Optical Engineering

OpticalEngineering.SPIEDigitalLibrary.org

Improved Hough transform by modeling context with conditional random fields for partially occluded pedestrian detection

Linfeng Jiang
Huilin Xiong

SPIE.

Linfeng Jiang, Huilin Xiong, "Improved Hough transform by modeling context with conditional random fields for partially occluded pedestrian detection," *Opt. Eng.* **57**(6), 063101 (2018), doi: 10.1117/1.OE.57.6.063101.

Improved Hough transform by modeling context with conditional random fields for partially occluded pedestrian detection

Linfeng Jiang and Huilin Xiong*
Shanghai Jiao Tong University, Department of Automation, Shanghai, China

Abstract. Traditional Hough transform-based methods detect objects by casting votes to object centroids from object patches. It is difficult to disambiguate object patches from the background by a classifier without contextual information, as an image patch only carries partial information about the object. To leverage the contextual information among image patches, we capture the contextual relationships on image patches through a conditional random field (CRF) with latent variables denoted by locality-constrained linear coding (LLC). The strength of the pairwise energy in the CRF is measured using a Gaussian kernel. In the training stage, we modulate the visual codebook by learning the CRF model iteratively. In the test stage, the binary labels of image patches are jointly estimated by the CRF model. Image patches labeled as the object category cast weighted votes for object centroids in an image according to the LLC coefficients. Experimental results on the INRIA pedestrian, TUD Brussels, and Caltech pedestrian datasets demonstrate the effectiveness of the proposed method compared with other Hough transform-based methods. [DOI: 10.1117/1.OE.57.6.063101]

Keywords: computer vision; Hough transform; contextual information; conditional random field; pedestrian detection.

Paper 180310 received Feb. 25, 2018; accepted for publication May 15, 2018; published online Jun. 1, 2018.

1 Introduction

Pedestrian detection is a fundamental challenge in computer vision due to great variation in appearance, changes in illumination, poor resolution, and partial occlusions. The general framework of pedestrian detection can be decomposed into three modules: (i) generate the region proposals that represent object hypotheses in a test image, (ii) classify the region proposals, and (iii) refine the region proposals to obtain accurate localization of pedestrians.

In the past years, the use of Hough transform framework has attracted considerable attention for pedestrian detection.[1–10] The applicability of the Hough transform framework can be attributed to its robustness against partial occlusions, as indicated in Refs. 1 and 3–5. Another attractive property of the Hough transform is its simplicity. The Hough transform framework for pedestrian detection includes three primary steps: (i) construct visual codebook, (ii) cast probabilistic votes for object center into a Hough image according to the codebook using voting elements of the test image, and (iii) search maxima in the Hough image as object hypotheses. Although some Hough transform methods demonstrate the significance of the visual codebook and voting weights[1,2,4] for detection performance, none use contextual information. Voting elements, which denote the image patches classified into object categories, cast probabilistic votes into a Hough image.

However, the image patch contains only partial information about an object, and its appearance is highly variable. Thus, it is difficult to disambiguate object patches from background patches by a classifier at the local level. Therefore, detection performance can be reduced due to noisy votes cast by background patches. Fortunately, conditional random field (CRF) frameworks modeling context have achieved an impressive performance for semantic segmentation,[11–15] image classification,[16] saliency detection,[17] and object detection.[18] The CRF distribution can be formulated by a probabilistic graphical model, in which variables are interdependent rather than independent. Given an image, CRF inference is performed by a maximum a posteriori (MAP) or maximum posterior marginal criterion, and all patches can be classified into an object category or background simultaneously. In other words, the CRF model uses whole image information instead of local information to obtain all patch labels.

In this paper, we build a CRF model that regards the locality-constrained linear coding (LLC)[19] code of a local feature as a latent variable, which is more informative than the corresponding local feature. In addition, we apply a Gaussian kernel to neighboring features to measure the strength of pairwise energy in the CRF framework. In the training stage, we iteratively modulate the codebook and CRF model parameters by a max-margin approach with a maximum-likelihood criterion. Furthermore, to learn the spatial-occurrence distribution of the codebook, offset vectors of the local feature to its object center in a training image are assigned to matching codewords. In the detection stage, all image patches are classified into an object category or background simultaneously by CRF inference, and the patches classified into an object category are used as voting

*Address all correspondence to: Huilin Xiong, E-mail: hlxiong@sjtu.edu.cn; fine0228@sjtu.edu.cn

elements in the Hough transform. The voting element casts weighted votes into the Hough image according to its LLC coefficients on codewords, and the use of LLC enables us to reduce the reconstruction error for representing the voting element by a linear combination of codewords.[20] This may result in more balanced probabilistic votes than uniform votes in the Hough image. Maxima are regarded as object hypotheses in the Hough image, in which all votes accumulate. The proposed method makes three main contributions:

- It optimizes the codebook through CRF learning.
- It casts weighted votes into the Hough image by the encoding strategy.
- It jointly classifies all image patches into an object category or background according to the CRF model, which includes patch-level contextual constraints.

We evaluated our method on the INRIA pedestrian, TUD Brussels, and Caltech pedestrian datasets. This work compromises speed, accuracy, and simplicity. Experiments demonstrated the effectiveness of the proposed method compared with other Hough transform-based methods, benefiting from the contextual information in images and the weighted Hough voting strategy. The rest of the paper is structured as follows. We review literature on the Hough transform methods, encoding methods, and CRF in Sec. 2. We describe our method for pedestrian detection in Sec. 3. We evaluate the proposed method on several challenging datasets in Sec. 4, and we provide our conclusions in Sec. 5.

2 Related Work

In this section, we first discuss the Hough transform-based methods for pedestrian detection and then briefly describe encoding methods and CRF that are related to the proposed method.

2.1 *Hough Transform Methods*

There is extensive literature dedicated to pedestrian detection.[21–39] Here, we review the methods based on the Hough transform framework[1,2,4–68–10] that are most relevant to our work.

In the past years, applications of the methods based on the Hough transform framework have resulted in progress in pedestrian detection. The majority of Hough transform methods usually focus on codebook learning, voting element generation, and hypotheses search. The advantage of the Hough transform methods is that they can detect pedestrians with low computational cost due to the simple structure[9] and can also locate a partially occluded pedestrian in an image using a small set of local patches.[1,3–5] The implicit shaped model (ISM)[1] has been widely derived by other Hough transform-based methods, which constructs a visual codebook by clustering local features in an unsupervised manner. Gall and Lempitsky[2] proposed the Hough forest to build decision trees in a supervised manner, where a set of leaves can be regarded as a discriminative codebook that produces probabilistic votes with better voting performance. Barinova et al.[4] proposed an MAP inference method rather than non-maximum suppression (NMS) to seek the maxima in the Hough image. Wang et al.[5] proposed a structured Hough transform method that incorporates depth-dependent contexts into a codebook-based pedestrian detection model. Cabrera and Lpez-Sastre[6] proposed a boosted Hough forest, in which decision trees are trained in a stage-wise fashion to optimize a global loss function. Liu et al.[9] proposed a pair Hough model (PHM) for detecting objects whose voting elements were extracted from interest points to handle the rotation of objects. In a study by Liu et al.,[10] extremely randomized trees (ERTs) were constructed from features of soft-labeled training blobs, and a Hough image was accumulated by votes from features based on the soft-labeled ERTs. Different from other Hough transform methods, the proposed method regards LLC codes as hidden variables in a unified CRF framework that exploits the contextual information between neighboring image patches, from which the visual codebook and CRF parameters are learned in a supervised manner.

2.2 *Encoding Methods*

Many approaches for encoding local features (image patches) have been proposed.[19,20,40] Lazebnik et al.[40] proposed spatial pyramid matching (SPM), which is a simple and computationally efficient extension of an orderless bag-of-features image representation. Yang et al.[20] developed an extension of the SPM method called ScSPM for nonlinear codes. Wang et al.[19] proposed LLC in place of the vector quantization (VQ) coding in traditional SPM utilizing the locality constraint to project each local feature into its local coordinate system. Moreover, dictionary learning plays a significant role in encoding.[17,41] Bach et al.[41] demonstrated that better results can be obtained when dictionary is modulated to the specific task. Yang and Yang[17] proposed a top-down saliency model that jointly learns a discriminative dictionary and a CRF to improve sparse coding (SC). However, codebooks optimized in these methods are utilized for image classification or saliency detection rather than Hough transform-based pedestrian detection.

The LLC can represent local features by codewords with lower reconstruction error than VQ[42] and SC.[20] This property of LLC motivated us to utilize the code coefficients of a voting element as codeword weights to cast better balanced votes in the Hough image.

2.2.1 *Locality-constrained linear coding*

Feature encoding decomposes a local feature $\mathbf{x}$ into a linear combination of codewords over the predefined codebook $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_M] \in \mathbb{R}^{N \times M}$, where $\mathbf{c}_i$ denotes the i'th codeword that is N-dimensional. While the SC[20] method applies a sparsity constraint to select similar codewords of local features from a codebook, the LLC method[19] incorporates a locality constraint that must lead to a sparsity constraint but not necessarily vice versa. The visual information of image patches contained in the codebook is transferred into the latent variables of the CRF model by the LLC, which is more informative than local features. The LLC code of a local feature $\mathbf{x}$ is obtained by solving the following optimization problem:

$$L(\mathbf{x}, \mathbf{C}) = \arg\min_{\mathbf{l}} \|\mathbf{x} - \mathbf{C}\mathbf{l}\|^2 + \lambda \|\mathbf{d} \odot \mathbf{l}\|^2 \quad \text{s.t. } \mathbf{1}^{\mathrm{T}}\mathbf{l} = 1, \tag{1}$$

where $\odot$ denotes the element-wise multiplication, λ is used to control the locality constraint, $\mathbf{l}$ is the vector of weights corresponding to the codewords, and $\mathbf{d} \in \mathbb{R}^M$ is the locality

adaptor that corresponds to the similarities between the codewords and local feature $\mathbf{x}$. Specifically

$$\mathbf{d} = \exp\left[\frac{\text{dist}(\mathbf{x}, \mathbf{C})}{\sigma}\right], \tag{2}$$

where $\text{dist}(\mathbf{x}, \mathbf{C}) = [\text{dist}(\mathbf{x}, \mathbf{c}_1), \ldots, \text{dist}(\mathbf{x}, \mathbf{c}_M)]^\top$, and $\text{dist}(\mathbf{x}, \mathbf{c}_i)$ denotes the Euclidean distance between $\mathbf{x}$ and $\mathbf{c}_i$. σ denotes the weight-decay speed for the locality adaptor. Note that the LLC code in Eq. (1) is not sparse in the sense of the l^0 norm, but it is sparse in the sense that the solution has few significant values. In the LLC method, the solution of the optimization problem can be translated into the following equation:

$$\tilde{L}(\mathbf{x}, \mathbf{C}) = [(\mathbf{C} - \mathbf{1}\mathbf{x}^\top)(\mathbf{C} - \mathbf{1}\mathbf{x}^\top)^\top + \lambda \text{diag}(\mathbf{d})] \setminus \mathbf{1}, \tag{3}$$

where $(\mathbf{C} - \mathbf{1}\mathbf{x}^\top)(\mathbf{C} - \mathbf{1}\mathbf{x}^\top)^\top$ denotes the data covariance matrix, $\setminus$ denotes matrix left division, λ is a parameter controlling the locality constraint, and $\mathbf{1} \in \mathbb{R}^M$ indicates the constant $\mathbf{1}$ vector

$$L(\mathbf{x}, \mathbf{C}) = \tilde{L}(\mathbf{x}, \mathbf{C}) / \mathbf{1}^\top \tilde{L}(\mathbf{x}, \mathbf{C}), \tag{4}$$

where / denotes the division. Equation (4) is used for vector unitization.

2.3 *Conditional Random Field*

A CRF is a flexible framework for modeling contextual information that can be grouped into three levels: pixels, patches, and objects. It is widely used for image semantic segmentation and patch-level labeling[11–15,18] by addressing computer vision problems with CRF inference. Kumar and Hebert[18] proposed the discriminative random field, which inherits the CRF concept for labeling man-made structures at patch level. To disambiguate local image information, He et al.[11] proposed a multi-CRF with three separate components at different scales for image semantic segmentation. Quattoni et al.[16] proposed a hidden-state CRF for image classification that models the latent structure of the input domain via intermediate hidden variables. Toyoda and Hasegawa[12] proposed a CRF incorporating local and global image information. Thus, global consistency of layouts is achieved from a global viewpoint. Shotton et al.[13] proposed a CRF model for semantic segmentation that uses a texture-layout filter incorporating texture, layout, and contextual information. Owing to the need to solve excessive boundary smoothing for semantic segmentation using an adjacency CRF structure, Krähenbühl and Koltun[14] proposed a fully connected CRF that establishes pairwise potentials consisting of a linear combination of Gaussian kernels on all pairs of pixels in the image. Chen et al.[15] proposed a DeepLab system that utilizes a fully connected CRF coupled with a deep convolutional network-based pixel-level classifier as well as long range dependencies to capture fine edge details. Yang and Yang[17] proposed a top-down saliency model by constructing a CRF upon SC of image patches; the codebook was optimized by jointly learning the CRF model. To speed-up the saliency detection procedure, Yang and Xiong[43] proposed a saliency detection method by combining LLC and CRF. While these saliency detection methods use CRF to generate saliency maps directly, the proposed method builds the CRF model to obtain Hough voting elements.

The CRF[13,18] is a conditional distribution over the labels $\mathbf{Y} = \{y_i\}_{i \in S}$ given the observations $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$, which can be written as

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z} \exp\left\{\sum_{i \in S} \phi_i(y_i|\mathbf{X}) + \alpha \sum_{i \in S} \sum_{j \in N_i} \phi_{ij}(y_i, y_j|\mathbf{X})\right\}, \tag{5}$$

where Z is a normalizing constant known as the partition function, ϕ_i and ϕ_{ij} are the unary and pairwise potentials, respectively, S is a set of sites that refers to elements (pixels or patches) in an image, N_i is a set of neighbors of site i, and α is a coefficient that modulates the effect of the pairwise potential ϕ_{ij}. In general, the unary potential ϕ_i denotes the penalty for a local classifier applied to an image patch and ignoring its neighbors. The pairwise potential ϕ_{ij} is seen as a penalty of label inconsistency that assumes neighboring pixels or patches should be classified into the same object category.

3 Our Method

Our pedestrian detection system consists of two modules: (i) a CRF model with latent variables denoted by LLC codes of image patches. The visual codebook can be optimized by learning this model and can further learn a spatial-occurrence distribution that specifies where each codeword may be found on the object. (ii) A Hough voting module. Patch labels are jointly estimated in a test image by CRF inference, and the patches classified into the object category are voting elements that cast weighted votes into the Hough image. Maxima in the Hough image are regarded as object hypotheses. An overview of the detection procedure is shown in Fig. 1.

3.1 *Conditional Random Field Model*

We exploit the contextual information in an image by a CRF model that uses LLC codes as latent variables and applying a Gaussian kernel to measure the strength of pairwise energy. This model is used for two purposes: (i) to optimize the codebook by learning the CRF model and (ii) to jointly classify image patches into the object category or background by CRF inference. To reduce Hough image noise resulting from background patches, image patches classified into the object category are used as voting elements (Sec. 3.4).

Yang and Yang[17] developed a CRF model upon SC of image patches for saliency detection. Inspired by this CRF model, we build a CRF framework for modeling the context constraint that uses a Gaussian kernel to measure the local feature similarity between neighboring nodes for pairwise energy

$$P[\mathbf{Y}|L(\mathbf{X}, \mathbf{C}), \mathfrak{v}] = \frac{1}{Z} e^{-E[L(\mathbf{X}, \mathbf{C}), \mathbf{Y}, \mathfrak{v}]}, \tag{6}$$

where Z is the partition function for normalization, $\mathbf{X} = \{\mathbf{x}_i\}_{i \in S}$ denotes a set of local features that is sampled from different sites S of the image, $\mathbf{Y} = \{y_i\}_{i \in S}$ denotes the corresponding labels, $\mathbf{C}$ is the visual codebook, $E[L(\mathbf{X}, \mathbf{C}), \mathbf{Y}, \mathfrak{v}]$ is the energy function, $L(\mathbf{X}, \mathbf{C}) = \{L(\mathbf{x}_i, \mathbf{C})\}_{i \in S}$ are the latent variables denoting LLC codes

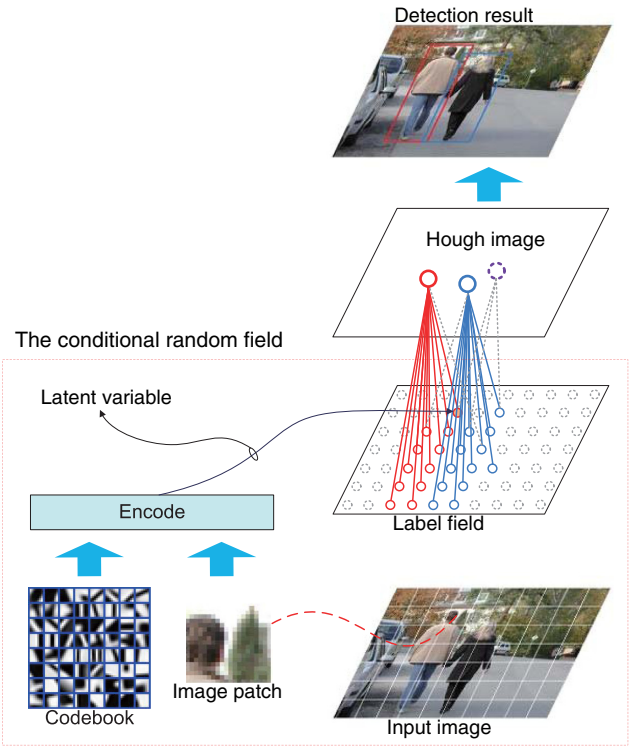

Fig. 1 Overview of the detection procedure. Local features (image patches) are densely extracted from the input image and encoded by LLC as latent variables in the CRF model; the codebook, as a visual dictionary, represents a set of object parts; all patches in the input image are classified into the object category or background simultaneously by CRF inference. The label field indicates a set of category labels on all image patches. Image patches classified into the object category are regarded as voting elements. A voting element casts weighted votes into the Hough image by its LLC code. A Hough image was accumulated by votes from voting elements. Maxima in the Hough image are regarded as object hypotheses. Best viewed in color.

of a set of local features $\mathbf{X}$, and $\boldsymbol{\upsilon} = [\boldsymbol{\upsilon}_1; \boldsymbol{\upsilon}_2]$ is the model parameter vector. For clarity, we simplify the notation by writing $\mathbf{l}_i \triangleq L(\mathbf{x}_i, \mathbf{C})$ and $\mathbf{L} \triangleq L(\mathbf{X}, \mathbf{C})$. The energy function is decomposed into unary and pairwise energy terms

$$E(\mathbf{L}, \mathbf{Y}, \boldsymbol{\upsilon}) = \sum_{i \in S} \varphi_i(\mathbf{l}_i, y_i, \boldsymbol{\upsilon}_1) + \sum_{i \in S} \sum_{j \in N_i} \varphi_{ij}(\mathbf{l}_i, \mathbf{l}_j, y_i, y_j, \boldsymbol{\upsilon}_2), \tag{7}$$

where S is a set of sites that refers to patches in an image and N_i is a set of neighbors of site i. The unary energy φ_i can be measured by the total contribution of sparse codes $-\mathrm{y}_i \boldsymbol{\upsilon}_1^{\mathrm{T}} \mathbf{l}_i$, where $\boldsymbol{\upsilon}_1 \in \mathbb{R}^M$ is the weight vector and M denotes the number of codewords. The pairwise energy φ_{ij} can be denoted as $\boldsymbol{\upsilon}_2 G(\mathbf{l}_i, \mathbf{l}_j)\mu(y_i, y_j)$, where the scalar $\boldsymbol{\upsilon}_2$ measures the weight of the pairwise energy term, $G(\mathbf{l}_i, \mathbf{l}_j)$ is a Gaussian kernel to measure the strength of pairwise energy, and μ is an indicator function equaling 1 for different labels. The Gaussian kernel is defined as

$$G(\mathbf{l}_i, \mathbf{l}_j) = \exp\left(-\frac{|\mathbf{l}_i - \mathbf{l}_j|^2}{2\theta^2}\right), \tag{8}$$

where $\mathbf{l}_i$ and $\mathbf{l}_j$ denote the LLC codes of neighboring local features $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively. The degree of similarity is controlled by the parameter θ.

Like most CRF models,[11–13] the energy function is linear with the parameter $\boldsymbol{\upsilon} = [\boldsymbol{\upsilon}_1; \boldsymbol{\upsilon}_2]$, but it is nonlinear with the codebook $\mathbf{C}$, which is implicitly defined by $L(\mathbf{x}, \mathbf{C})$ in Sec. 2.2. This nonlinear parametrization makes it challenging to learn the model. We discuss the learning approach in Sec. 3.2.

3.2 *Joint CRF and Codebook Learning*

Following Yang and Yang's[17] method, we learn the CRF parameters and codebook in accordance with the CRF model. Let $\mathcal{X} = \{\mathbf{X}^{(k)}\}_{k=1}^K$ be a set of K training images and $\mathcal{Y} = \{\mathbf{Y}^{(k)}\}_{k=1}^K$ be corresponding set of labels. We aim to estimate the CRF parameter vector $\boldsymbol{\upsilon}$ and the codebook $\mathbf{C}$ by maximizing the joint likelihood of training data

$$\max_{\boldsymbol{\upsilon} \in \mathbb{R}^{M+1}, \mathbf{C} \in \mathcal{C}, \mathbf{L}^{(k)}} \prod_{k=1}^K P\{\mathbf{Y}^{(k)} | L[\mathbf{X}^{(k)}, \mathbf{C}], \boldsymbol{\upsilon}\}, \tag{9}$$

where $\mathbf{L}^{(k)} \triangleq L[\mathbf{X}^{(k)}, \mathbf{C}]$ and $\mathcal{C}$ is the convex set of codebooks that satisfies the following constraint:

$$\mathcal{C} = \{\mathbf{C} \in \mathbb{R}^{N \times M}, \|\mathbf{c}_i\|_2 \le 1, \forall i = 1, 2, \ldots, M\}. \tag{10}$$

The evaluation of the partition function Z of Eq. (6) is an NP-hard problem. Referring to the max-margin CRF learning approach,[44] we look for the optimal weights $\boldsymbol{\upsilon}$ and codebook $\mathbf{C}$ that assign the training labels $\mathbf{Y}^{(k)}$, a probability that is greater than or equal to any other labeling $\mathbf{Y}$ of instance k

$$P[\mathbf{Y}^{(k)} | \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] \ge P[\mathbf{Y} | \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] \quad \forall \mathbf{Y} \setminus \mathbf{Y}^{(k)} \quad \forall k. \tag{11}$$

The partition function Z can be canceled from both sides of the constraints [Eq. (7)], and we express the constraints in terms of energies

$$E[\mathbf{Y}^{(k)}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] \le E[\mathbf{Y}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}]. \tag{12}$$

Moreover, we desire the energy of ground truth $E[\mathbf{Y}^{(k)}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}]$ to be lower than that of any other energies $E[\mathbf{Y}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}]$ of label configurations on the training data. Thus, we have a new constraint set

$$E[\mathbf{Y}^{(k)}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] \le E[\mathbf{Y}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] - \Delta[\mathbf{Y}, \mathbf{Y}^{(k)}]. \tag{13}$$

The margin function $\Delta[\mathbf{Y}, \mathbf{Y}^{(k)}] = \sum_{i=1}^m \mathrm{I}[y_i, y_i^{(k)}]$, where I is an indicator function equal to 1 for different labels. There are an exponential number of constraints with respect to labeling $\mathbf{Y}^{(k)}$ for each training image. Inspired by the cutting plane algorithm,[45] the most violated constraints can be found by solving

$$\hat{\mathbf{Y}}^{(k)} = \arg \min_{\mathbf{Y}} E[\mathbf{Y}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] - \Delta[\mathbf{Y}, \mathbf{Y}^{(k)}]. \tag{14}$$

Therefore, the optimal weight $\boldsymbol{\upsilon}$ and the codebook $\mathbf{C}$ can be learned by minimizing the following objective function:

$$\min_{\boldsymbol{\upsilon}, \mathbf{C} \in \mathcal{C}} \frac{\gamma}{2} \|\boldsymbol{\upsilon}\|^2 + \sum_{k=1}^K \ell^k(\boldsymbol{\upsilon}, \mathbf{C}), \tag{15}$$

Algorithm 1 Joint CRF and codebook learning

Input: $\mathcal{X}$ (training images) and $\mathcal{Y}$ (patch labels); $\mathbf{C}^{(0)}$ (initial dictionary); $\boldsymbol{\upsilon}^{(0)}$ (initial CRF weight vector); T (number of iterations); K (number of training images).

Output: the codebook $\mathbf{C}$ and the weight $\boldsymbol{\upsilon}$.

for $t = 1$ to T **do**

  Permute training samples $(\mathcal{X}, \mathcal{Y})$

  For $k = 1$ to K **do**

    Evaluate the latent variables $\mathbf{l}_i$ by Eq. (1)

    Solve the most violated labeling $\hat{\mathbf{Y}}^{(k)}$ by Eq. (14)

    Update the weight $\boldsymbol{\upsilon}^t$ and codebook $\mathbf{C}^t$ by the loss function $\ell^k(\boldsymbol{\upsilon}, \mathbf{C})$

  end for

end for

where $\ell^k(\boldsymbol{\upsilon}, \mathbf{C}) \triangleq E[\hat{\mathbf{Y}}^{(k)}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}] - E[\mathbf{Y}^{(k)}, \mathbf{L}^{(k)}, \boldsymbol{\upsilon}]$ and γ controls the regularization of the weight $\boldsymbol{\upsilon}$.

The above objective function is optimized by a stochastic gradient descent algorithm, which is summarized in Algorithm 1.

3.3 *Learning the Spatial-Occurrence Distribution*

In this section, we learn the nonparametric spatial-occurrence distribution P_C for each codeword of the optimized codebook $\mathbf{C}$, which can be used to cast votes into the Hough image in the test stage. An occurrence represents an image patch of the training images, which matches a codeword. As in the other Hough transform methods,[1,4,5] a codeword represents a specific object part whose position relative to the object center is uncertain. Each codeword corresponds to a set of occurrences in the training images.

As shown in Algorithm 2, we perform an iteration over all training images to match the codewords to local features. Here, we activate the codewords whose similarity exceeds a matching threshold of 0.7 (discussed in Sec. 4.1). For every codeword, we store all occurrence positions that reflect its spatial distribution over the object area in a nonparametric form (as a list of occurrences).

3.4 *Weighted Hough Voting Strategy*

In Sec. 3.3, the visual codebook $\mathbf{C}$ was optimized by learning the CRF model iteratively, and voting elements were obtained by CRF inference in the test image. We now describe the Hough voting procedure based on the CRF model that regards the LLC code of an image patch as a latent variable. A flowchart of the detection procedure is shown in Fig. 1. The voting element consistently casts weighted votes into the Hough image according to its LLC code. To locate the objects in the test image, maxima in the Hough image are regarded as object hypotheses. Moreover, to handle scale variations, a test image is resized by a set of scale factors, and hypotheses are computed independently in the Hough images at each scale.

Algorithm 2 Learning the spatial-occurrence distribution

Input: $\mathcal{X}$ (training images); K (number of training images); $\mathbf{C}$ (the codebook learned in Algorithm 1); M (number of codewords).

Output: the occurrences U.

//$U[m]$, a list of occurrences, denotes the spatial distribution of codeword $\mathbf{c}_m$ in a nonparametric manner.

for $m = 1$ to M **do**

  $U[m] = \varnothing$//Initialize occurrences for codeword $\mathbf{c}_m$.

end for

for $k = 1$ to K **do**

  Let (o_x, o_y) be the object center.

  Extract local features in image $\mathbf{X}^{(k)}$.

  for $j = 1$ to J **do**// J local features in image $\mathbf{X}^{(k)}$.

    Let $\mathbf{x}_j$ be the local feature at location (l_x, l_y, l_s).

    for $m = 1$ to M **do**

      if similarity $(\mathbf{c}_m, \mathbf{x}_j) \geq t$ **then**

        //Record an occurrence of codeword $\mathbf{c}_m$

        $U[m] = U[m] \cup (o_x - l_x, o_y - l_y, l_s)$

      end if

    end for

  end for

end for

Different from other Hough transform approaches,[1,2,4–6,8–10] our Hough voting procedure is cast into a probabilistic framework with a coding strategy. Let $\mathbf{x}$ be the local feature observed at location $\tilde{\mathbf{l}}$ in the test image. By matching it to the visual codebook, a set of valid interpretations $\mathbf{c}_i$ with probabilities $p(\mathbf{c}_i|\mathbf{x}, \tilde{\mathbf{l}})$ can be obtained. If a codeword matches, it casts votes for different object positions. That is, for every $\mathbf{c}_i$, votes for several object categories O_n and a position $\mathbf{h}$ can be obtained according to the learned spatial-occurrence distribution $p(O_n, \mathbf{h}|\mathbf{c}_i, \tilde{\mathbf{l}})$. The voting probability of a local feature can be formally expressed by the following marginalization:

$$p(O_n, \mathbf{h}|\mathbf{x}, \tilde{\mathbf{l}}) = \sum_i p(O_n, \mathbf{h}|\mathbf{x}, \mathbf{c}_i, \tilde{\mathbf{l}}) p(\mathbf{c}_i|\mathbf{x}, \tilde{\mathbf{l}}), \tag{16}$$

for $i = 1, \ldots, N$, where N is the number of codewords. Since the unknown local feature $\mathbf{x}$ has been replaced by a known interpretation $\mathbf{c}_i$ in the test image, the first term can be considered independent from $\mathbf{x}$. Also, local features matched to the codebook are independent of their location. Thus, the equation is reduced to

$$p(O_n, \mathbf{h}|\mathbf{x}, \tilde{\mathbf{l}}) = \sum_i p(O_n, \mathbf{h}|\mathbf{c}_i, \tilde{\mathbf{l}}) p(\mathbf{c}_i|\mathbf{x}), \tag{17}$$

$$= \sum_i p(\mathbf{h}|O_n, \mathbf{c}_i, \tilde{\mathbf{l}}) p(O_n|\mathbf{c}_i, \tilde{\mathbf{l}}) p(\mathbf{c}_i|\mathbf{x}), \tag{18}$$

where $p(\mathbf{h}|O_n, \mathbf{c}_i, \tilde{\mathbf{l}})$ is the voting probability for an object position given its category label O_n, codeword $\mathbf{c}_i$, and location $\tilde{\mathbf{l}}$. The probability $p(O_n|\mathbf{c}_i, \tilde{\mathbf{l}})$ denotes the confidence that the codeword is matched on the object category O_n against the background. Finally, $p(\mathbf{c}_i|\mathbf{x})$ denotes the probability that local feature $\mathbf{x}$ matches to codeword $\mathbf{c}_i$. The object scale is regarded as a third dimension in the voting space. If a local feature extracted from location (x, y, s) matches a codeword that has been observed at position $(x_{\tilde{l}}, y_{\tilde{l}}, s_{\tilde{l}})$ on a training image, it votes for the following coordinates:

$$x_{\text{vote}} = x - x_{\tilde{l}}(s/s_{\tilde{l}}), \tag{19}$$

$$y_{\text{vote}} = y - y_{\tilde{l}}(s/s_{\tilde{l}}), \tag{20}$$

$$s_{\text{vote}} = s/s_{\tilde{l}}. \tag{21}$$

Thus, the voting probability $p(\mathbf{h}|O_n, \mathbf{c}_i, \tilde{\mathbf{l}})$ is obtained by summing the votes for all stored observations from the learned occurrence distribution P_c. The ensemble of all such votes is used to obtain a nonparametric probability density estimate for the position of the object center.

The probability $p(\mathbf{c}_i|\mathbf{x})$ of a match between a local feature and codeword is obtained according to the LLC algorithm[19] described above. In other words, the LLC code $\mathbf{l} = L(\mathbf{x}, \mathbf{C})$ is regarded as weighted probabilities for Hough voting.

Next, maxima are sought to be object hypotheses in the Hough voting space, in which all votes are accumulated. The search process includes two stages. We first accumulate the voting probabilities in a three-dimensional Hough space and find maxima as candidates. We then employ the mean-shift algorithm[1] to refine the locations of hypotheses. Intuitively, the probability $p(O_n, \mathbf{h})$ of an object hypothesis is obtained by summing the individual voting probabilities $p(O_n, \mathbf{h}, \mathbf{x}_k, \tilde{\mathbf{l}}_k)$ over all observations, and we arrive at the following equation:

$$p(O_n, \mathbf{h}) = \sum_k p(O_n, \mathbf{h}|\mathbf{x}_k, \tilde{\mathbf{l}}_k) p(\mathbf{x}_k, \tilde{\mathbf{l}}_k), \tag{22}$$

for $k = 1, \ldots, K$, where K is the number of local features in the test image. $p(\mathbf{x}_k, \tilde{\mathbf{l}}_k)$ is the probability of local feature $(\mathbf{x}_k, \tilde{\mathbf{l}}_k)$ being sampled for object O_n located at $\mathbf{h}$. Nonetheless, it is necessary to tolerate small shape deformations to be robust for intraclass variations of the object. Thus, the mean-shift framework[1] is formulated with the following kernel density estimate:

$$\hat{p}(O_n, \mathbf{h}) = \frac{1}{V_b} \sum_k \sum_j p(O_n, \mathbf{h}_j|\mathbf{x}_k, \tilde{\mathbf{l}}_k) G\left(\frac{\mathbf{h} - \mathbf{h}_j}{b}\right), \tag{23}$$

where the Gaussian kernel G is a radially symmetric, nonnegative function, centered at zero and integrating to one, b is the kernel bandwidth, and V_b is its volume. The mean-shift search using this formulation will quickly converge to local modes of the underlying distribution. Moreover, the search procedure can be interpreted as kernel density estimation for the position of the object center.

Candidates of objects with high scores are usually close to each other in the Hough image. This may lead to the same object corresponding to multiple candidates, resulting in false positives. To reduce redundancy, we adopt NMS on the overlapped object hypotheses. We fix the intersection over union (IoU) threshold for NMS at 0.7.

4 Experiments

4.1 *Datasets*

To evaluate the effectiveness of the proposed method in different scenes, we choose three publicly available pedestrian datasets, namely, INRIA pedestrian, TUD Brussels, and Caltech pedestrian. Pedestrians in these datasets are mostly upright but are of different degrees of occlusions, and pose and scale changes, together with the variations in background and illuminations.

4.1.1 *INRIA Pedestrian*

The INRIA pedestrian dataset consists of 614 training images and 288 test images, which is challenging due to the variability of pedestrian poses, illumination changes, and highly cluttered backgrounds (mountains, buildings, vehicles, etc.).

4.1.2 *TUD Brussels*

The TUD Brussels dataset contains 508 images (one pair per second) at a resolution of 640×480, which are recorded from a car driving in the inner city of Brussels. This dataset is challenging due to partial occlusion, cluttered backgrounds (e.g., poles, parked cars, buildings, and crowds), and numerous small-scale pedestrians.

4.1.3 *Caltech Pedestrian*

The Caltech pedestrian dataset and its associated benchmark are among the most popular pedestrian detection datasets. It consists of about 10 h of videos (30 frames per second) collected from a vehicle driving through urban traffic. Every frame in the Caltech dataset has been densely annotated with the bounding boxes of pedestrian instances. In total, there are 350,000 bounding boxes of about 2300 unique pedestrians labeled in 250,000 frames. The pedestrians in the Caltech pedestrian dataset appear in many positions, orientations, and background variety. In the reasonable evaluation setting, the performance is evaluated on pedestrians over 50-pixels tall with no or partial occlusion.

4.2 *Experiment Procedure*

All experiments are carried out on a workstation equipped with a Titan Xp GPU and an Intel Xeon(R) CPU E5-2620 v4 @ 2.10 GHz. The evaluation tool is based on the codes from the official websites of Caltech and PASCAL VOC. Bounding boxes of objects are predicted in an image at test time. By default, predicted bounding boxes are considered positives when the IoU overlaps by more than 0.5 with ground-truth bounding boxes, and the rest are considered negatives. We use precision recall (PR) curve to evaluate pedestrian datasets.[4,26,28] Following,[9,28] we use average precision (AP) to measure detection performance on these datasets, which denotes the area under the

PR curve. The AP was calculated in accordance with the criteria of PASCAL VOC.

We densely extract scale-invariant feature transform features from images with a step length of 16 pixels. The codebook is optimized by training the CRF model with 12 iterations. The matching threshold is set to 0.7 for learning the spatial-occurrence distribution of the optimized codebook **C** (Sec. 3.3). The number K of LLC neighbors is set to 20. The codebook size M is set to 512. Implemented on a CPU to detect pedestrians from the Caltech pedestrian dataset, the Hough transform-based ISM[1] and Barinova et al.'s method[4] require 0.48 and 0.55 s per image, respectively, whereas the proposed method requires 0.62 s per image. Our method only requires 0.14 s (per image) extra computational time than ISM, because it mainly benefits from the efficient LLC[19] and inference algorithms in the CRF model.

4.3 *Result Analysis*

Figure 2 shows the PR curves of our method compared to conventional pedestrian detection approaches (HOG,[21] FPDW,[23] CrossTalk,[25] LatSvm-V2,[22] ACF,[30] Roerei,[26] MT-DPM,[27] and NAMC[32]) on the INRIA pedestrian, TUD Brussels, and Caltech pedestrian datasets according to the reasonable setting. The APs of these methods are shown in Table 1. It can be observed that our method obtained obvious improvements over the Hough transform-based methods[1,4,9] on these datasets. This is mainly attributable to two properties of our method that solve two challenging problems in the INRIA, TUD, and Caltech datasets: (i) the proposed method relies on image patches; hence, it can cope with the partial occlusions that are common in pedestrian datasets and (ii) the CRF model can effectively reduce the voting noise generated by the cluttered background.

We further evaluated the proposed method on three subsets of the Caltech pedestrian dataset according to its evaluation settings ("Occ = none," "Occ = partial," and "Occ = heavy"). Pedestrians are full, 65% to 100%, and 20% to 65% on those three settings, respectively. Table 2 shows that our method achieved APs of 66.4%, 47.3%, and 25.5% on these respective evaluation settings. Our method shows obvious improvements over the Hough transform-based methods[1,4,9] on these evaluation settings.

For the TUD pedestrian dataset, we masked ground-truth objects with proportions of 20%, 40%, and 60% from the left to right side, respectively, owing to an absence of occlusion information in this dataset. As shown in Fig. 3, our method has obvious improvements on these masked proportions compared to Hough transform-based ISM[1] and Barinova et al.'s[4] method.

In addition, we verified the significance of codebook optimization, codebook size, number of LLC neighbors, and weighted voting strategy on detection performance.

Table 1 Performance comparison in terms of AP (%) on the INRIA, TUD Brussels, and Caltech pedestrian datasets according to the reasonable setting.

Dataset	INRIA	TUD	Caltech
HOG[21]	73.3	40.1	26.5
LatSvm-V2[22]	91.0	51.5	35.9
Roerei[26]	93.9	54.8	51.9
FPDW[23]	88.3	60.3	40.3
CrossTalk[25]	88.7	60.0	45.1
ACF[30]	90.6	63.6	47.9
NAMC[32]	91.7	—	**66.7**
ISM[1]	86.0	54.2	49.5
Barinova et al.'s[4]	90.2	58.4	57.3
PHM[9]	86.5	—	—
Ours	**94.4**	**67.1**	65.0

Note: The bold values denote the best detection performances in terms of AP.

4.3.1 *Impact of the codebook optimization*

We initialized the codebook by the K-means clustering algorithm and then optimized the codebook by learning the CRF model. The codebook optimization was driven by top-down prior knowledge in a supervised manner. As shown in Fig. 4(a), detection performance improved rapidly in the first several iterations and converged after 12 iterations.

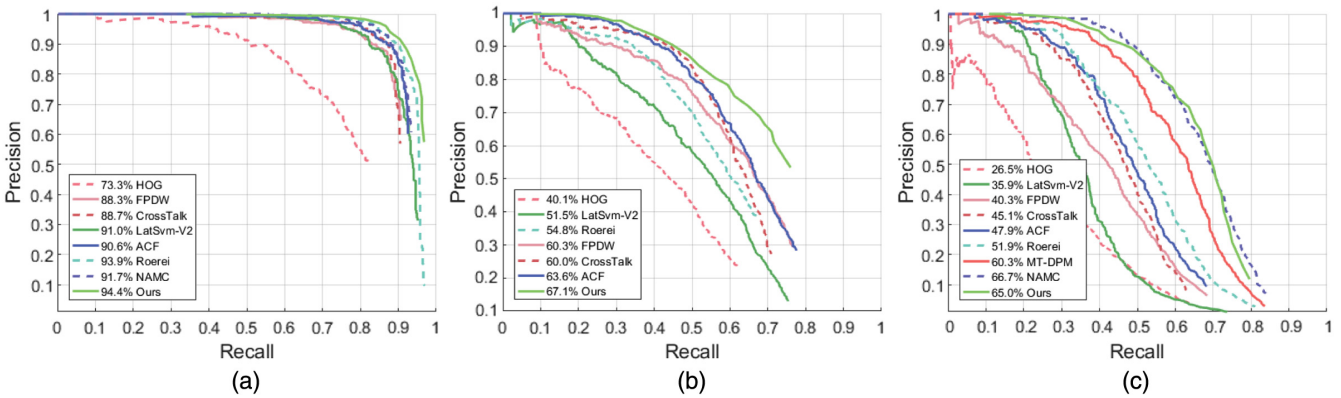

Fig. 2 Detection performance comparisons of our method and other methods on the (a) INRIA, (b) TUD Brussels, and (c) Caltech pedestrian datasets according to the reasonable setting. Best viewed in color.

Table 2 Detection performance comparisons of our method and other methods on three Caltech evaluation settings ("Occ = none," "Occ = partial," and "Occ = heavy").

Method	Occ = none	Occ = partial	Occ = heavy
MT-DPM + Context[27]	65.6	16.3	7.7
NAMC[32]	69.4	22.7	3.9
DeepCascade[33]	71.6	26.9	5.3
SCF + AlexNet[46]	80.5	34.5	15.3
TA-CNN[35]	81.4	45.9	16.4
SA-FastRCNN[37]	91.3	44.5	14.4
DeepParts[47]	89.5	**67.1**	24.2
F-DNN + SS[38]	**92.8**	60.4	**30.9**
Ours	66.4	47.3	25.5

Note: The bold values denote the best detection performances in terms of AP.

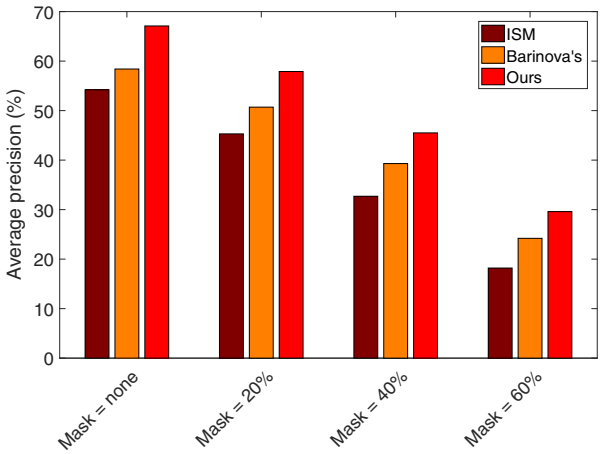

Fig. 3 Detection performance comparisons of our method and other methods on the TUD Brussels dataset with several masked proportions (none, 20%, 40%, and 60%). Our method achieved APs of 67.1%, 57.9%, 45.5%, and 29.6% on these respective masked proportions, which shows obvious improvements over the other Hough transform-based methods.

The stochastic nature of the learning algorithm resulted in some performance perturbation in some iterations.

4.3.2 *Impact of the matching threshold*

At test time, occurrence distributions of the codebook C were used to cast votes into the Hough image for pedestrian detection; thus, they are significant to detection performance of the proposed method. Learning occurrence distributions mainly depends on the matching threshold that represents the similarity between a codeword and an object patch of a training image. Intuitively, the occurrence distributions may be impacted by noise when the matching threshold is set to a relatively low value. On the contrary, the occurrence distributions are likely to lack some important occurrences when the matching threshold is set to a relatively high value. To find the optimal matching threshold, we evaluated the detection performance with different values of the matching threshold. Figure 4(b) shows the detection results on the INRIA pedestrian and TUD Brussels datasets with different values of the matching threshold. We found that our method achieved a relatively high AP when the matching threshold was 0.7.

4.3.3 *Impact of the LLC parameter K*

To focus on the impact of the number K of LLC neighbors, the codebook size was fixed at 512. As shown in Fig. 4(c), detection performance improved dramatically when K was <15, and it converged when K was >20. The experimental results show that the number of LLC neighbors had a great impact on detection performance.

4.3.4 *Impact of the codebook size*

To investigate the impact of codebook size on detection performance, we compared detection performance with codebook sizes of 256 and 512, with the parameter K of LLC fixed at 20. As shown in Table 3, the AP was 92.6% when $M = 256$ on the INRIA pedestrian dataset and 94.4% when $M = 512$. The AP was 62.7% when $M = 256$ on the TUD Brussels dataset and 67.1% when $M = 512$. We found that $M = 512$ gives better detection results than $M = 256$.

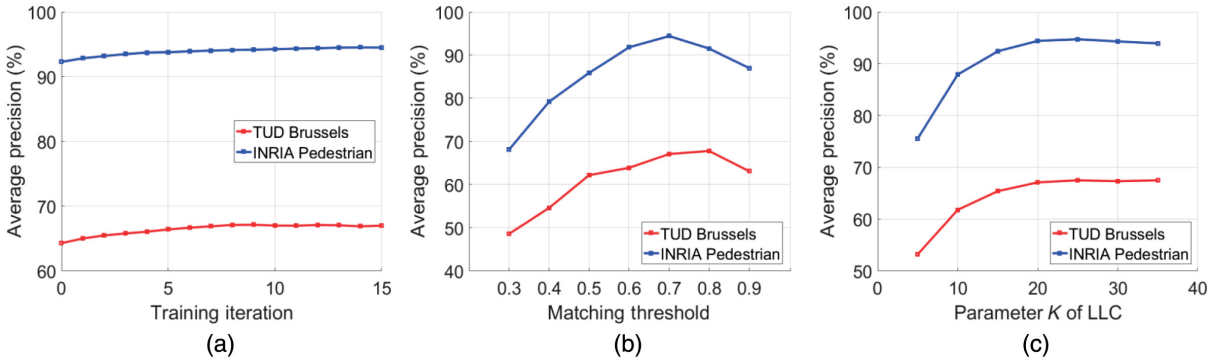

Fig. 4 (a) Detection results of our method when the matching threshold varies. (b) Detection results when the parameter K of LLC varies and codebook size M is 512. (c) Performance gain with training iterations when the parameter K of LLC is 20 and codebook size M is 512.

Table 3 Performance comparison in terms of codebook size M on the TUD Brussels and INRIA pedestrian datasets.

Method	TUD	INRIA
$M = 256$	62.7	92.6
$M = 512$	**67.1**	**94.4**

Note: The bold values denote the best detection performances in terms of AP.

Table 4 Performance comparison in terms of voting strategies on the TUD Brussels and INRIA pedestrian datasets.

Method	TUD	INRIA
Uniform voting	63.1	91.5
Weighted voting	**67.1**	**94.4**

Note: The bold values denote the best detection performances in terms of AP.

4.3.5 *Performance of the weighted voting strategy*

As for the weighted voting strategy (Sec. 3.4), we used the LLC coefficients instead of uniform weights as voting weights on codewords. The codebook size was fixed at 512. The parameter K of LLC was fixed at 20. As shown in Table 4, the APs of the weighted voting were 4.0% and 2.9% higher, respectively, than the uniform voting on the INRIA pedestrian and TUD Brussels datasets.

4.3.6 *Effectiveness of the CRF model using the deep convolutional features*

To investigate the effectiveness of the CRF model in detecting pedestrians using the deep convolutional features, we capture contextual relationships on the high-quality object candidates provided by the method RPN + BF.[36] The region of interest (RoI) features of size $512 \times 7 \times 7$ are naturally extracted from the object candidates in the feature maps as in Ref. 36. An object candidate is regarded as a node in the CRF model within a fully connected form. The unary potential of the CRF model is the cost of the confidence score on an object candidate outputted by RPN + BF, which denotes the inverse likelihood of an object candidate taking the label of pedestrian. The pairwise potential relies on the RoI features of a pair of object candidates, which measures the cost of similar object candidates with different labels (e.g., the binary labels, pedestrian, and background) as in Refs. 48 and 49. We feed the RoI features of object candidates of all test images into the CRF model. Finally, the marginal probability distributions of all object candidates can be simultaneously obtained using the mean field inference in the CRF model. The PR curves are obtained by utilizing the marginal probabilities (as the confidence scores) of the pedestrian label, rather than utilizing the initial confidence scores provided by RPN + BF. In Fig. 5, it can be observed that the CRF model achieved APs of 98.7% and 93.2% on the INRIA and Caltech datasets, respectively, which obtains improvements of 1.3% and 2.2% over the RPN + BF.

5 Conclusion

In this work, we propose a pedestrian detection method that integrates context modeling and weighted voting strategy in a unified Hough transform framework. The noisy votes from background patches can be reduced by exploiting contextual information on image patches in an image. The coding coefficients based on the optimized codebook contribute to casting highly balanced votes in the Hough image. The experimental results on the INRIA pedestrian, TUD Brussels, and Caltech pedestrian datasets demonstrated the effectiveness of the proposed method compared with other Hough transform-based methods. In future studies, we intend to exploit contextual information among multiple images for pedestrian detection since the contextual information that we try to exploit in this work is only from a single image.

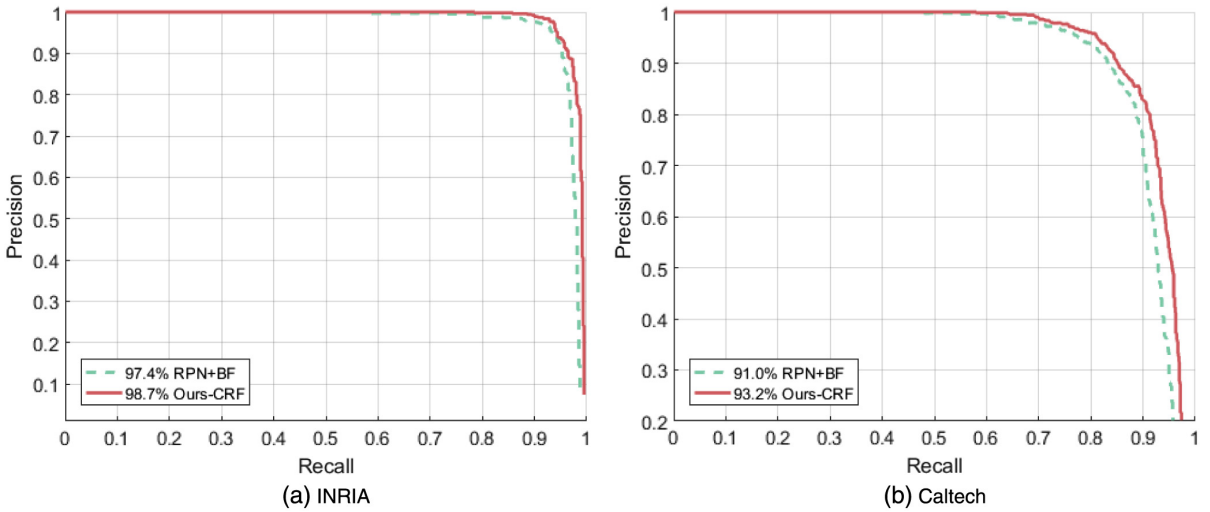

Fig. 5 Detection performance comparisons on the (a) INRIA and (b) Caltech pedestrian datasets according to the reasonable setting. Best viewed in color.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China with Grant Nos. 61375008 and 61673274.

References

1. B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection with interleaved categorization and segmentation," *Int. J. Comput. Vision* **77**(1–3), 259–289 (2008).
2. J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *Decision Forests for Computer Vision and Medical Image Analysis*, A. Criminisi and J. Shotton, Eds., pp. 143–157, Springer, London (2013).
3. J. Gall et al., "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2188–2202 (2011).
4. O. Barinova, V. Lempitsky, and P. Kholi, "On detection of multiple object instances using Hough transforms," *IEEE Trans. Software Eng.* **34**(9), 1773–1784 (2012).
5. T. Wang, X. He, and N. Barnes, "Learning structured Hough voting for joint object detection and occlusion reasoning," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1790–1797 (2013).
6. C. R. Cabrera and R. J. Lpez-Sastre, "Because better detections are still possible: multi-aspect object detection with boosted Hough forest," in *British Machine Vision Conf.* (2015).
7. X. Lou et al., "Invariant Hough random ferns for RGB-D-based object detection," *Opt. Eng.* **55**(9), 091403 (2016).
8. F. Milletari et al., "Hough-CNN: deep learning for segmentation of deep brain regions in MRI and ultrasound," *Comput. Vision Image Understanding* **164**, 92–102 (2017).
9. Y. Liu et al., "A novel rotation adaptive object detection method based on pair Hough model," *Neurocomputing* **194**, 246–259 (2016).
10. Y. Liu et al., "Soft Hough forest-ERTs: generalized Hough transform based object detection from soft-labelled training data," *Pattern Recognit.* **60**, 145–156 (2016).
11. X. He, R. S. Zemel, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, II-695–II-702 (2004).
12. T. Toyoda and O. Hasegawa, "Random field model for integration of local information and global information," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(8), 1483–1489 (2008).
13. J. Shotton et al., "TextonBoost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *Int. J. Comput. Vision* **81**(1), 2–23 (2009).
14. P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," *Adv. Neural Inf. Process. Syst.* 109–117 (2011).
15. L. C. Chen et al., "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *IEEE Int. Conf. on Learning Representations (ICLR)*, IEEE (2015).
16. A. Quattoni et al., "Hidden conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(10), 1848–1852 (2007).
17. M. H. Yang and J. Yang, "Top-down visual saliency via joint CRF and dictionary learning," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 2296–2303 (2012).
18. S. Kumar and M. Hebert, "Discriminative random fields," *Int. J. Comput. Vision* **68**(2), 179–201 (2006).
19. J. Wang et al., "Locality-constrained linear coding for image classification," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 3360–3367 (2010).
20. J. Yang et al., "Linear spatial pyramid matching using sparse coding for image classification," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 1794–1801 (2009).
21. N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 886–893, IEEE (2005).
22. P. F. Felzenszwalb et al., "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(9), 1627–1645 (2010).
23. P. Dollr, S. J. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *British Machine Vision Conf. (BMVC)*, Vol. 2, p. 7 (2010).
24. W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 3258–3265, IEEE (2012).
25. P. Dollár, R. Appel, and W. Kienzle, "Crosstalk cascades for frame-rate pedestrian detection," *Lect. Notes Comput. Sci.* **7573**, 645–659 (2012).
26. R. Benenson et al., "Seeking the strongest rigid detector," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3666–3673 (2013).
27. J. Yan et al., "Robust multi-resolution pedestrian detection in traffic scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3033–3040, IEEE (2013).
28. X. Ren and D. Ramanan, "Histograms of sparse codes for object detection," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3246–3253 (2013).
29. B. Hariharan, C. Zitnick, and P. Dollár, "Detecting objects using deformation dictionaries," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1995–2002 (2014).
30. P. Dollar et al., "Fastest feature pyramids for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8), 1532–1545 (2014).
31. B. C. Ko, J. E. Son, and J. Y. Nam, "View-invariant, partially occluded human detection in still images using part bases and random forest," *Opt. Eng.* **54**(5), 053113 (2015).
32. C. Toca, M. Ciuc, and C. Patrascu, "Normalized autobinomial Markov channels for pedestrian detection," in *BMVC*, pp. 175.1–175.13 (2015).
33. A. Angelova et al., "Real-time pedestrian detection with deep network cascades," in *BMVC*, Vol. 2, p. 4 (2015).
34. A. Verma et al., "Pedestrian detection via mixture of CNN experts and thresholded aggregated channel features," in *Proc. of the IEEE Int. Conf. on Computer Vision Workshops*, pp. 555–563 (2015).
35. Y. Tian et al., "Pedestrian detection aided by deep learning semantic tasks," in *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2015).
36. L. Zhang et al., "Is faster R-CNN doing well for pedestrian detection?" *Lect. Notes Comput. Sci.* **9906**, 443–457 (2016).
37. J. Li et al., "Scale-aware fast R-CNN for pedestrian detection," *IEEE Trans. Multimedia* **20**, 985–996 (2017).
38. X. Du et al., "Fused DNN: a deep neural network fusion approach to fast and robust pedestrian detection," in *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pp. 953–961, IEEE (2017).
39. G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection and segmentation," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 4960–4969, IEEE (2017).
40. S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 2169–2178, IEEE (2006).
41. F. Bach, J. Mairal, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(4), 791–804 (2012).
42. G. Csurka et al., "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision ECCV*, Vol. 44, No. 247, pp. 1–22 (2004).
43. Z. Yang and H. Xiong, "Computing object-based saliency via locality-constrained linear coding and conditional random fields," *Visual Comput.* **33**(11), 1403–1413 (2017).
44. M. Szummer, P. Kohli, and D. Hoiem, "Learning CRFs using graph cuts," *Lect. Notes Comput. Sci.* **5303**, 582–595 (2008).
45. T. Joachims, T. Finley, and C. N. J. Yu, "Cutting-plane training of structural SVMs," *Mach. Learn.* **77**(1), 27–59 (2009).
46. J. Hosang et al., "Taking a deeper look at pedestrians," in *Proc. of the IEEE Conf. on Computer Vision and Pattern* (2015).
47. Y. Tian et al., "Deep learning strong parts for pedestrian detection," in *IEEE Int. Conf. on Computer Vision*, pp. 1904–1912, IEEE (2016).
48. Z. Hayder, M. Salzmann, and X. He, "Object co-detection via efficient inference in a fully-connected CRF," *Lect. Notes Comput. Sci.* **5303**, 330–345 (2014).
49. Z. Hayder, X. He, and M. Salzmann, "Structural kernel learning for large scale multiclass object co-detection," in *IEEE Int. Conf. on Computer Vision (ICCV)*, pp. 2632–2640, IEEE (2015).

Linfeng Jiang received his BS degree in computer science from Chongqing University, Chongqing, China, in 2005, and his MS degree in computer science from Kunming University of Science and Technology, Kunming, China, in 2011. Currently, he is working toward his PhD in the Department of Automation, Shanghai Jiao Tong University (SJTU), Shanghai, China. He is interested in computer vision and probabilistic graphical theory for context modeling.

Huilin Xiong received his BSc and MSc degrees in mathematics from Wuhan University, Wuhan, China, in 1986 and 1989, respectively. He received his PhD in pattern recognition and intelligent control from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology, Wuhan, China, in 1999. He joined SJTU, Shanghai, China, in 2007, and currently, he is a professor in the Department of Automation, SJTU. His research interests include pattern recognition, machine learning, and bioinfomatics.