Improving Pedestrian Detection with Selective Gradient Self-Similarity Feature

Si Wu*, Robert Laganière, Pierre Payeur

VIVA Research Lab
School of Electrical Engineering and Computer Science
University of Ottawa, 800 King Edward, Ottawa, ON K1N 6N5, Canada

Abstract

Gradient features play important roles for the problem of pedestrian detection, especially the Histogram of Oriented Gradients (HOG) feature. To improve detection accuracy in terms of feature extraction, HOG has been combined with multiple kinds of low-level features. However, it is still possible to exploit further discriminative information from the classical HOG feature. Inspired by the symmetrical characteristic of pedestrian appearance, we present a novel feature of Gradient Self-Similarity (GSS) in this work. GSS is computed from HOG, and is applied to capturing the patterns of pairwise similarities of local gradient patches. Furthermore, a supervised feature selection approach is employed to remove the non-informative pairs. As a result, the Selective GSS feature (SGSS) is built on a concise subset of pair comparisons. The experimental results demonstrate that significant improvement is achieved by incorporating HOG with GSS/SGSS. In addition, considering that HOG is a prerequisite for GSS/SGSS, it is intuitional to develop a two-level cascade of classifiers for obtaining improved detection performance. Specifically, the first level is a linear SVM with the multiscale HOG features to efficiently remove easy negatives. At the second stage, the already computed HOG features are reused to produce the corresponding GSS/SGSS features, and then the combined features are used to discriminate true positives from candidate image regions. Although simple,

*Corresponding author
Email address: ez.wusi@gmail.com (Si Wu)

this model is competitive with the state-of-the-art methods on the well-known datasets.

Keywords: Pedestrian detection, contour description, self-similarity, feature selection, cascade

---

1. Introduction

Vision based pedestrian detection is a challenging task of great practical interest in the field of computer vision because of variant appearance and shapes of human. A popular paradigm for pedestrian detection is to convert the problem to binary classification. Discriminative methods extract features inside local regions and construct classifiers for detection. A sliding window strategy is often used. However, this problem involves searching a large number of local image regions for a few objects. Cascade classifiers have been applied to cope with this problem of imbalance [1]. In contrast to conventional classifiers designed for a low overall classification error rate, cascade classifiers are required to obtain a very high detection rate and moderate false positive rate within each layer. Another breakthrough was the introduction of gradient based features to pedestrian detection. Inspired by SIFT [2], Dalal and Triggs proposed the Histogram of Oriented Gradient (HOG) features and reported its impressive performance [3]. Currently, HOG is considered to be an unexcelled single feature. There are many works that fused HOG feature with other features to improve its performance [4] [5] [6] [7].

The success of the HOG-based methods indicates that contour is an important clue for pedestrian detection. The existing methods are usually based on partitioning a detection window into a set of subregions, extracting contour features in each subregions, and combining the obtained local features. Although impressive progress has been made in local contour representation, the symmetrical characteristic of pedestrian's appearance was been ignored. As shown in Figure 1, the fragment contours in local regions located in the symmetrical positions on pedestrian's body are similar, on the other hand those located in

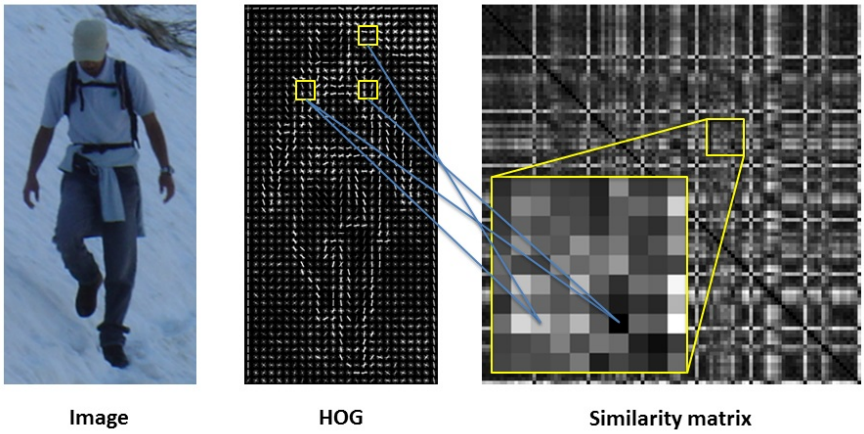


Figure 1: Illustration of pairwise similarity of HOG blocks. For an image example (left), we show the total energy in each orientation of the HOG cells (middle), and the pairwise similarity matrix of the HOG blocks (right). In the matrix, cells with higher similarity are darker. As shown in the zoomed subfigure, the two blocks located in the foreground are similar because of the symmetric characteristic of pedestrian's appearance. On the other hand, the block in the foreground is dissimilar to the one in the background.

the foreground are dissimilar to the one in the background. In addition, we found the fact that both the front and profile of pedestrians look symmetrical in most instances. A few examples are shown in Figure 2. There are apparent symmetry in shape even in different views between the subregions of shoulders, trunk, arms and legs. Therefore, it is possible to measure the similarities among the subregions within the detection window and include the similarities into the representation vector for enhancing contour description.

Improving feature extraction is one of valuable research directions for pedestrian detection as suggested in [8]. Inspired by the fact that pedestrian's appearance is usually symmetrical, we present a new feature based on local gradient similarity in this work. This feature, termed Gradient Self-Similarity (GSS), captures pairwise statistics of spatially localized gradient orientation distribution. Since HOG is one of the most commonly used and effective features for capturing local gradient patterns, we adopt this feature to represent each blocks

Figure 2: Examples of pedestrians with symmetrical appearances in different views.

in a sliding window. The similarities among the blocks are measured by the Euclidean distance in the feature space. We define the GSS feature as a vector composed of the upper triangular elements of the similarity matrix of the HOG features associated with the blocks. However the high dimensionality of GSS may make the computational cost of feature extraction expensive. Considering that some pairs play more important roles than others, we use the Feature Generation Machine (FGM) [9] to perform feature selection. FGM employs a sparse SVM to determine a subset of the feature for classification while retaining the discriminative information. As a result, only a few informative pairs are selected to construct the selective GSS feature (SGSS). GSS/SGSS is a kind of HOG based mid-level features, and achieves additional gains from HOG in terms of exploring the association between fragment features. For the purpose of improving detection efficiency and effectiveness, our proposed framework for pedestrian detection is a short cascade, which has two levels: the first level is

a linear SVM classifier combining with multiscale HOG to efficiently rejects as many of the negative samples while keeping almost all positive samples to the next stage. For the candidate image regions, the HOG features are already computed and reused to produce the corresponding SGSS features. At the second level, we combine the HOG feature and the SGSS feature to discriminate true positives. We also explore the application of the combined feature associated with different classifiers including linear SVM, histogram intersection kernel based SVM (HIKSVM) and AdaBoost. The AdaBoost based cascade achieves the best performance, and is comparable to the state-of-the-art methods on multiple well-known datasets.

The main contributions of this work are two-folder: first, according to our observation on the symmetrical characteristics of pedestrian's appearance, we develop the SGSS feature as a mid-level feature capturing the patterns of similarities among local gradient distributions to significantly improve pedestrian detection rate. Second, considering that our SGSS feature is computed from HOG, we design a two-level cascade for pedestrian detection, in which the HOG feature computed on the first level is reused to construct the GSS feature at the second stage. Our method is therefore based on the computation of a single low-level feature (the HOG). This is an interesting simplification considering that feature extraction is often a computationally costly step in classification approaches. Moreover, we show that the proposed approach provides competitive results. The remainder of this paper is organized as follows. In Section 2, we discuss relevant works on feature extraction and discriminative methods for the pedestrian detection problem. In Section 3, we provide details on the proposed GSS feature and the corresponding feature selection approach. In Section 4, we introduce our cascade of classifiers. In Section 5, we provide the implementation details of the proposed model. In Section 6, we present experimental results based on the proposed approach, and the comparison results with existing methods are also reported. Finally, the conclusion of this paper is presented in Section 7.

2. Related Work

In the past decade, great progress in the research of pedestrian detection has been made through the investigation of different approaches for feature extraction, classification, and articulation handling. The surveys [10] [8] provide comprehensive introductions on the existing pedestrian detection approaches. For feature extraction, Haar wavelet feature was used in the early work of pedestrian detection [11]. In contrast, HOG [3] is a popular feature used in the modern pedestrian detectors. This feature collects gradient information in local cells into histograms using normalizing overlapping blocks. Local normalization makes this representation robust to small pose variations and changes in illumination. Although there is no single feature outperforming HOG, multiple kinds of features have been reported to complement HOG, such as the motion descriptor based on Histogram of Optic Flow (HOF) [4], the texture descriptors based on Local Binary Patterns (LBP) [7] and center symmetric local trinary patterns (variants of LBP) [12], and the Color Self-Similarity (CSS) feature [6]. To combine multiple kinds of low level pixel-wise features, Enzweiler and Gavrila [13] proposed a multilevel mixture-of-experts model built on HOG and LBP features computed from intensity, depth and dense flow data. Dollár et al. [14] proposed an uniform framework for integrating grayscale, LUV color, and gradient magnitude quantized by orientation. A near real-time version of this method was provided in [15]. Based on HOG, a number of high-level features were developed, such as the global pose invariant descriptor [16]. Shape is also a commonly used cue for object detection [17] [18] [19] [20] [21]. In [18], the shape descriptors (shapelets) were learned from gradients in local patches, and combined by boosting to build an overall detector. Another way to represent mid-level edge features is based on contour. Lim et al. [21] clustered patches of hand drawn contours to generate sketch tokens to capture local edge structure. Combining with other multiple image channels, the representation of per-pixel token labelings is utilized as a feature for a boosted detector. Another dictionary based feature is to use sparse coding to construct the histogram of per-pixel

sparse codes for local representation in [20]. The dictionaries are unsupervised learned by K-SVD. Also using an unsupervised technique to learn features from data, a convolutional network model is used to learn multi-stage shape features in [19]. In this work, we are inspired by the symmetrical characteristic of pedestrian's appearance, and propose the GSS feature to capture the patterns of the similarities of fragment contours in local regions. HOG is used as a source of low-level features from which our GSS feature is computed. Different from CSS, we here explore the pairwise statistics of spatially localized gradient distributions instead of color. Furthermore, a supervised feature selection method is used to remove the non-informative components in GSS, and produce the SGSS feature. To the best of our knowledge, SGSS has not yet been used as a feature for pedestrian detection.

The most commonly used discriminative approaches to the pedestrian detection problem are various boosting classifiers [14] [22] [23] and SVM classifiers [3] [24] [25] which are usually in the form of cascade. For instance, in the work of Viola and Jones [26], the integral image concept is used for fast feature computation, the AdaBoost algorithm is used for automatic feature selection, and a cascade structure is used for efficient detection. In [27], boosted decision trees were applied to a two-level cascade architecture. Felzenszwalb et al. [24] proposed a deformable part model (DPM) in which unknown part positions was modeled in a latent SVM. In another work [28], based on DPM, an ordering of the model's parts was used to define a hierarchy of the models to gain speed which is analogous to a classical cascade. In addition, the histogram intersection kernel has been shown to be more effective than the Euclidean distance for many classification problems when using histogram features. However, for non-linear SVM classifiers, the runtime complexity is high. Maji et al. [25] proposed an approximated intersection kernel SVM which provides great speedup such that the nonlinear SVM can be used in sliding window detection. Recently, deep models have begun to be applied to pedestrian detection [29] [30] [31]. Different from the classical cascaded classifiers trained sequentially without optimization, Zeng et al. [31] proposed a multi-stage contextual deep model which jointly

trains the classifiers at each stage through back-propagation.

Algorithmically, we use in this work a two-level cascade in which the first level is a linear SVM, and the second level is a linear SVM, HIKSVM or AdaBoost. This design is justified by the fact that the first level is efficient and can quickly remove most false positives, and the already computed HOG features can be reused to generate our SGSS features for further discrimination on the second level. We study here the effectiveness of these classifiers when used in conjunction with the proposed SGSS feature.

3. Gradient Self-Similarity

The concept of HOG is to represent objects by dense grids of gradient histograms that characterize an object's contour and its spatial information to some extent. The detection window is usually divided into cells represented by gradient histograms, and each 2×2 neighboring cells constitute a block. The L_2 normalization is performed on each block, which makes the HOG feature robust to illumination changes. Since HOG has exhibited excellent performance in representing local gradient distributions, we employ HOG to encode the local subregions (blocks) in a detection window, and measure the similarities of these subregions by computing the distances in the feature space. To present the patterns of similarities between spatially located blocks, we begin with the introduction of the GSS feature in Section 3.1. In Section 3.2, we present an effective approach of feature selection to remove the redundant components, which provide no more information than the selected subset of components in GSS.

3.1. GSS Feature

Let $H = (H_1, H_2, \ldots, H_m)$ be the HOG feature in a detection window, where H_i, $i = 1, 2, \ldots, m$, denote the features of the blocks. Since each block consists of four cells, let $H_i = (H_{i1}, H_{i2}, H_{i3}, H_{i4})$ be the concatenated histograms of the i-th block. We measure the similarities of fragment contours through the

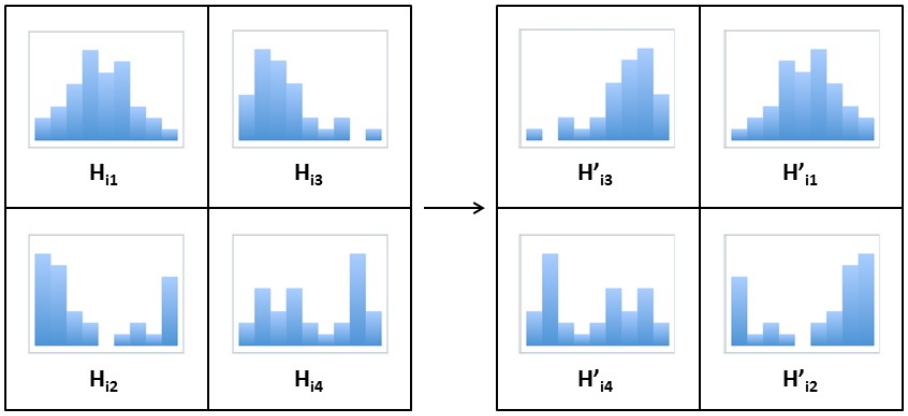


Figure 3: An example of the horizontal flip operation for HOG blocks.

distances of HOG blocks in the feature space. Since pedestrians are vertically symmetrical, we consider that the blocks located on the symmetrical positions of pedestrian's body, such as the left and right shoulders, should be similar, but the distances between them may be very large because of the complementarity of their gradient orientations. To solve this problem, one feasible way is to horizontally flip the HOG blocks as shown in Figure 3. Let H'_i denote the flipped vector of H_i. We define the distance matrix as follows:

$$\mathcal{D}_{i,j} = \min(d(H_i, H_j), d(H'_i, H_j)) \tag{1}$$

where d denotes the distance metric. Eq. (1) indicates that the similarity between HOG blocks is determined by the minimum distance between the flipped and non-flipped cases. There are many possibilities to define d. We tested a number of widely used distance functions including the L_2-norm, χ^2-distance, dot product, and cosine of the angle between dominant gradient orientations in the experiments. We use the L_2-norm as it yields the best performances. The corresponding similarity matrix is computed by applying the following transform which guarantees that the similarity values are within the range $(0, 1]$,

$$\mathcal{S}_{i,j} = \frac{1}{1 + \left(\frac{\mathcal{D}_{i,j} - \mathcal{D}_{min}}{\mathcal{D}_{max} - \mathcal{D}_{i,j}}\right)^2} \tag{2}$$

where $\mathcal{D}_{min}$ and $\mathcal{D}_{max}$ denote the minimum value and the maximum value respectively (for the cases of the distance defined by dot product and cosine, the formula of similarity computation in Eq. (2) is slightly adjusted by inverting the fraction of the denominator because the similarities between the blocks are proportional to the corresponding distances). Since $\mathcal{S}$ is a symmetric matrix, the GSS feature is defined as follows:

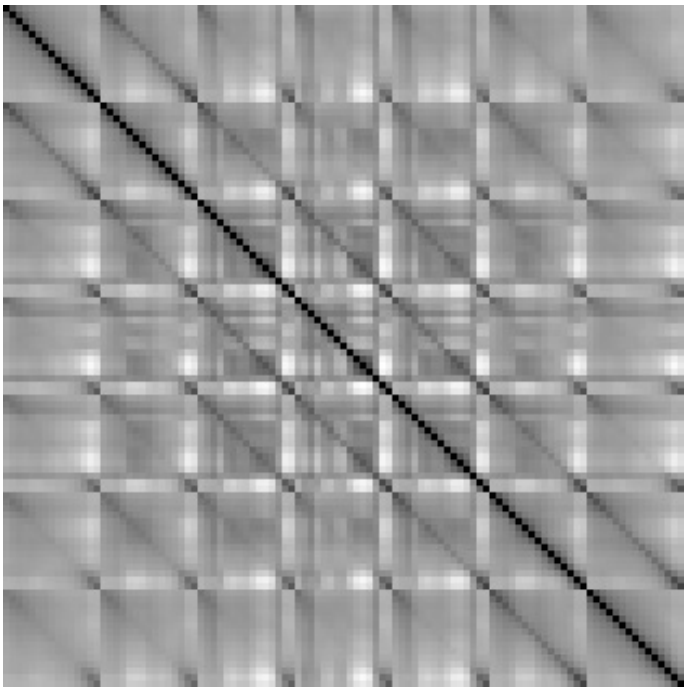
$$\begin{aligned} &F_{GSS} = (g_1, g_2, \ldots, g_n), \\ &g_k \in \mathcal{S}_{upper} = \{\mathcal{S}_{i,j} | i < j\}, k = 1, 2, \ldots, n, \end{aligned} \tag{3}$$

where $\mathcal{S}_{upper}$ is the set of upper triangular elements of the similarity matrix, and this feature vector has $n = \frac{m \times (m-1)}{2}$ dimensions. We exhibit the capability of GSS in capturing pairwise similarity patterns of human appearance by means of an example in Figure 4. We compute the average similarity matrix for positive training samples. This matrix is shown in Figure 4(a). Each row indicates the similarities between a HOG block and all the others. We also show several representative rows laid out at the corresponding spatial locations of these blocks. It is noted that there exist pedestrian structures in the sub-images in Figure 4(b).

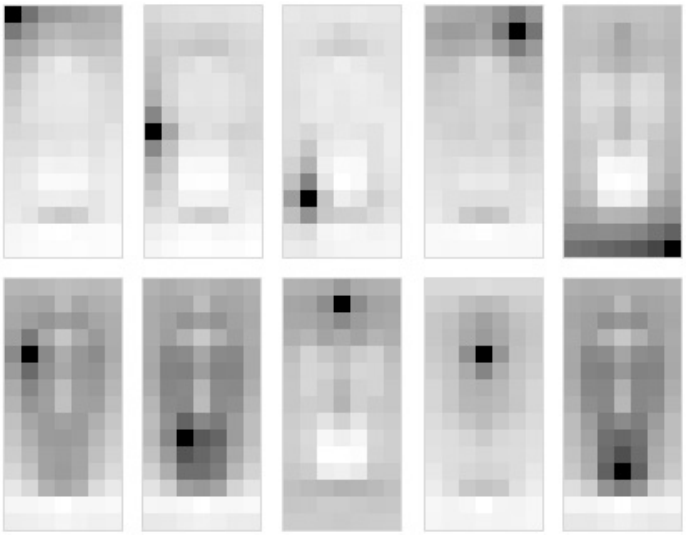
Finally, the GSS feature undergoes two normalization steps. The first step is to perform power normalization through the following operation applied on each component independently:

$$f(z) = |z|^{\alpha}, \tag{4}$$

with $\alpha > 0$. We empirically observed that this transform indeed improves the discrimination capability of the GSS feature. The interpretation of this observation is that the transform in Eq. (4) is a kind of nonlinear mappings which re-assigns the values of the elements in the GSS feature such that the transformed feature has a higher discriminating power. We found experimentally that setting $\alpha = 2$ consistently leads to near-optimal results. The power normalized GSS feature is subsequently L_2-normalized by $v := \frac{v}{\|v\|_2}$ in the second step.



(a)



(b)

Figure 4: Gradient self-similarity as a mid-level feature captures pedestrian structures. (a) The average similarity matrix of positive samples. (b) The representatives of the meaningful rows of the similarity matrix visualized by spatial layout.

3.2. Selective GSS Feature

For cases where there are many features and comparatively few samples, feature selection techniques are often used. They bring the benefit of shortening training times and enhancing generalization by reducing overfitting. High dimensional vectors may indeed result in great challenges for computation and training, and in the case of our GSS feature, it is clear that the similarities of some block pairs may be non-informative. We therefore opt for FGM as a tool to perform feature selection such that the trained classifier will be made of simplified decision rules for faster prediction. In contrast to the Principal Component Analysis (PCA) [32] that transforms the data into a set of linearly uncorrelated variables in an unsupervised way, FGM is a supervised method which reduces the dimensionality of GSS, while preserving discriminative information. Although the Partial Least Square (PLS) analysis [33] is a supervised dimensionality reduction technique and has been shown to be effective for the pedestrian detection problem [34], full features still need to be computed before PLS projection which maintains the complexity of the feature extraction process.

Given a set of labeled samples $\{x_i, y_i\}$, $i = 1, 2, \ldots, n$, where x_i is the GSS feature vector and y_i is the label, FGM aims at finding a sparse solution with respect to the input features to a linear SVM can be learnt by minimizing the following structural risk functional:

$$\begin{aligned} &\min_{t \in T} \min_{\omega, \xi, \rho} \quad \left(\frac{1}{2}\|\omega\|^2 + \frac{\lambda}{2} \sum_{i=1}^{n} \xi_i^2 - \rho \right) \\ &s.t. \quad y_i \omega'(x_i \odot t) \geq \rho - \xi_i, \end{aligned} \tag{5}$$

where ω is the weight vector, the feature selection vector $t = (t_1, t_2, \ldots, t_m) \in T$, $T = \{t | t_j \in \{0, 1\}, j = 1, 2, \ldots, m\}$ which controls the sparsity of the SVM decision hyperplane: $\omega'(x \odot t)$, and λ is the regularization parameter that balances the model complexity and the fitness of the decision hyperplane. Eq. (5) is a mixed integer programming problem. After convex relaxation, Tan et al. [9] proposed an efficient cutting plane algorithm to find a sparse feature solution.

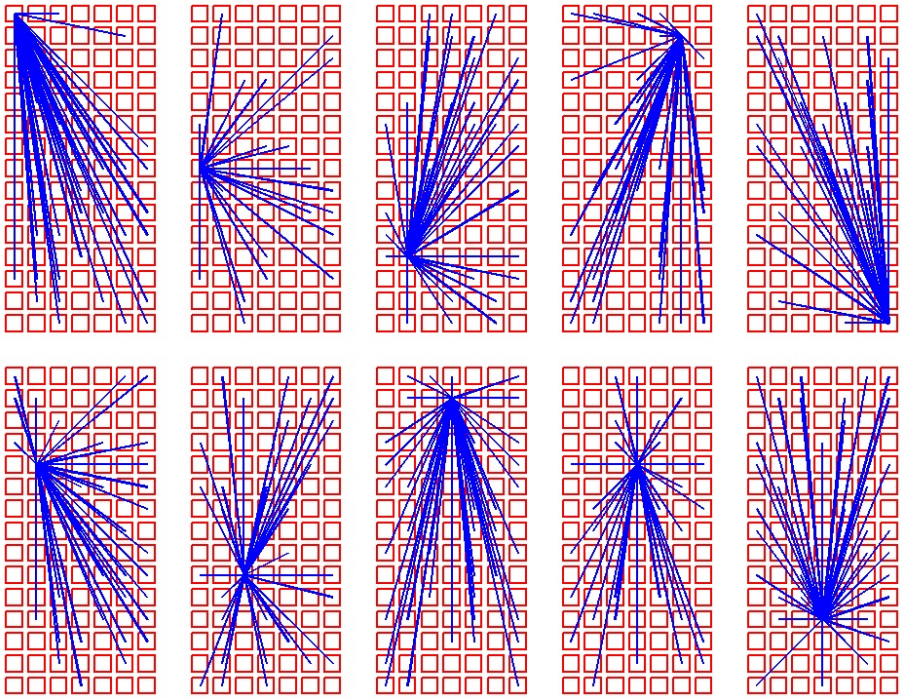


Figure 5: The selected pairs associated with the anchor blocks shown in Figure 4(b) after using FGM based feature selection. The squares denote the blocks in the detection window, and the lines denote the selected pairs.

Once feature selection performed on the training data by applying FGM, the feature subset composed of the selected elements in the GSS feature is concise, while maintaining the discriminating power almost as high as the original GSS feature as it will be shown in Section 6. Thus we define the selective gradient self-similarity feature (SGSS) as the selection of FGM:

$$\begin{aligned} &F_{SGSS} = (g_{i_1}, g_{i_2}, \ldots, g_{i_K}), \\ &s.t. \quad t_{i_k} = 1, i_k \in \{1, 2, \ldots, n\}, k = 1, 2, \ldots, K. \end{aligned} \tag{6}$$

An example of selected pairs of FGM is shown in Figure 5. It is noted that most selected pairs involve the blocks, which are located near the contours of the pedestrian structures for the anchor blocks shown in Figure 4(b). This fact indicates that the contours contain discriminating information, which is consistent with human perception. In contrast to the HOG feature representing the contour information piece by piece, the SGSS feature is capable to explore the association patterns of pieces of contour, which can be seen as a mid-level features on top of HOG blocks. For this reason, the SGSS feature is considered to be, to a certain extent, complementary to the HOG feature.

4. Cascade

Another major component for pedestrian detection systems is the classifier. We therefore explore the applicability of the developed feature combined with the commonly used classifiers including linear SVM, HIKSVM and AdaBoost. As explained before, the proposed GSS feature is computed from the HOG feature. In order to obtain excellent detection performance while keeping a low computational cost, we introduce a framework composed of two-level cascade of classifiers. On the first level, a linear SVM is trained in the HOG feature space. The goal of this level to reject as many negatives as possible, while still passing almost all of the positives to the next level. The first level is computationally efficient. The second level makes the final decisions for the candidates including positives and difficult negatives accepted by the first level. Since the HOG features of the candidates have already been computed on the

first level, it is straightforward to compute the corresponding GSS features to build more discriminative descriptors combining HOG and GSS. Although GSS feature is high-dimensional and the computation cost is expensive, the number of the candidates is usually small. In addition, since we performed feature selection using FGM, the obtained SGSS feature is composed of a small number of informative components. These ones are combined with the HOG feature to train the classifier of the second level to make the final decision. We here apply three different classifiers to the second level of our short cascade.

4.1. Linear SVM

For simplicity, we propose to use a linear SVM model as a baseline classifier at the second level of the cascade. A linear SVM classifier learns the hyperplane that optimally separates pedestrians from background, and usually provides good performance in comparison to other linear classifiers. The combined representation vectors of the HOG feature and the corresponding SGSS feature are then fed to the linear SVM for efficient classification.

4.2. Approximated Intersection Kernel SVM

Kernelized SVMs are typically used for machine learning based discriminant. Replacing the linear SVM with a nonlinear kernel usually improves performance at the cost of much higher run times because the application of kernerlized SVMs to classification requires computing the kernel distance between the input vector and each of the support vectors. As a result, kernelized SVMs are rarely used for detection task because of their high computational load. To make this computation more efficient, we employ an approximated intersection kernel SVM [25] on the second level of the cascade which has the benefit of being independent to the number of support vectors.

For a trained HIKSVM, the decision function is given as follows:

$$\begin{aligned} h(x) &= \sum_{r=1}^{R} \alpha_r y_r k(x, x_r) + b \\ &= \sum_{r=1}^{R} \alpha_r y_r \left(\sum_{i=1}^{m} \min(x(i), x_r(i)) \right) + b, \end{aligned} \tag{7}$$

where $k(\cdot, \cdot)$ is the kernel function, and x_r, $r = 1, 2, \ldots, R$, are support vectors. Exchanging the summations in Eq. (7), we obtain

$$\begin{aligned} h(x) &= \sum_{i=1}^{m} \left(\sum_{r=1}^{R} \alpha_r y_r \min(x(i), x_r(i)) \right) + b \\ &= \sum_{i=1}^{m} \left(\sum_{1 \le r \le p} \bar{\alpha}_r^i \bar{y}_r^i \bar{x}_r^i + x(i) \sum_{p < r \le m} \bar{\alpha}_r^i \bar{y}_r^i \right) + b \\ &= \sum_{i=1}^{m} h_i(x(i)) + b, \end{aligned} \tag{8}$$

where $\bar{x}_r^i$ denotes the increasingly sorted values of x_r in the i-th dimension, and $\bar{\alpha}_r^i$ and $\bar{y}_r^i$ are the corresponding weight and label. After computing $h_i(\bar{x}_r)$, $h_i(x(i))$ can be estimated by first finding p and then linearly interpolating between $h_i(\bar{x}_p)$ and $h_i(\bar{x}_{p+1})$. In practice, the input data is quantized in each dimension, and the piecewise constant approximation is used to compute h_i. As a result, only a lookup table is required for prediction. In our case, the SGSS feature can be quantized before training the intersection kernel model. The discrete SGSS feature is then made more robust to changes in gradients. The quantization distortion of the SGSS feature does not cause loss in classification accuracy because of the piecewise constant approximation of h_i.

4.3. AdaBoost

AdaBoost offers another fast approach to learning over high dimensional data. In contrast to SVMs, boosting methods minimize the classification error on the training data by combining weak classifiers iteratively. Choosing the appropriate weak classifier is important to produce a strong classifier. We use

the regression stumps as our weak classifiers, which are very simple and computationally inexpensive because they classify input samples according to a single dimension of the combined feature vector of HOG and SGSS. We use the Gentle AdaBoost algorithm [35] to train the model on the second level of our cascade, which is very similar to other AdaBoost algorithms. During the training phase, the same weight is initially assigned to each sample. A weak classifier is then trained on the weighted training set. The misclassified samples are assigned to higher weights, which enable the training process to more focus on a subset of misclassified data. However, classic AdaBoost algorithm is sensitive to noisy data and outliers. Gentle AdaBoost fits a regression function by minimizing a weighted least-squares loss, and modifies the weighting method to put less weight on outlier samples, which leads to better generalization performance. When the number of individual regression stumps is met, the output of the trained weak classifiers is combined into a weighted sum, which is defined as the final output of the boosted classifier. The runtime of this model is linear in the number of regression stumps.

5. Implementation

Since our objective is to explore the applicability of the SGSS feature, we here use a simple two-level cascade for the task of pedestrian detection. The first level is the commonly used HOG and linear SVM combo. For the candidates passing the first level, the already computed HOG features are used to compute the corresponding SGSS features. The HOG feature and the SGSS feature are then concatenated and fed to the classifiers (linear SVM, HIKSVM, or AdaBoost) on the second level. We will first present the details on the parameter setting and the training procedure for this model, and subsequently introduce the postprocessing technique in the following subsections.

5.1. Parameter Setting

Our classification model scans a 64×128 detection window with a stride of 8×8 across the image, running a pretrained classifier on the descriptors

extracted from each resulting image window. For multiscale detection, we use a scale stride of 1.05. The widely used version of the HOG feature consists of 7×15 blocks of histogram features with 36 dimensions per block. Thus there are 5460 block pairs and the corresponding GSS feature is a 5460 dimensional vector of similarities. For feature selection on the GSS feature, the regularization parameter λ in Eq. (5) controls the tradeoff between the model complexity and the fitness. The greater the value of λ is, the higher the dimension of the SGSS feature is. In the experiment, the value of λ is empirically set to 10, and about 30% elements of the GSS feature are selected. In the cascade model, the threshold of the first level is set to -2.5 that pass about 97% positives while rejecting about 98% negatives on the INRIA pedestrian dataset. For the second level, we use the SVM tool LIBSVM [36] to train a linear SVM and a HIKSVM setting both the values of the parameter C balancing the training error and the rigid margin to 0.1. In addition, we also trained a boosted classifier with 500 regression stumps.

5.2. Training Procedure

We train the classifiers on both the two levels of the cascade on the INRIA dataset. Generally, for machine learning algorithms, more training data means better performance. However, for the scanning window classifiers, there are too many negative samples to fit into memory at a single time, and another relevant issue is that training becomes time consuming in the case. As a result, the bootstrapping process is crucial to obtain best performance while keeping the memory requirements manageable. We train the classifiers involved in the cascade with initial subsets of negative samples. For the linear SVM on the first level, 2 negative samples are selected at random for each negative training image. For the classifiers on the second level of the cascade, 2 negative samples having responses from the first level greater than a preset threshold are selected randomly. Next, the negative samples that are incorrectly classified by the initial classifiers are extracted. The training procedure is repeated by including a subset of these difficult negatives into the training set. In our case, we limit

the number of hard negative samples added to the training set to 2 for each image. This process is repeated until the change in the miss rates between two iterations is smaller than a prespecified threshold.

5.3. Postprocessing

In the test phase, the proposed cascade is performed on each test image in all positions and scale with the window stride and the scale factor specified above. Each object is usually detected in multiple overlapping bounding boxes. To eliminate repeated detections, non-maximal suppression is used to merge the multiscale nearby predictions having the final classifier responses greater than a certain threshold. Specifically, we sort the surviving windows by response, then iteratively take the highest one and remove the less confident windows that sufficiently overlap it. In the experiment, the overlap threshold is set to 0.65.

The PASCAL evaluation criterion is usually used to assess detection performance. A detection is considered to be a true positive if the detected bounding box overlaps more than 50% with the ground truth bounding box, where the overlap is measure as the ratio of the intersection area to the union area. For the test images, the ground truth bounding boxes are tight in both height and width of pedestrian. However, the positive training samples are normalized only according to the height such that the change in the foreground area is significant, especially for the case of profile. As a result, it may occur that a detected bounding box well fits a pedestrian in height but fails to match the ground truth because of the width. To solve this problem, we roughly divide the positives into two groups according to the width of pedestrian. For each group, an appropriate cropping solution is made. Once detection is obtained, we compare the detection with the prototypes of the two groups in the HOG_SGSS feature space, and adopt the cropping solution of the closed group.

6. Experiments and Discussion

In this section, we evaluate our GSS/SGSS feature and the proposed cascades on well-known datasets. All detection rates are compared using False-Positive-Per-Image (FPPI) curves. First, to confirm the improvement on detection accuracy by introducing the GSS feature, we employ a linear SVM, and compare the detection performance with HOG and the combinations of HOG and GSS based on various distance metrics. Second, to show the effectiveness of feature selection, we study the involved parameters, and compare the performance of using SGSS and GSS. We also evaluate the cascades associating SGSS with different classifiers. Finally, we compare the AdaBoost based cascade using multiscale HOG and the corresponding SGSS with state-of-the-art approaches.

6.1. Dataset

The test dataset includes the INRIA [3], ETH [37], TUD-Brussels [5], and Caltech [38] pedestrian datasets. Although the scale of the INRIA dataset is relative small, it is popular for evaluating the methods of pedestrian detection due to variable appearance, wide variety of articulated poses, complex backgrounds and illumination changes. The training set includes 2416 images of mirrored pedestrian samples and 1218 pedestrian-free images, and the test set includes 288 images with 589 annotated pedestrians and 453 pedestrian-free images. Only the positive testing images are used for evaluation. The ETH and TUD-Brussels datasets are captured in urban areas using a camera mounted to a stroller or vehicle. In the TUD-Brussels dataset, there are 508 image pairs with overall 1326 annotated pedestrians. In addition, the ETH dataset consists of three test sets including 999, 450 and 354 consecutive frames with 5193, 2359 and 1828 annotated pedestrians respectively. The Caltech dataset is the most challenging and the largest by far. It contains 11 subsets of videos, the first 6 for training and the last 5 for test. There are total 350k pedestrian bounding boxes around 2300 unique pedestrians annotated. The evaluation on this dataset is performed using every 30-th frame.

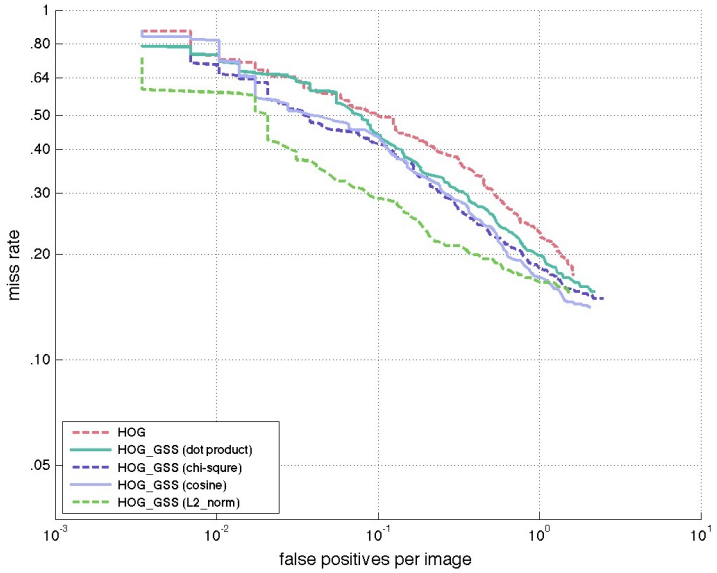


Figure 6: Comparison of the different types of GSS features on the INRIA dataset.

6.2. Distance Metric

The definition of the distance metric in Eq. (1) is the key to construct discriminative GSS features. The first experiment is to explore several possibilities for defining the function d. Having obtained the HOG feature of a sliding window, we here test the common distance functions including the L_2-norm, dot product, χ^2-distance, and cosine function (for each pair of HOG blocks, the value of d is defined as the mean of the cosine values of the angles between the dominant gradient orientations of the corresponding cells). We evaluate the different combinations of the HOG feature and these types of GSS features by training linear SVMs and testing them on the INRIA dataset. The results shown in Figure 6 demonstrate that the addition of our GSS feature gives a significant boost to detection accuracy, which indicates that these GSS features are complementary to HOG indeed. Some representative results shown in Figure 7 more specifically demonstrate the enhanced discriminability in the cases of occlusion and deformation. Compared with the other three types of distance

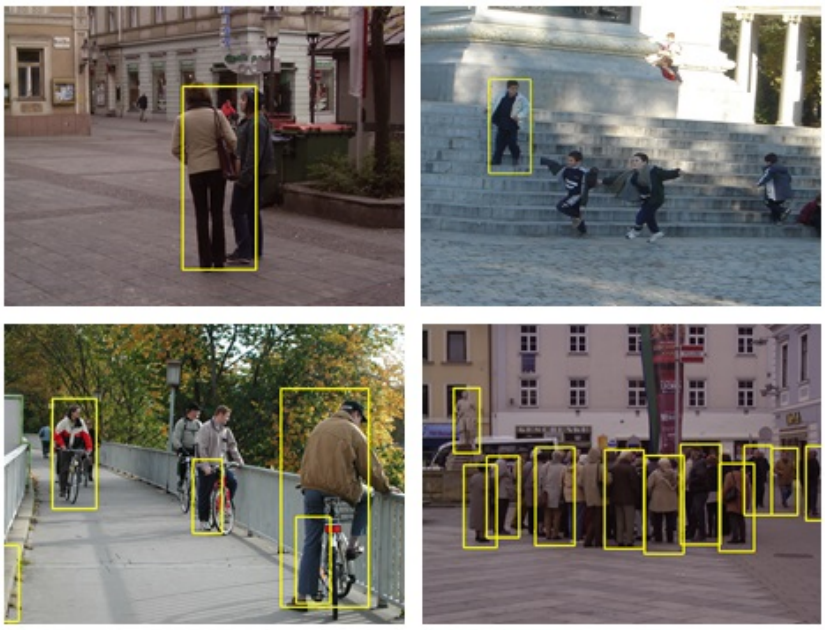
functions, the L_2-norm is the best. HOG_GSS (L_2-norm) is consistently better than HOG, and improves by 0.2 the detection rate at 10^{-1} FPPI. In the subsequent experiments, we will use the L_2-norm based GSS feature.

6.3. Feature Selection

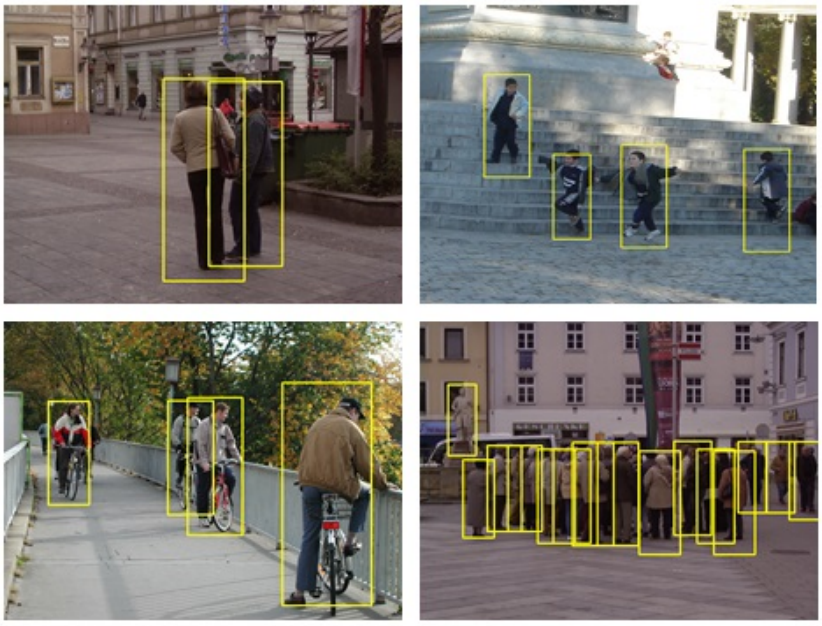
To reduce the computation cost of the GSS feature while keeping discriminative information, we apply FGM to determine a concise subset of GSS components as the SGSS feature. Since FGM is supervised, it is guaranteed that the obtained feature will be discriminative. The parameter λ controls the dimension of SGSS. We here test different values of λ: 0.1, 1, 10 and 100. In each case, we combine the HOG feature and the obtained SGSS feature to retrain a linear SVM, and the performance is shown in Figure 8. With the value of λ increasing, the dimension of SGSS becomes higher, and the corresponding performance is closer to that of GSS. Even in the case of $\lambda = 0.1$, the SGSS feature of 426 dimensions improves the detection rate by 0.15 at 10^{-1} FPPI on the INRIA dataset. The change in performance is not significant when $\lambda = 10$ and 100. In the following experiments, we set the value of λ to 10 because the dimensionality of the SGSS feature is less than half that of the GSS feature (5460) with only a minor loss in detection rate.

6.4. Cascade Evaluation

In this experiment, we evaluate the detection performance of the cascades introduced in Section 4 on the INRIA dataset. To fully explore the discrimination capability of the SGSS feature, we use the multiscale HOG feature, which includes 3 different window size: 64×128, 32×64 and 16×32. Specifically, for a 64×128 sliding window, we resize the image region to 32×64 and 16×32, and compute the corresponding HOG features. To compute the corresponding GSS feature, we take the similarities of the blocks in different scales into consideration. As a result, there are total 129 HOG blocks, and the corresponding GSS feature has 8256 dimensions. After feature selection, the SGSS feature only has 2162 dimensions. A common linear SVM trained with the multiscale



(a)



(b)

Figure 7: Some representative results of (a) HOG and (b) HOG_GSS (L_2-norm).

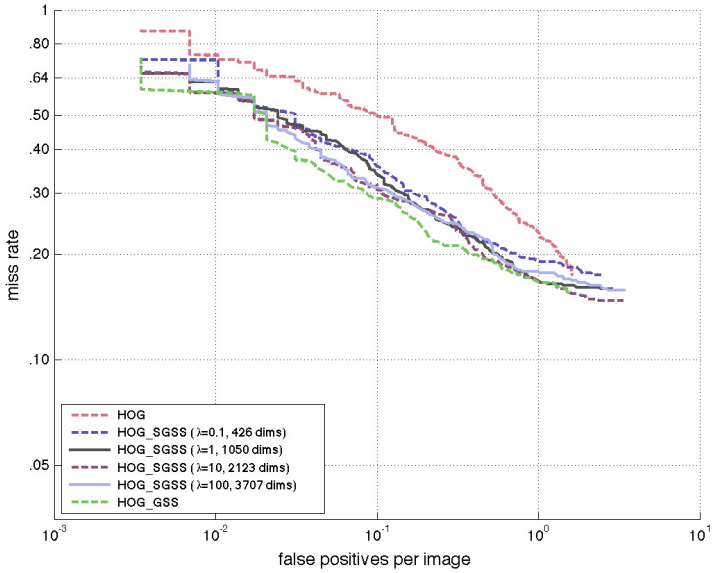


Figure 8: Comparison of the GSS feature and the SGSS features on the INRIA dataset.

HOG feature is adopted on the first level of the three level-2 classifiers to be evaluated. On this second level, the feature composed of the multiscale HOG feature and the SGSS feature are then fed to a linear SVM, HIKSVM or AdaBoost classifiers. The results shown in Figure 9 demonstrate that the two-level cascade significantly outperform the linear SVM associated with a single scale HOG. This is mainly due to the multiscale representation and our complementary SGSS feature. In addition, both HIKSVM and AdaBoost used on the second levels of the cascade are better than linear SVM, and the performance of AdaBoost is the best.

6.5. Comparison

We finally evaluate our SGSS feature based classifier on the INRIA, ETH, TUD-Brussels and Caltech pedestrian datasets, and compare the proposed approach with the existing methods. We here employ the AdaBoost-based cascade using the multiscale HOG and the corresponding SGSS as the one used in the

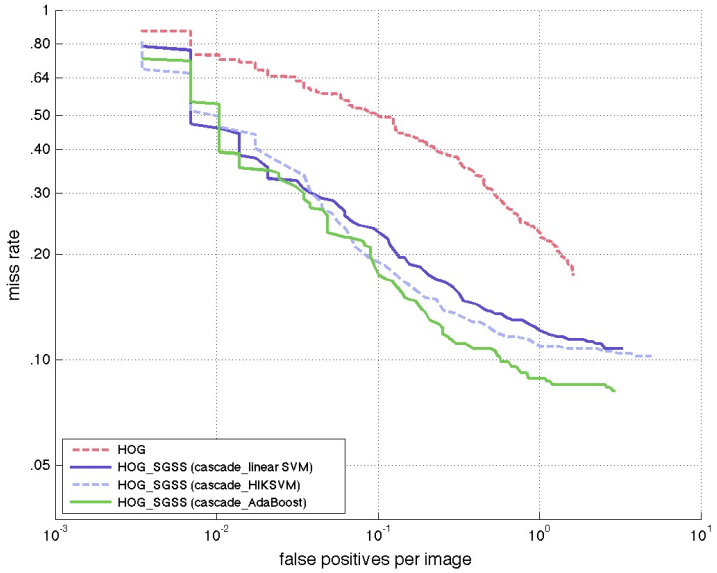


Figure 9: Comparison of linear SVM, HIKSVM and AdaBoost as the 2nd level of the cascade on the INRIA dataset.

above experiment. The results are shown in Figures 10-13; note that for all the experiments, our classifer has been trained on the INRIA dataset. Our detector significantly outperforms the baseline detector (HOG) by about 0.32, 0.15, 0.23 and 0.17 in a detection rate of 10^{-1} FPPI on the four datasets respectively. Although the proposed model is simple, our detector is close to DPM (LatSvm-V2) as the best detector purely based on the HOG feature on the INRIA dataset, and exhibits better performance on the other three datasets. The other state-of-the-art methods consider more feature channels such as color and gradient magnitude. Despite this fact, our approach provides very competitive results, especially on the ETH, TUD-Brussels and Caltech datasets. The relative ordering of the proposed method is roughly preserved across different datasets, which indicates that our SGSS feature is robust to imaging condition changes.

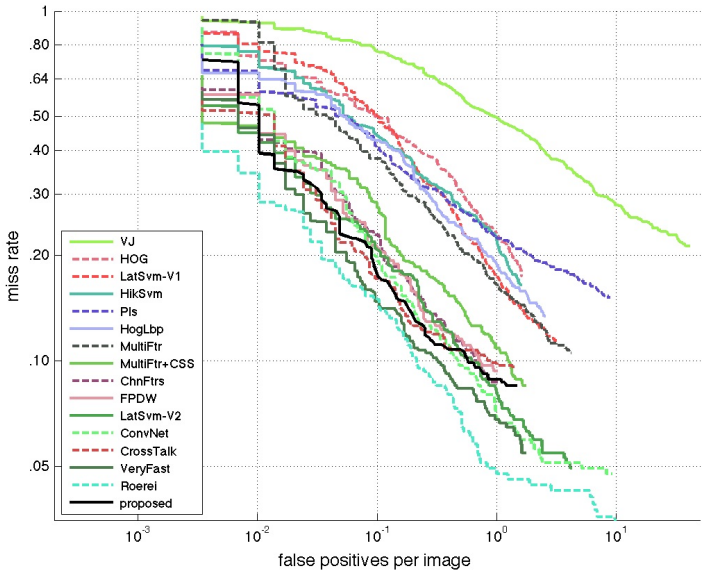


Figure 10: Comparison of different methods on the INRIA dataset.

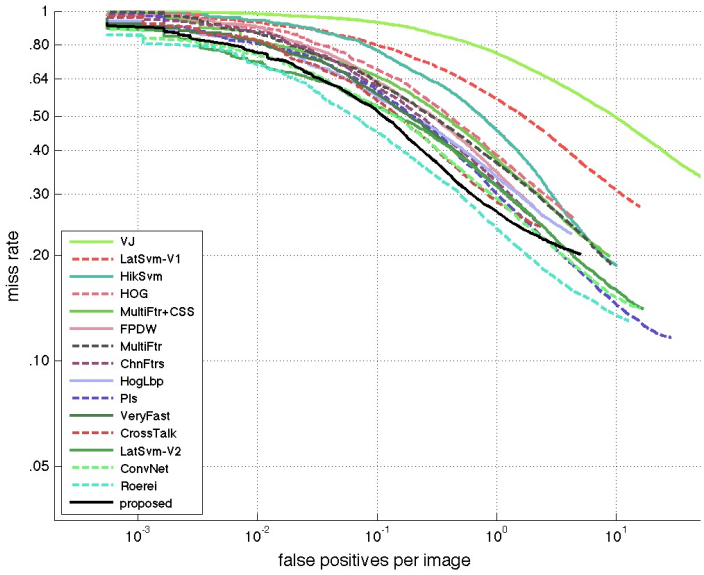


Figure 11: Comparison of different methods on the ETH dataset.

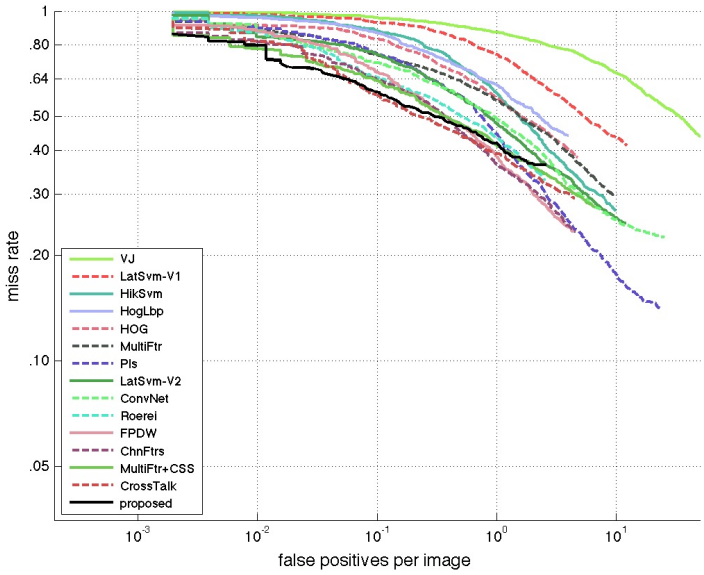


Figure 12: Comparison of different methods on the TUD-Brussels dataset.

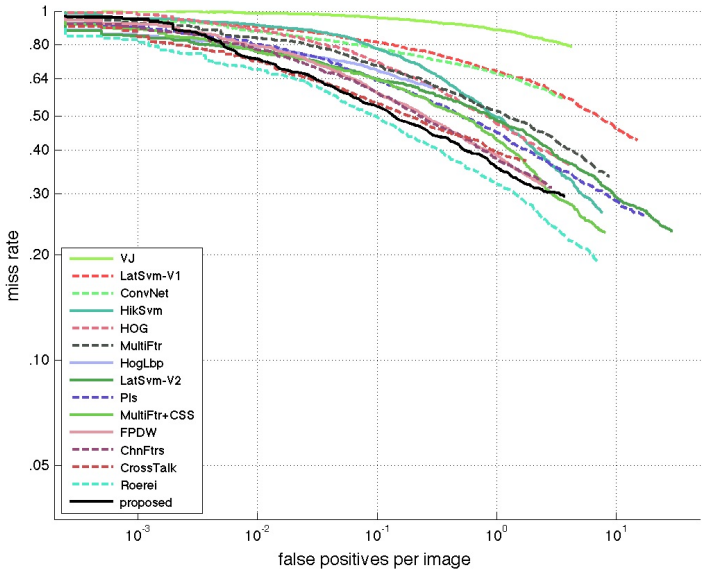


Figure 13: Comparison of different methods on the Caltech (reasonable) dataset.

7. Conclusion

In this paper, we presented a mid-level feature: termed GSS, which captures the patterns of pairwise similarities of local gradient distributions. To obtain a concise subset of elements in the GSS feature without losing discriminative information, we employed a sparse SVM to generate the SGSS feature. Considering that our SGSS feature is derived from HOG, a two-level cascade for pedestrian detection was designed to use a linear SVM with HOG to filter candidate image regions. The second layer of this cascade reused the computed HOG to construct the corresponding SGSS and make the final decisions. Instead of computing other low-level features, we use SGSS to mine further discriminative information from the already computed HOG. The results of the experiments demonstrate that the GSS/SGSS feature is capable of improving the detection performance, and the resulting two-level cascade is competitive with other top-performing approaches.

Note that the proposed GSS/SGSS feature is built on simple regular grids and composed of comparisons of a number of HOG block pairs in the sliding window. This leads to the question on how to design an ideal sampling pattern, which would work better than regular grids. Inspired by the work of Alahi et al. [39] mimicking the human visual system, we now would like to design a center-symmetric sampling pattern which has higher density of points near the center with a variation of the Gaussian kernel size in order to gain performance in our future work.

References

[1] S. Paisitkriangkrai, C. Shen, J. Zhang, Fast pedestrian detection using a cascade of boosted covariance features, IEEE Transactions on Circuits and Systems for Video Technology 18 (8) (2008) 1140–1151.

[2] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[3] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 886–893.

[4] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: European Conference on Computer Vision, Springer, 2006, pp. 428–441.

[5] C. Wojek, S. Walk, B. Schiele, Multi-cue onboard pedestrian detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 794–801.

[6] S. Walk, N. Majer, K. Schindler, B. Schiele, New features and insights for pedestrian detection, in: IEEE conference on Computer vision and pattern recognition, 2010, pp. 1030–1037.

[7] X. Wang, T. X. Han, S. Yan, An HOG-LBP human detector with partial occlusion handling, in: IEEE International Conference on Computer Vision, 2009, pp. 32–39.

[8] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: An evaluation of the state of the art, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (4) (2012) 743–761.

[9] M. Tan, L. Wang, I. W. Tsang, Learning sparse svm for feature selection on very high dimensional datasets, in: International Conference on Machine Learning, 2010, pp. 1047–1054.

[10] D. Geronimo, A. M. Lopez, A. D. Sappa, T. Graf, Survey of pedestrian detection for advanced driver assistance systems, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (7) (2010) 1239–1258.

[11] P. Viola, M. Jones, Robust real-time object detection, International Journal of Computer Vision 4 (2001) 34–47.

[12] Y. Zheng, C. Shen, R. Hartley, X. Huang, Effective pedestrian detection using center-symmetric local binary/trinary patterns, CoRR, abs/1009.0892.

[13] M. Enzweiler, D. M. Gavrila, A multilevel mixture-of-experts framework for pedestrian classification, IEEE Transactions on Image Processing 20 (10) (2011) 2967–2979.

[14] P. Dollár, Z. Tu, P. Perona, S. Belongie, Integral channel features., in: British Machine Vision Conference, Vol. 2, 2009, p. 5.

[15] P. Dollár, S. Belongie, P. Perona, The fastest pedestrian detector in the west., in: British Machine Vision Conference, Vol. 2, 2010, p. 7.

[16] Z. Lin, L. S. Davis, A pose-invariant descriptor for human detection and segmentation, in: European Conference on Computer Vision, Springer, 2008, pp. 423–436.

[17] B. Wu, R. Nevatia, Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors, International Journal of Computer Vision 75 (2) (2007) 247–266.

[18] P. Sabzmeydani, G. Mori, Detecting pedestrians by learning shapelet features, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–8.

[19] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3626–3633.

[20] X. Ren, D. Ramanan, Histograms of sparse codes for object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3246–3253.

[21] J. J. Lim, C. L. Zitnick, P. Dollár, Sketch tokens: A learned mid-level representation for contour and object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3158–3165.

[22] P. Viola, M. J. Jones, D. Snow, Detecting pedestrians using patterns of motion and appearance, in: IEEE International Conference on Computer Vision, 2003, pp. 734–741.

[23] B. Wu, R. Nevatia, Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors, in: IEEE International Conference on Computer Vision, Vol. 1, 2005, pp. 90–97.

[24] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1627–1645.

[25] S. Maji, A. C. Berg, J. Malik, Classification using intersection kernel support vector machines is efficient, in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.

[26] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001, pp. I–511.

[27] D. Tang, Y. Liu, T.-K. Kim, Fast pedestrian detection by cascaded random forest with dominant orientation templates., in: British Machine Vision Conference, 2012, pp. 1–11.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, Cascade object detection with deformable part models, in: IEEE conference on Computer vision and pattern recognition, 2010, pp. 2241–2248.

[29] W. Ouyang, X. Wang, A discriminative deep model for pedestrian detection with occlusion handling, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3258–3265.

[30] W. Ouyang, X. Zeng, X. Wang, Modeling mutual visibility relationship in pedestrian detection, in: IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3222–3229.

[31] X. Zeng, W. Ouyang, X. Wang, Multi-stage contextual deep learning for pedestrian detection, in: IEEE International conference on Computer vision, 2013.

[32] I. Jolliffe, Principal component analysis, Wiley Online Library, 2005.

[33] M. Barker, W. Rayens, Partial least squares for discrimination, Journal of Chemometrics 17 (3) (2003) 166–173.

[34] W. R. Schwartz, A. Kembhavi, D. Harwood, L. S. Davis, Human detection using partial least squares analysis, in: IEEE International Conference on Computer Vision, 2009, pp. 24–31.

[35] J. Friedman, T. Hastie, R. Tibshirani, et al., Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), The annals of statistics 28 (2) (2000) 337–407.

[36] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27.

[37] A. Ess, B. Leibe, L. Van Gool, Depth and appearance for mobile scene analysis, in: IEEE International Conference on Computer Vision, 2007, pp. 1–8.

[38] P. Dollár, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: A benchmark, in: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, 2009, pp. 304–311.

[39] A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: Fast retina keypoint, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 510–517.