Switchable Deep Network for Pedestrian Detection

Ping Luo[1,3,*] Yonglong Tian[1,*] Xiaogang Wang[2] Xiaoou Tang[1,3]
[1]Department of Information Engineering, The Chinese University of Hong Kong
[2]Department of Electronic Engineering, The Chinese University of Hong Kong
[3]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences
pluo.lhi@gmail.com yltian@ie.cuhk.edu.hk xgwang@ee.cuhk.edu.hk xtang@ie.cuhk.edu.hk

Abstract

In this paper, we propose a Switchable Deep Network (SDN) for pedestrian detection. The SDN automatically learns hierarchical features, salience maps, and mixture representations of different body parts. Pedestrian detection faces the challenges of background clutter and large variations of pedestrian appearance due to pose and viewpoint changes and other factors. One of our key contributions is to propose a Switchable Restricted Boltzmann Machine (SRBM) to explicitly model the complex mixture of visual variations at multiple levels. At the feature levels, it automatically estimates saliency maps for each test sample in order to separate background clutters from discriminative regions for pedestrian detection. At the part and body levels, it is able to infer the most appropriate template for the mixture models of each part and the whole body. We have devised a new generative algorithm to effectively pretrain the SDN and then fine-tune it with back-propagation. Our approach is evaluated on the Caltech and ETH datasets and achieves the state-of-the-art detection performance.

1. Introduction

Pedestrian detection is an important topic in computer vision [5, 30, 9, 36, 34]. This problem is particularly challenging because pedestrian images undergo large variations of visual appearance due to the changes of poses, viewpoints, clothing, lighting, and resolutions. Background clutters in a detection window also confuse the detectors. Some examples are shown in Fig.1 (a).

Many pedestrian detectors [5, 34, 36, 8, 17, 11] have been developed to address these challenges. They extract manually designed features, such as HOG [5] and Haar-like descriptors [34] or their combinations [36, 18], from images, and then employ classifiers such as boosting [8], SVM [5], and structure SVM [36] to decide whether a detection window should be classified as a pedestrian. Hierarchical deformable part-based models (DPM) [40, 17, 11] are proposed to handle moderate pose variation. In order to handle more complex and larger variations, a mixture of templates is learned for each body part [2, 40]. Such templates (e.g., poselets [2]) are learned through clustering pose annotations and region appearance.

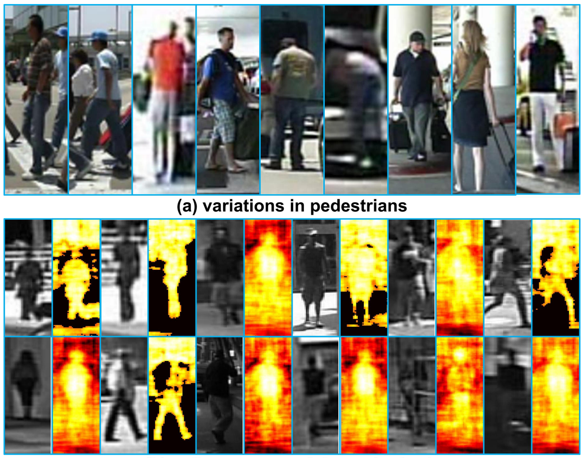

(a) variations in pedestrians
(b) saliency maps of the entire body

Figure 1. Pedestrian detection is challenging due to background clutter, poses, and large variations of appearance of the upper- and lower-body, as shown in (a). It is hard to learn a single model to represent each body part or the whole body. Background clutter also confuses detectors. SDN learns hierarchical features, salience maps, and mixture representations of the entire body and different body parts. The saliency maps that separate the background clutters and the discriminative regions are shown in (b).

In recent years, deep learning has been applied to pedestrian detection and achieved promising results [30, 24, 25]. Instead of using handcrafted features, it can automatically learn features in an unsupervised or supervised fashion, such as restricted Boltzmann machine (RBM) [12], and discriminative RBM [13]. They are often stacked into multiple layers so as to map the raw data into gradually higher-level representations [15, 31, 30]. Then, the entire network is fine-tuned with label information and the top layer output is often adopted as features to train classifiers. However, the hierarchical representations learned by deep models do not have semantic meanings (such as the body

[*] indicates equal contribution.

parts of head-shoulder, upper-body, and lower-body) as in previous hierarchical deformable part-based models [40, 17, 11, 16, 37]. Ouyang and Wang [25] extend DPM to a deep model by learning feature representations and jointly optimizing the key components of DPM. However, they did not explicitly model mixture of templates for each body parts as in [2, 40] and did not depress the influence of background clutters.

We propose a novel Switchable Deep Network (SDN) for pedestrian detection. The SDN automatically learns hierarchical feature representations that correspond to body parts and the whole body. The key contribution of the model is that it introduces a new Switchable Restricted Boltzmann Machine (SRBM) to explicitly model the complex mixture of visual appearance at multiple levels. SRBM is used to build switchable layers added into the hierarchy of the SDN. At each feature level, SRBM estimates saliency maps (indicating a pixel is on the background or a pedestrian) for each test sample. For instance, in the root layer, the saliency map separates background clutter from discriminative regions for pedestrian detection. Some examples are shown in Fig.1 (b). In a part layer, the saliency map also helps to localize each part in the same way. In addition, our deep model learns a mixture of templates for each part to represent it in different views and poses. SRBM can infer the most appropriate template for each part or the whole body. Since all the body parts and their templates have semantic meanings, they are initialized through clustering image regions. A new generative algorithm is devised to effectively pre-train the SDN and then fine-tune it with back-propagation.

In summary, our work makes three key contributions. First, we propose a unified deep model to jointly learn features, saliency maps, and mixture representations of the whole body and different body parts in a hierarchy. This makes it possible to maximize the strengths of all of the components. Second, we enrich the traditional convolutional neural network (CNN) by introducing a switchable layer built with a new switchable restricted Boltzmann machine. This layer depresses background clutters by estimating saliency maps and handles complex pedestrian appearance variations with mixture of part templates. Our third contribution is to propose a EM-like algorithm to pre-train the switchable layer. With this algorithm, some hidden variables can be estimated directly in the E-step without Gibbs sampling, so that it can reduce the computation time compared with the conventional methods.

1.1. Related Works

We review previous works in three aspects as follows.

Feature Learning. Recent works on deep neural networks such as [15, 12, 30, 13, 42, 19, 23, 41, 20, 32, 33] are capable to learn features in terms of complex object categories. For instance, [12, 15, 30, 19] unsupervisedly pre-trained the networks in a layerwise manner. Moreover, [13, 41, 20] layerwisely pre-trained the network with supervised information and showed superior results. In this paper, we learn discriminative features with a new pre-training strategy, which incorporates label information.

Hierarchical Deformable Models. DPM is one of the widely used methods [11]. It learns a two-layer hierarchy of root and part templates using a weakly-supervised latent SVM. Zhu *et al.* [40] and Lin *et al.* [17] extended this hierarchy with more layers and the mixture of templates. Bourdev and Malik [2] modeled complex variations of part appearance with poselets, which are a set of templates learned through clustering. However, these methods rely on the hand-crafted features, the discriminative capacities of which are not optimized for pedestrian detection.

Mixture of Deep Models. Recent studies [21, 4, 31] have shown that a mixture of deep models works better than singleton. Nair and Hinton [21] proposed the mixture of RBMs to learn features from raw pixels, by including a gating variable to determine which RBM should be activated. Sohn *et al.* [31] partitioned the learned features to two components: relevant features on the foreground and the irrelevant features on the background. Ciresan *et al.* [4] separated the data into several groups according to some domain knowledge (multi-scales, for example), and then constructed an ensemble of deep neural networks for image classification. Unlike the existing works that focus either on learning features or constructing an ensemble of models, the mixture of the switchable layer in the SDN is designed to model high-level object hierarchy as well as saliency maps, and is jointly trained and optimized with the features extracted by the convolutional layers. Therefore, it is more robust to account for pedestrian variations and enables us to incorporate more domain knowledge (such as the design of body parts and initialization of part templates) into the network for object detection.

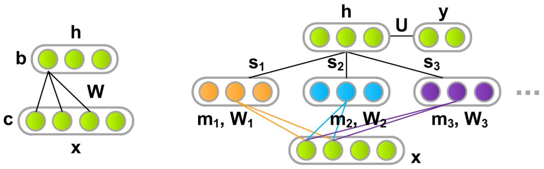
2. Switchable Restricted Boltzmann Machine

The proposed switchable restricted Boltzmann machine (SRBM) is a key building block in the SDN to model the hierarchical feature representations and the mixtures of body parts and entire body for pedestrian detection. We first review the regular RBM in Sec.2.1 and then introduce SRBM and its pre-training method in Sec.2.2.

A graphical model with both observed and hidden variables can be formulated as follows

$$p(\mathbf{V};\Theta) = \frac{1}{Z}\sum_{\mathbf{H}} \exp\{-E(\mathbf{V},\mathbf{H})\}, \tag{1}$$

where $\mathbf{V},\mathbf{H}$ are the sets of observed and hidden variables, and Z is the normalizing constant. $E(\mathbf{V},\mathbf{H})$ is an energy



(a) RBM (b) Switchable RBM

Figure 2. The architectures of the regular RBM and the SRBM.

function and Θ denotes a set of parameters that can be optimized using maximum likelihood estimation, where the gradient can be computed by

$$\frac{\partial \log p(\mathbf{V})}{\partial \Theta} = -\mathbb{E}_{p(\mathbf{H}|\mathbf{V})}[\frac{\partial E(\mathbf{V},\mathbf{H})}{\partial \Theta}] + \mathbb{E}_{p(\overline{\mathbf{V}},\overline{\mathbf{H}})}[\frac{\partial E(\overline{\mathbf{V}},\overline{\mathbf{H}})}{\partial \Theta}]. \quad (2)$$

$\frac{\partial E(\mathbf{V},\mathbf{H})}{\partial \Theta}$ is the partial derivative of the parameters. The first term in Eq.(2) calculates the expectation of the hidden variables given the observed data and the second term calculates the expectation of the joint probability under the current model, which has to be inferred by sampling. For example, [12] approximated the gradient of Eq.(2) by Gibbs sampling.

2.1. Restricted Boltzmann Machine

RBM is a Markov Random Field that defines on both the observed and hidden variables. In the traditional RBM as shown in Fig.2 (a), $V^{rbm} = \{\mathbf{x}\}$ and $H^{rbm} = \{\mathbf{h}\}$ are the input and output (hidden features) of the layer, respectively, and $\Theta = \{\mathbf{W}, \mathbf{c}, \mathbf{b}\}$ contains the weight matrices and the bias vectors of the input and output. Note that bold letters indicate vector or matrix. Its energy function is written as

$$E(\mathbf{x},\mathbf{h}) = -\mathbf{c}^T\mathbf{x} - \mathbf{b}^T\mathbf{h} - \mathbf{x}^T\mathbf{W}\mathbf{h}, \quad (3)$$

where the first two terms can be considered as the unary potentials as in MRF, while the last term is the pairwise potential. The probabilities of one set of variables given the other are conditional independent and the conditional probabilities for Gibbs sampling are as follows

$$\begin{aligned} p(\mathbf{h}=1|\mathbf{x}) &= \tau(\mathbf{W}\mathbf{x}+\mathbf{b}), \quad (4)\\ p(\mathbf{x}=1|\mathbf{h}) &= \tau(\mathbf{W}^T\mathbf{h}+\mathbf{c}), \quad (5)\end{aligned}$$

where $\tau(\mathbf{x}) = 1/(1+\exp(-\mathbf{x}))$ is the sigmoid function.

2.2. Switchable RBM

One of the contributions of this study is to extend RBM by modeling the mixture and saliency maps using SRBM. As shown in Fig.2 (b), $V^{srbm} = \{\mathbf{x},\mathbf{y}\}$ and $H^{srbm} = \{\mathbf{h},\mathbf{m},\mathbf{s}\}$, where $\mathbf{y},\mathbf{m},\mathbf{s}$ denote the labels, the saliency maps, and the switch variables indicating which component in the mixture is activated. We employ both the input data and the labels as observed variables other than only using the data as in RBM, because supervised information can improve classification performance [13]. The energy function is formulated as

$$\begin{aligned} E(\mathbf{x},\mathbf{y},\mathbf{h},\mathbf{s},\mathbf{m};\Theta) = &-\sum_{k=1}^{K} s_k\mathbf{h}_k^T(\mathbf{W}_k(\mathbf{x}\circ\mathbf{m}_k)+\mathbf{b}_k)\\ &-\sum_{k=1}^{K} s_k\mathbf{c}_k^T(\mathbf{x}\circ\mathbf{m}_k) - \mathbf{y}^T\mathbf{U}\sum_{k=1}^{K} s_k\mathbf{h}_k - \mathbf{d}^T\mathbf{y}, \end{aligned} \quad (6)$$

in which K indicates the number of components in the mixture and $\Theta = \{\mathbf{W},\mathbf{b},\mathbf{c},\mathbf{U},\mathbf{d}\}$, where $\mathbf{U}$ is a fully-connect weight matrix to transform the features to labels and $\mathbf{d}$ is the bias vector of the label. The switch variable $s_k \in [0,1], \sum_{k=1}^{K} s_k = 1$ indicates which component is activated. In the SRBM, the output features are the linear combination of the hidden features of different components; that is $\mathbf{h} = \sum_{k=1}^{K} s_k\mathbf{h}_k$ as shown in Eq.(6). For each component, $\mathbf{m}_k \in [0,1]^{n\times m}$ is the saliency map representing the discriminative regions of the pedestrian. As shown in Fig.1 (b), the value of 0 indicates background and the value of 1 indicates discriminative regions. Element-wise product of two vectors is denoted with $\circ$.

Similar to RBM, the observed and hidden variables are conditionally independent given the others, and the conditional probabilities can be derived as below.

$$\begin{aligned} &p(\mathbf{h}_k=1|\mathbf{x},\mathbf{y},\mathbf{s},\mathbf{m}) = \tau(s_k(\mathbf{W}_k(\mathbf{x}\circ\mathbf{m}_k)+\mathbf{b}_k+\mathbf{U}^T\mathbf{y})),\\ &p(\mathbf{x}=1|\mathbf{h},\mathbf{s},\mathbf{m}) = \tau(\sum_{k=1}^{K} s_k\mathbf{m}_k(\mathbf{W}_k^T\mathbf{h}_k+\mathbf{c}_k)),\\ &p(\mathbf{y}=1|\mathbf{h},\mathbf{s}) = \tau(\mathbf{U}(\sum_{k=1}^{K} s_k\mathbf{h}_k)+\mathbf{d}). \end{aligned} \quad (7)$$

Eq.(7) shows that the sampling of $\mathbf{h},\mathbf{x},\mathbf{y}$ can be derived in a similar way as RBM. Moreover, the conditional probabilities of $\mathbf{m},\mathbf{s}$ are

$$\begin{aligned} &p(\mathbf{m}_k=1|\mathbf{x},\mathbf{h},\mathbf{s}) = \tau(s_k\mathbf{x}(\mathbf{W}_k^T\mathbf{h}_k+\mathbf{c}_k)),\\ &p(s_k=1|\mathbf{x},\mathbf{y},\mathbf{h},\mathbf{m}) = \frac{1}{Z}\exp\{\mathbf{h}_k^T(\mathbf{W}_k(\mathbf{x}\circ\mathbf{m}_k)+\mathbf{b}_k)\\ &\qquad +\mathbf{c}_k^T(\mathbf{x}\circ\mathbf{m}_k)+\mathbf{y}^T\mathbf{U}\mathbf{h}_k\}, \end{aligned} \quad (8)$$

where the saliency map of the k-th component can be considered as the correlation between the original input $\mathbf{x}$, and the recovered input $\mathbf{W}_k^T\mathbf{h}_k + \mathbf{c}_k$ by this component. High correlation indicates high saliency. The computation of $\mathbf{s}$ is similar to Eq.(6) and suggests that, if a component has a smaller energy value, it is more likely to be activated.

However, the optimization procedure for SRBM has comparatively high computational cost because the calculation of Eq.(2) must alternately sample five different kinds of variables as in Eq.(7) and (8). We simplify the training procedure using an EM-like algorithm by considering the switch variables as pseudo-observed variables. In this case, we can estimate their values directly in the E-step, and then update the parameters in the M-step using Eq.(2) by sampling the other four variables. This strategy saves 20 percent of the pre-training time.

Pseudo-observed SRBM. The joint probability of $\mathbf{x}$, $\mathbf{y}$, and the pseudo-observed variables $\mathbf{s}$ is written as

$$p(\mathbf{x}, \mathbf{y}, \mathbf{s}; \Theta) \propto p(\mathbf{x}, \mathbf{y}|\mathbf{s}; \Theta) p(\mathbf{s}). \tag{9}$$

The prior is specified by $p(\mathbf{s}) = \prod_{k=1}^{K} \lambda_k^{s_k}$, where $\lambda_k = \frac{1}{N}\sum_{n=1}^{N} s_{nk}$ is the mixing coefficient indicating the fraction of training samples assigned to the k-th components. $p(\mathbf{x}, \mathbf{y}|\mathbf{s})$ can be defined by integrating over $\mathbf{h}$ and $\mathbf{m}$,

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}|\mathbf{s}) &\propto \frac{1}{Z} \sum_{\mathbf{h},\mathbf{m}} e^{-E(\mathbf{x},\mathbf{y},\mathbf{h},\mathbf{s},\mathbf{m})} \\ &\propto \frac{1}{Z} e^{\mathbf{d}^T\mathbf{y}} \prod_{k=1}^{K} \sum_{\mathbf{h}_k} e^{s_k(\mathbf{y}^T\mathbf{U}+\mathbf{b}^T)\mathbf{h}_k} \cdot \\ &\quad \prod_{i=1}^{|\mathbf{x}|} (1 + e^{s_k(\mathbf{W}_{k,i*}^T \mathbf{h}_k + c_{ki}) x_i}) \\ &\propto \frac{1}{Z} e^{\mathbf{d}^T\mathbf{y}} \prod_{k=1}^{K} \prod_{i=0}^{|\mathbf{x}|} \sum_{\mathbf{h}_k} G_{ik}, \end{aligned} \tag{10}$$

where the energy function $E(\cdot)$ is given in Eq.(6). The second equation in (10) integrates $\mathbf{m}$ and the last one sums over both $\mathbf{m}, \mathbf{h}$, where G_{ik} expresses the expansion of the product of $|\mathbf{x}|$ binomials. For example, G_{0k} and G_{1k} are $e^{s_k(\mathbf{y}^T\mathbf{U}+\mathbf{b}^T)\mathbf{h}_k}$ and $e^{s_k(\mathbf{y}^T\mathbf{U}+\mathbf{b}^T)\mathbf{h}_k} \cdot (e^{s^k(\mathbf{W}_{k,1*}^T \mathbf{h}_k + c_{k1}) x_1} + e^{s_k(\mathbf{W}_{k,2*}^T \mathbf{h}_k + c_{k2}) x_2} + ... + e^{s_k(\mathbf{W}_{k,i*}^T \mathbf{h}_k + c_{ki}) x_i} + ...)$, respectively. Thus, the integration of G_{0k} over $\mathbf{h}_k$ is $\prod_{j=1}^{|\mathbf{h}|}(1 + e^{s_k(\mathbf{y}^T\mathbf{U}_{*j}+b_j)})$. More details are provided in the supplementary material[1]. Combining Eq.(9) and (10) with the Bayes rules, the posterior distribution becomes $p(\mathbf{s}|\mathbf{x}, \mathbf{y}) = \frac{1}{Z}\prod_{k=1}^{K} \lambda_k^{s_k} \sum_{\mathbf{h},\mathbf{m}} \exp\{-E(\mathbf{x}, \mathbf{y}, \mathbf{h}, \mathbf{s}, \mathbf{m})\}$, from which we can estimate the value of s_k.

Implementation details. The pre-training contains two steps: (1) initialization and (2) EM optimization. In the first step, we start by grouping the input to K components using k-means. As many variants of RBM [13], we then train a regular RBM for each component to initialize the weight matrixes. To obtain discriminative power and save computation time, we retain the weights related to the discriminative hidden features and discard the others by using t-Test [39]. In the second step, the EM algorithm proceeds as follows. We evaluate $\mathbf{s}$ in the E-step and keep $\mathbf{s}$ fixed and maximize the log likelihood $\log p(\mathbf{x}, \mathbf{y}|\mathbf{s})$ with respect to Θ in the M-step, which is similar to Eq.(2) and can be calculated following contrastive divergence [12]. The details are given in Alg.1.

Algorithm 1 Pre-training pseudo-observed SRBM

```
Input: input ${\mathbf{x}}$ and labels ${\mathbf{y}}$;
Output: $\Theta = \{\mathbf{W}, \mathbf{U}, \mathbf{b}, \mathbf{c}, \mathbf{d}\}$;
group input data ${\mathbf{x}}$ into $K$ components using k-means;
train RBM to initialize $\mathbf{W}, \mathbf{b}, \mathbf{c}$ for each component and
    initialize $\mathbf{U}, \mathbf{d}, \lambda$ randomly; prune $\mathbf{W}$ by t-Test;
if not stopping criterion loop
  E-step: estimate $\mathbf{s}$ for each $\mathbf{x}$;
  M-step:
    for a minibatch $\mathbf{x}$ do
      perform gibbs sampling for $t$ steps according to
        Eq.(7) and (8) to obtain $\mathbf{x}^0, \mathbf{y}^0, \mathbf{h}^0, \mathbf{m}^0$, and
        $\overline{\mathbf{x}}^t, \overline{\mathbf{y}}^t, \overline{\mathbf{h}}^t, \overline{\mathbf{m}}^t$;
      $\nabla\Theta \leftarrow \mathbb{E}[\frac{\partial E(\mathbf{x}^0,\mathbf{y}^0,\mathbf{h}^0,\mathbf{m}^0,\mathbf{s})}{\partial\Theta}] - \mathbb{E}[\frac{\partial E(\overline{\mathbf{x}}^t,\overline{\mathbf{y}}^t,\overline{\mathbf{h}}^t,\overline{\mathbf{m}}^t,\mathbf{s})}{\partial\Theta}]$;
    end
    update parameters, $\Theta \leftarrow \Theta + \eta\nabla\Theta$;
    update mixing coefficients, $\lambda_k = \frac{1}{N}\sum_{n=1}^{N} s_{nk}$;
end loop
```

3. Switchable Deep Network (SDN)

We stack a convolutional layer, four switchable layers (that is, modeled with SRBM), and one logistic regression layer into the SDN for pedestrian detection. As shown in Fig.3, the convolutional layer learns to extract low- and mid-level features, the switchable layers model high-level mixture representations and salience maps of the entire body and different body parts (head-shoulder, upper-body, and lower-body), and the logistic regression layer predicts labels. This architecture is designed for pedestrian detection. More layers can be added to handle more complex object hierarchies.

The input image data $\mathbf{x}^0$ (Fig.3 (a)) have six channels, each of which is in the size of 108 × 36. The first three channels are obtained by resizing the bounding box centered on the pedestrian with three different scales and then extract the Y-channels of these three images in the YUV color spaces. The last three channels are the edge maps of the first three channels by using Sober edge detector. This is to encourage the SDN to learn features with multi-scales and boundary cues.

As shown in Fig.3 (a), the convolutional layer outputs 64 channels by learning 64 filters, each with a size of $9 \times 9 \times 6$. This layer can be formulated as below

$$\mathbf{x}_j^1 = \tanh^{abs}(\sum_{i=1}^{I} \mathbf{W}_i^1 * \mathbf{x}_i^0 + b_j^1), \tag{11}$$

[1] http://mmlab.ie.cuhk.edu.hk/publications.html

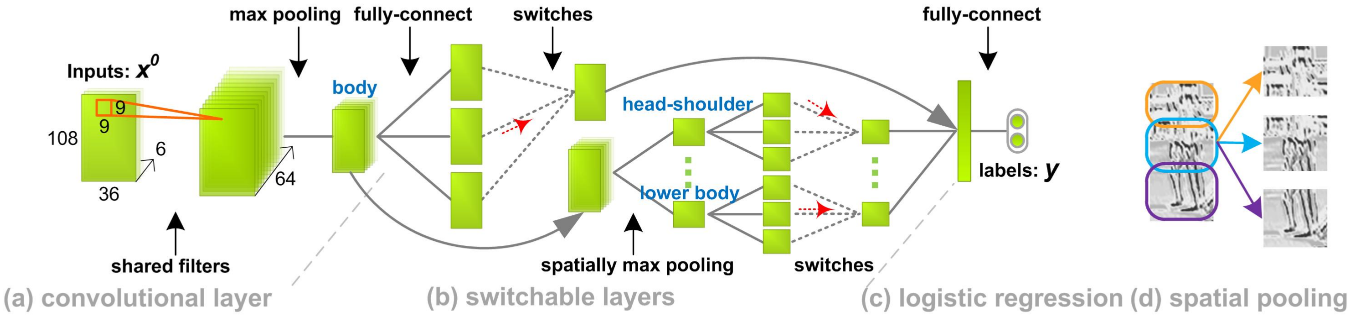


Figure 3. Architecture of the SDN. It stacks three types of layers, including a convolutional layer (a) at the bottom to extract low- and mid-level features, four switchable layers (b) in the middle for high-level mixture representations (*i.e*., body, head-shoulder, upper-body, and lower-body), and a logistic regression layer (c) on the top for label prediction. The spatial max pooling in (b) is illustrated in (d). It divides the whole body into three sub-regions and pass their feature maps to next layers.

where $\tanh^{abs}(\cdot) = |tanh(\cdot)|$ is the absolute values of the hyperbolic tangent function, $*$ indicates convolution, and $i = 1...6$ and $j = 1...64$ are the indices of the input and output channels, respectively. $\mathbf{W}^1$ and $\mathbf{b}^1$ are the filter matrixes and bias vector. The output $\mathbf{x}^1$ are then sub-sampled by a max pooling layer to obtain more compact representation.

As shown in Fig.3 (b), we stack four switchable layers as a hierarchy to model the decomposition of pedestrians, including a root layer for body, three sub-layers for head-shoulder, upper-body, and lower-body, respectively. Each switchable layer is a mixture of K components ($K = 10$ in our experiment), each of which connects to the input using a fully-connect weight matrix that is to capture the global pose or view of pedestrians. The l-th layer can be computed as

$$\mathbf{x}^l = \sum_{k=1}^{10} s_k^l \tanh^{abs}(\mathbf{W}_k^l(\mathbf{x}^{l-1} \circ \mathbf{m}_k^l) + \mathbf{b}_k^l)), \tag{12}$$

where k is the index of components and $\circ$ denotes the element-wise product. $\mathbf{m}_k$ denotes the saliency map. s_k denotes the switch variable, which serves as a gate and outputs the features of the most informative component. One possible output of the SDN is illustrated by the red arrows of Fig.3 (b). Both $\mathbf{s}$ and $\mathbf{m}$ are the hidden variables, the values of which vary for different samples and have to be inferred during training and testing. Furthermore, the spatially max pooling layer partitions the learned features of the entire body into three parts, as shown in Fig.3 (d). Such partition works fine for pedestrian detection and the number of partitions can be changed to deal with the other object categories.

As shown in Fig.3 (c), the logistic regression layer predicts the label by concatenating the output of all the switchable layers as input,

$$\mathbf{y} = \tau(\mathbf{W}^L \mathbf{x}^{L-1} + \mathbf{b}^L), \tag{13}$$

where W^L is a fully-connect weight matrix.

3.1. Pre-training and Fine-tuning

The training stage of SDN needs to update a set of filters and infer the hidden variables; that is, switches and saliency maps. This is challenging because there are a large number of such variables. For instance, if we divide the body into three parts and have five components for each part, there are millions of parameters and over $2,000$ hidden variables. Therefore, we have adopted the same scheme as many other deep learning methods have done, which is to pre-train the network in a layerwise manner and then fine-tune all the parameters.

We use Gabor filters to initialize the filters of the convolutional layer, because Gabor filters can capture the boundary shapes of pedestrians. For the four switchable layers in Fig.3 (b), we pre-train them following Alg.1.

As the existing methods [15, 12, 13, 4] have proved, fine-tuning deep networks can improve classification performance. Similarly, we fine-tune all the parameters of the SDN by minimizing the error entropy

$$Err(\mathbf{x}^0; \Theta) = \mathbf{y} \log \bar{\mathbf{y}} + (1 - \mathbf{y}) \log(1 - \bar{\mathbf{y}}), \tag{14}$$

in which $\bar{\mathbf{y}}$ is the predicted label. The parameters are updated using stochastic gradient descent. For instance, we update the weights by $\mathbf{W}_{t+1} \leftarrow \mathbf{W}_t - \eta \frac{\partial Err}{\partial \mathbf{W}_t}$. For the convolutional layers and the logistic regression layer as shown in Fig.3 (a) (c), $\frac{\partial Err}{\partial \mathbf{W}}$ are calculated in the same way as the traditional CNN [14]. For the switchable layer, the gradient of the weight matrix for the k-th component is computed as

$$\frac{\partial Err}{\partial \mathbf{W}_k} = s_k(\mathbf{x}^{l-1} \circ \mathbf{m}_k)\mathbf{e}^{l^T}, \tag{15}$$

which is the outer product of the back-propagation error $\mathbf{e}^l$ and the input of the k-th component. Then, the error $\mathbf{e}^l$ is obtained by a recurrence relation as

$$[\mathbf{e}^l]_i = \left\{ \begin{array}{ll} [\beta^l(1 - \tanh^2(\delta^l))]_i, & [\tanh(\delta^l)]_i > 0 \\ [\beta^l(\tanh^2(\delta^l) - 1)]_i, & otherwise \end{array} \right., \tag{16}$$

where $[\cdot]_i$ denotes the i-th element of a vector and $diag(\cdot)$ is the diagonal matrix. Furthermore, $\delta^l = \mathbf{W}_k^l(\mathbf{x}^{l-1} \circ \mathbf{m}_k^l) + \mathbf{b}^l$ is the output of the l-th layer without the absolute hyperbolic tangent function, and $\beta^l = diag(\mathbf{W}^{l+1^T}\mathbf{e}^{l+1})$. The back-propagation error is computed in this piecewise manner because of the absolute function.

3.2. Inference

In the testing stage, unlike the existing deep learning methods that deal with different samples with the same network architecture, the SDN selects the most appropriate structure for each sample. This is achieved by inferring the hidden variables $\mathbf{s}, \mathbf{m}$. First, the switch variables $\mathbf{s}$ can be estimated based on Eq.(10). Second, we can infer the saliency maps $\mathbf{m}$ as discussed in Eq.(8) given $\mathbf{x}$, $\mathbf{h}$, and $\mathbf{s}$.

4. Experiments

We conduct experiments on the Caltech dataset [9] and the ETH dataset [10]. The former consists of approximately 10 hours video in an urban environment. A total of $350,000$ bounding boxes and $2,300$ pedestrians were annotated. The latter has three testing sequences with a total of 1804 frames. To reduce the computational time, we adopt a simple detector trained with HOG+CSS+SVM to prune the candidate windows at both the training and the testing stages. We keep approximately $60,000$ windows that are not pruned by the detector for training. At the testing stage, SDN takes less than 0.1 second per image after the HOG+CSS+SVM detector has pruned most candidate windows. For both datasets, we strictly follow the criteria proposed in [9] to evaluate the performance, where the log-average miss rate is computed by averaging the miss rates at nine False-Positive-Per-Image (FPPI) rates, which are evenly spaced in log-space in the range from 10^{-2} to 10^0. Moreover, we test on the *reasonable* subsets of both datasets. These subsets are widely used and consider pedestrians with heights larger than 49 pixels according to the ground truth.

We compare with the best-performing methods as suggested by the Caltech and ETH benchmarks[2], which report the top results of these two datasets, including VJ [34], HOG [5], DBN-Isol [24], ACF [6], ACF-Caltech [6], MultiFtr+CSS [35], MultiResC [28], Roerei [1], MOCO [3], MT-DPM [38], ChnFtrs [8], HogLbp [36], Pls [29], CrossTalk [7], LatSVM-V2 [11], MLS [22], ConvNet [30], and UDN [25]. All of these approaches detect pedestrians on static images, like our method, rather than using video motion as additional information. We have also excluded the results of using contextual information. For example, Ouyang and Wang [26] used a two-pedestrian detetor to improve single-pedestrian detection and showed that their two-pedestrian detector can be used to improve any single-pedestrian detectors. In [27] two neighboring pedestrians were jointly detected. Yan *et al.* [38] used a vehicle detector to improve pedestrian detection. These context-based approaches are complementary to ours. The above works use various features, classifiers, deep networks, and context information. We summarize them below.

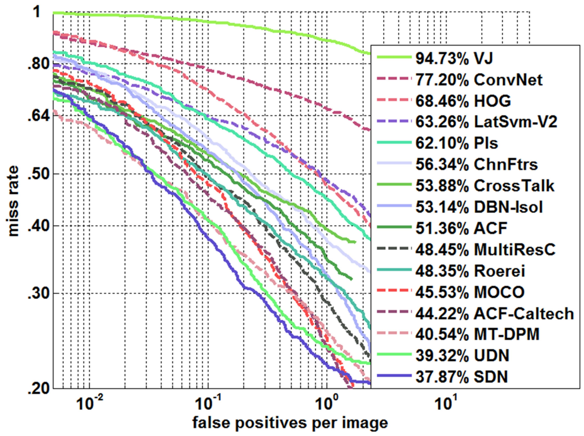


Figure 4. Overall performance of Caltech-Test dataset.

Features: Haar (VJ), HOG (HOG, LatSVM-V2, MT-DPM), LBP (HogLbp), CSS (MultiFtr+CSS); **Classifiers**: latent-SVM (LatSVM-V2, MOCO, MT-DPM), boosting (VJ, ChnFtrs, CrossTalk); **Deep Learning**: DBN-Isol, ConvNet, UDN.

4.1. Performance on the Caltech-Test Dataset

Overall Performance. For evaluation purposes we pre-train and fine-tune the SDN using the Caltech-Train dataset, which is also adopted as training data by the recent best-performing methods, such as [38, 6]. We compare the result with the existing approaches in Fig.4, where SDN achieves the smallest miss rate of 37.87 percent among all the detectors without using context information. SDN outperforms the other two detectors (DBN-Isol and ConvNet) with deep learning by at least 15 percent. DBN-Isol did not learn features and used DBN to infer the visibility status of body parts. ConvNet learned features in an unsupervised way. Our SDN learns low- and mid-level features and high-level mixture representations jointly in a supervised way. SDN also outperforms the methods based on deformable part models such as LatSVM-V2 and MT-DPM, which extracted the HOG features of multiple resolutions, while our method directly learns features from raw pixels in multiple resolutions. It takes three hours to train a SDN (including both pre-training and fine-tuning) on a single NVIDIA GTX 760 GPU.

Effectiveness of Architecture. The switchable layers in the SDN utilize the output of the convolutional layer as input. In fact, they can employ any other hand-crafted features as input. We test different features

[2] http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

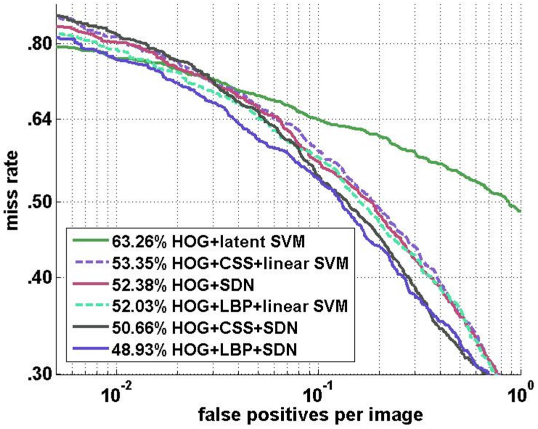


Figure 5. The performance of SDN combined with different hand-crafted features are compared to the existing approaches on the Caltech-Test dataset.

combined with the switchable layers, including HOG, HOG+CSS, and HOG+LBP, and compare our approaches with the existing methods, which are HOG+latent-SVM, HOG+LBP+linear-SVM, and HOG+CSS+linear-SVM. All the above methods employ the same experimental setting. The results are given in Fig.5. For instance, HOG+SDN improves HOG+latent SVM by 11 percent and the HOG+CSS+SDN reduces the miss rate by 3 percent compared to HOG+CSS+linear SVM. In the above settings, we observe that HOG+LBP+SDN achieves the best result that is 48 percent. Since multiple features can improve performance, the difference of miss rates between combining HOG+CSS is smaller than only utilizing HOG features.

Effectiveness of Feature Learning. We evaluate the use of multiple scales of the images as input for feature learning. There are three different combinations: (1) size of the bounding box of the pedestrian multiply by 1.1 (one scale), (2) size of the bounding box multiply by 1.0 and 1.25 (two scales), and (3) size of the bounding box multiply by 1.0, 1.25, and 1.45 (three scales). We separately examine the influence of the Y-channels and the edge maps as introduced in Sec.3. Fig.6 shows the results, which are obtained by directly employing the output of the convolutional layer as features and using logistic regression for classification. We demonstrate that Y-channels are more informative than the edge maps and the use of multiple scales tends to improve the performances of both of them. The best miss rate (43.98 percent) is obtained by using Y-channels in three scales. However, the multi-scales combination of the Y-channels and the edge maps achieves the miss rate of 40.12 percent.

4.2. Performance on the ETH Dataset

We follow the existing approaches [24, 11, 1] that evaluate their methods on the ETH dataset with a common setting, which is to use the INRIA-Train dataset as training data. This is done in order to evaluate the generalization capacity of the SDN. Fig.7 plots the results on the ETH dataset. SDN again achieves the lowest average miss rate. It outperforms the deep learning based methods DBN-Isol and ConvNet by 6.38 percent and 9.64 percent, respectively.

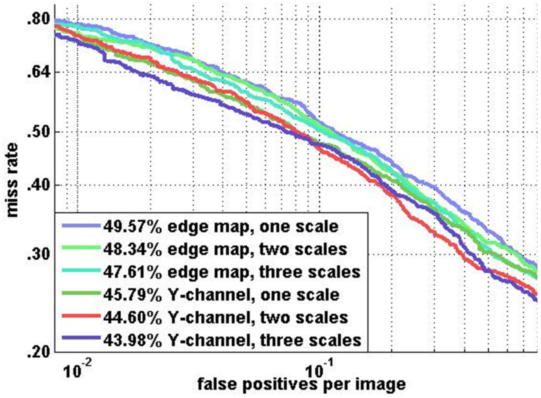


Figure 6. Performance of multiple scales feature learning on the Caltech-Test dataset.

5. Conclusions

In this paper, we have proposed a switchable deep network to model background clutter and complex appearance variations in pedestrian detection. This SDN improves the conventional convolutional neural network by adding multiple switchable layers, which are built with a new switchable restricted Boltzman machine. This new deep model jointly learns hierarchical features, salience maps, and mixture representations of body parts. It achieves state-of-the-art performance on the public benchmark datasets.

6. Acknowledgement

This work is supported by the General Research Fund sponsored by the Research Grants Council of Hong Kong (Project No.CUHK 417110, CUHK 417011, CUHK 429412), National Natural Science Foundation of China (91320101), Shenzhen Basic Research Program (JC201005270350A, JCYJ20120903092050890, JCYJ20120617114614438), and Guangdong Innovative Research Team Program (No.201001D0104648280).

The authors would like to thank Wanli Ouyang and Xingyu Zeng for helpful suggestion, and Yonglong Tian for his main contribution in experiments and part of the derivation of the model.

References

[1] R. Benenson, M. Mathias, T. Tuytelaars, and L. V. Gool. Seeking the strongest rigid detector. *CVPR*, 2013.

[2] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[3] G. Chen, Y. Ding, J. Xiao, and T. Han. Detection evolution with multi-order contextual co-occurrence. *CVPR*, 2013.

[4] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. *CVPR*, 2012.

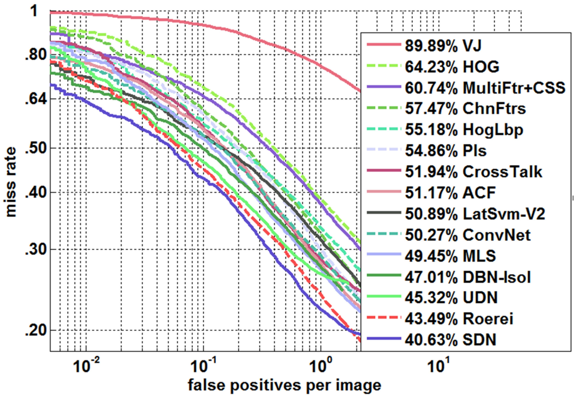


Figure 7. Overall performance on the ETH dataset.

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

[6] P. Dollár, R. Appel, S. Belongie, and P. Perona. Aggregate channel features. *Piotr's Image and Video Matlab Toolbox*, 2013.

[7] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. *ECCV*, 2012.

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. *BMVC*, 2009.

[9] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: an evaluation of the state of the art. *TPAMI*, 34(4):743–761, 2012.

[10] A. Ess, B. Leibe, and L. V. Gool. Depth and appearance for mobile scene analysis. *ICCV*, 2007.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32(9), 2010.

[12] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[13] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. *ICML*, 2008.

[14] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[15] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. *ICML*, 2009.

[16] L. Lin, T. Wu, J. Porway, and Z. Xu. A stochastic graph grammar for compositional object representation and recognition. *Pattern Recognition*, 2009.

[17] Z. Lin and L. S. Davis. Shape-based human detection and segmentation via hierarchical part-template matching. *TPAMI*, 32(4):604–618, 2010.

[18] P. Luo, L. Lin, and H. Chao. Learning shape detector by quantizing curve segments with multiple distance metrics. *ECCV*, 2010.

[19] P. Luo, X. Wang, and X. Tang. Hierarchical face parsing via deep learning. *CVPR*, 2012.

[20] P. Luo, X. Wang, and X. Tang. Pedestrian parsing via deep decompositional neural network. *ICCV*, 2013.

[21] V. Nair and G. Hinton. Implicit mixtures of restricted Boltzmann machines. *NIPS*, 2008.

[22] W. Nam, B. Han, and J. H. Han. Improving object localization using macrofeature layout selection. *ICCV Workshop on Visual Surveillance*, 2011.

[23] W. Ouyang, X. Chu, and X. Wang. Multi-source deep learning for human pose estimation. *CVPR*, 2014.

[24] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. *CVPR*, 2012.

[25] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. *ICCV*, 2013.

[26] W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. *CVPR*, 2013.

[27] W. Ouyang, X. Zeng, and X. Wang. Modeling mutual visibility relationship with a deep model in pedestrian detection. *CVPR*, 2013.

[28] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. *ECCV*, 2010.

[29] W. Schwartz, A. Kembhavi, D. Harwood, and L. Davis. Human detection using partial least squares analysis. *ICCV*, 2009.

[30] P. Sermanet, K. Kavukcuoglu, S. C. Pedestrian, and Y. LeCun. Pedestrian detection with unsupervised multi-stage feature learning. *CVPR*, 2013.

[31] K. Sohn, G. Zhou, C. Lee, and H. Lee. Learning and selecting features jointly with point-wise gated Boltzmann machines. *ICML*, 2013.

[32] Y. Sun, X. Wang, and X. Tang. Hybrid deep learning for face verification. *ICCV*, 2013.

[33] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. *CVPR*, 2014.

[34] P. Viola, M. J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2005.

[35] S. Walk, N. Majer, K. Schindler, and B. Schiele. New features and insights for pedestrian detection. *CVPR*, 2010.

[36] X. Wang, X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. *CVPR*, 2009.

[37] X. Wang and L. Lin. Dynamical and-or graph learning for object shape modeling and detection. *NIPS*, 2012.

[38] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li. Robust multi-resolution pedestrian detection in traffic scenes. *CVPR*, 2013.

[39] Z. Zhao, F. Morstatter, S. Sharma, S. Alelyani, A. Anand, and H. Liu. Advancing feature selection research asu feature selection repository. *technical report*, 2013.

[40] L. Zhu, Y. Chen, and A. Yuille. Learning a hierarchical deformable template for rapid deformable object parsing. *TPAMI*, 33(6):1029–1043, 2010.

[41] Z. Zhu, P. Luo, X. Wang, and X. Tang. Deep learning identity preserving face space. *ICCV*, 2013.

[42] Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover canonical-view faces in the wild with deep neural networks. *Technical Report, arXiv.org*, 2014.