.40±0.50
SC+SPM+Linear SVM[27]	80.28±0.93
RBC+SPM+Linear SVM	80.18±0.77
RBC+FC+Linear SVM	81.55±0.32
VQ+FC+IKSVM	**83.37±0.59**

Table 2. Classification rate (%) comparison on 15 Scene dataset.

feature descriptor FC and the coding method RBC outperform the other methods and this is the case for linear SVM. When using SPM, the classification rate of RBC is nearly identical to sparse coding [27], but RBC is more efficient than sparse coding. [12] uses simple vector quantization to encode SIFT features, SPM to encode the spatial information, and a intersection kernel SVM to do the classification, which yields a very good classification result. When we follow their framework, but using FC instead of SPM, we get a nearly 2 percent improvement. In particular, this demonstrates that FC can work well with both linear SVM and non-linear SVM.

6.3. Graz 17 dataset

The Graz 17 dataset [18] is a multi-class dataset composed of 17 object categories. We had to exclude some classes from the performance evaluation, since they contain a very small number of training images. There are 8 classes contains more than 45 training images, but the Motorbikes class has wrong entries in the ground-truth. Therefore, we evaluate the object detection performance on 7 classes.

For each of the 7 classes, we extract multiple scales dense SIFT features for both training and testing images, the scales of dense SIFT feature are determined according to size of objects. Number of nearest neighbors in RBC is 3. During detection, 5 different scales of sliding windows are searched.

In Fig. 7, we show the FC models of the 7 classes, the number and positions of FC model is calculated as the description in Section 5.2. The second and third row are some of the detection results at equal error rate, in the middle of third throw, the position of the bike with large occlusion can be perfectly detected using our FC method.

We plot recall-precision curve for each class, then calculate the area under the curve (AUC) and equal error rate (EER) to compare with two state-of-the-art methods [21] and [17]. As show in Table 3, we use the same number of training and testing images with [21] and [17]. Under the RP AUC measure, we get higher results than [21] in 5 of the 7 classes. Under the RP EER measure, we also get higher results than [17] in 5 of the 7 classes.

We also report the performance of BoF method, the only difference between BoF and FC is that BoF disregard all the spatial information in each window while FC encode the spatial information. By using FC, there is a significant improvement in the classes airplanes, cars (rear), bikes (side), and bottle. The large performance increase as compared to BoF, over 38% on airplanes, over 47% on faces, and over 43% on bikes, clearly demonstrates the benefits of FC for objects with variable but relatively stable spatial layouts. In contrast, the improvement of FC is not so obvious on the horses (side) class. This is due to the significant changes in poses of horses. Although FC tolerates moderate object deformations, significant pose changes drastically change the spatial layout of object parts. Consequently, FC requires then a significantly larger number of training images.

6.4. TUD-Pedestrians

We also evaluate object detection performance of the proposed approach on the TUD-Pedestrians dataset [1]. This dataset consists of several series of video images containing side-view pedestrians. It provides two training datasets, one has 210 images and another has 400 images. The test part consists of 250 images with 311 pedestrians with large variability in poses, articulation, size, and clothing.

For all the training and testing images, the SIFT features are extracted under three scales, 16×16, 24×24, 32×32 pixels in step size of every 6 pixels. We use 10 nearest neighbors in RBC encoding.

Our training method only uses the bounding box information. For every training image, we randomly select 10 negative training examples outside the true bounding boxes in different scales. The background of training and testing images is not consistent, which is a big challenge for any kind of discriminative method based on local image descriptors.

We have trained our detector using 400 training images. In table 4, we have compared our recall value at equal error rate and final detection rate with other methods. The results of other methods are taken from [1]. Our method outperforms the popular human detector, HOG [8], and the 4D-ISM [20], which is specially designed to detect people in cluttered scenes with partial occlusion. The best detection result on this dataset is obtained by Andraluka *et al.* in [1] (in [2], the detection result has been improved, but just a little). Their method is specially designed for pedestrian detection and utilizes explicit part model with a structure manually designed to represent pedestrians and a sophisticated statistical inference framework. In contrast, we have a universal object detection method, where learning the appearance only requires a set of training examples of target objects. Some of our pedestrian detection results at equal error rate are shown in Fig. 8. They demonstrate that by better coding the local descriptors and their spatial information our pedestrian detector is robust to noisy background. We also illustrate the main problem of our approach on this

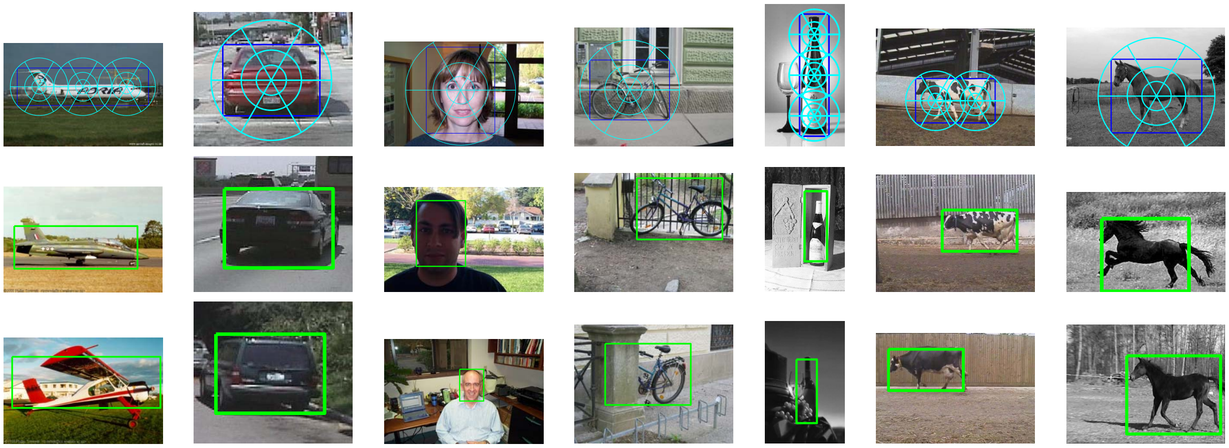


Figure 7. The first row illustrates our FC model for 7 classes of Graz 17 dataset. The other two rows show our detection results at equal error rate.

	Number of images		RP AUC			RP EER			
Class	Training	Testing	[21]	BoF	FC	[21]	[17]	BoF	FC
Airplanes	100	400	**0.9310**	0.5402	0.9250	6.8%	7.4%	26.8%	**6.7%**
Cars (rear)	100	400	0.9912	0.8969	**0.9939**	1.8%	2.3%	7.5%	**0.5%**
Faces	100	217	0.9850	0.5129	**0.9906**	2.8%	3.6%	29.9%	**0.9%**
Bikes (side)	90	53	0.6959	0.2795	**0.7100**	32.1%	**28.0%**	49.0%	28.3%
Bottles	54	64	0.9468	0.7810	**0.9514**	9.4%	9.0%	13.9%	**4.9%**
Cows (side)	45	65	0.9975	**1**	**1**	1.5%	**0.0%**	**0.0%**	**0.0%**
Horses (side)	55	96	**0.9680**	0.9069	0.9397	6.3%	8.2%	7.2%	**5.2%**

Table 3. Comparison of detection performance with [21], [17] and BoF on 7 object classes from Graz 17 dataset.

Methods	HOG	4D-ISM	PartISM	Our method
Recall at EER	-	0.68	0.84	0.73
Detection rate	0.71	0.81	0.92	0.84

Table 4. Recall at equal error rate and detection rate of HOG, 4D-ISM, PartISM and our detector.

dataset. We fail to detect the child in the right-bottom image. The size of this object is simply too small for the SIFT descriptor. It would require computing SIFT in 4×4 regions, which does not yield any useful features. Consequently, detecting smaller objects require higher resolution images.

7. Conclusion and Future Work

The proposed Feature Context (FC) with multiple reference points is a simple but efficient way to incorporate spatial arrangement into both image classification and object detection. It is superior to the popular SPM [12] method, due to its accuracy and flexibility. We also introduce a new coding method called Radial Basis Coding (RBC). When combined with RBC and max-pooling, FC can get a very high accuracy using only dense SIFT features on two large-scale datasets. FC is also suitable to describe objects when only the bounding boxes of objects are given in training. Using sliding window strategy, FC can outperform more sophisticated object detectors [21, 17, 20], which is a nontrivial accomplishment. In the future, we would like to combine FC method with the interest region detectors. This would allow us to adjust the placement of the FC reference points to the image content.

Acknowledgement

The work was supported by the NSF under Grants IIS-0812118, BCS-0924164, OIA-1027897, the AFOSR Grant FA9550-09-1-0207, the DOE Award 71498-001-09, the National Natural Science Foundation of China (NSFC) Grants 60873127 and 60903096.

References

[1] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. *Proc. of CVPR*, 2008.

[2] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. *Proc. of CVPR*, 2009.

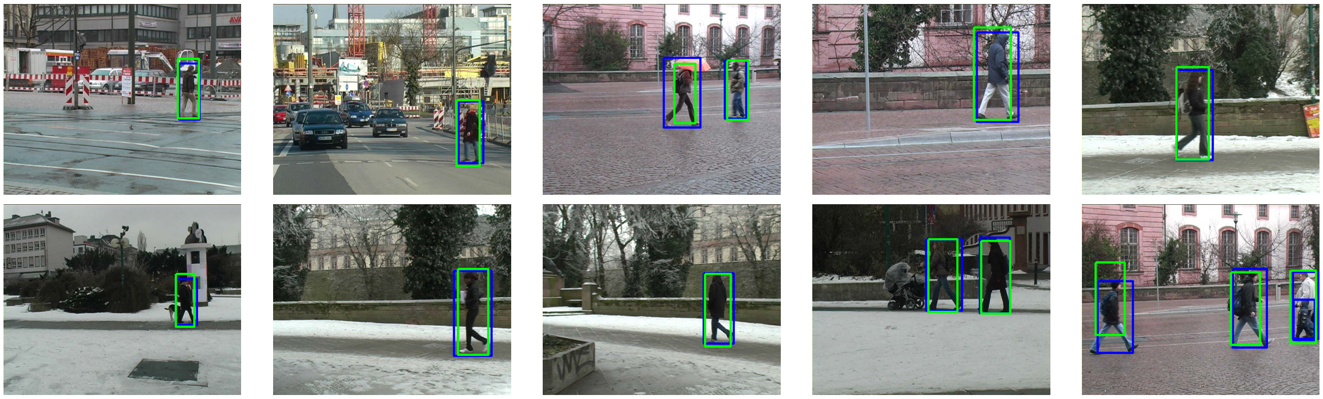


Figure 8. Some of the pedestrian detection results at equal error rate. Ground truth bounding box is plotted in blue, detection result is plotted in green.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *In IEEE Trans. on PAMI*, 2002.

[4] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. *Proc. of CVPR08*, 2008.

[5] A. M. Bronstein and M. M. Bronstein. Spatially-sensitive affine-invariant image descriptors. *ECCV*, 2010.

[6] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. *CVPR*, 2010.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Proc. of CVPR*, 2005.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[10] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *In IEEE CVPR Workshop on Generative-Model Based Vision*, 2004.

[11] K. Grauman and T. Darrell. Pyramid match kernels criminative classification with sets of image features. *ICCV*, 2005.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proc. of CVPR*, 2006.

[13] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. Sift flow: dense correspondence across different scenes. *ECCV*, 2008.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.

[15] M. Marszaek and C. Schmid. Spatial weighting for bag-of-features. *CVPR*, 2006.

[16] A. Oliva and A. Torraba. Modeling the shape of the scene: A holistic representation of the spatial envelop. *IJCV*, 2001.

[17] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. *CVPR*, 2006.

[18] A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 2008.

[19] F. Porikly. Integral histogram: A fast way to extract histograms in cartesian spaces. *CVPR*, 2005.

[20] E. Seemann and B. Schiele. Cross-articulation learning for robust detection of pedestrians. *DAGM*, 2006.

[21] J. Shotton, A. Blake, and R. Cipolla. Multi-scale categorical object recognition using contour fragments. *In IEEE Trans. on PAMI*, 2008.

[22] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. *Proc. of ICCV*, pages 1470–1477, 2003.

[23] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. *CVPR*, 2010.

[24] Z. Tu and X. Bai. Auto-context and its application to high-level vision tasks and 3d brain image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010.

[25] J. C. van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. *ECCV*, 2008.

[26] J. Wang, J. Yang, K. Yu, F. Lv, and Y. G. T. Huang. Locality-constrained linear coding for image classification. *Proc. of CVPR*, 2010.

[27] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. *Proc. of CVPR*, 2009.

[28] J. Yang, K. Yu, and T. Huang. Efficient highly over-complete sparse coding using a mixture model. *ECCV*, 2010.

[29] K. Yu, T. Zhang, and Y. Gong. Nonlinear learning using local coordinate coding. *Proc. of NIPS*, 2009.

[30] H. Zhang, A. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest heighbor classi?cation for visual category recognition. *CVPR*, 2006.

[31] W. Zheng, S. Gong, and T. Xiang. Quantifying contextual information for object detection. *ICCV*, 2009.

[32] Q. Zhu, L. Wang, Y. Wu, and J. Shi. Contour context selection for object detection: A set-to-set contour matching approach. *ECCV*, 2008.