# Adaptive Pedestrian Detection by Modulating Features in Dynamical Environment

Song Tang*, Lijuan Chen*, Jinpeng Mi*, Mao Ye†, Jianwei Zhang, Qingdu Li

***Abstract*— The accuracy of a trained pedestrian detector is always decreased in a new scenario, if the distributions of the samples in the testing and training scenarios are different. Traditional methods solve this problem based on domain adaption techniques. Unfortunately, most of existing methods need to keep source samples or label target samples in the detection phase. Therefore, they are hard to be applied in the real applications with dynamical environment. For this problem, we propose a feature modulation model, which consists of a Simple Dynamical Neural Network (SDNN) and a Modulating Neural Network (MNN). In SDNN, a dynamical layer is adopt to adaptively weight the feature maps, whose parameters are predicted by MNN. For each candidate proposal, the SDNN generates a proprietary deep feature respectively. Our contributions include 1) the first feature-based unsupervised domain adaptation method which is very suitable for real applications and 2) a new scheme of dynamically weighting feature maps, in which the corresponding training method is also given. Experimental results confirm that our method can achieve the competitive results on two pedestrian datasets.**

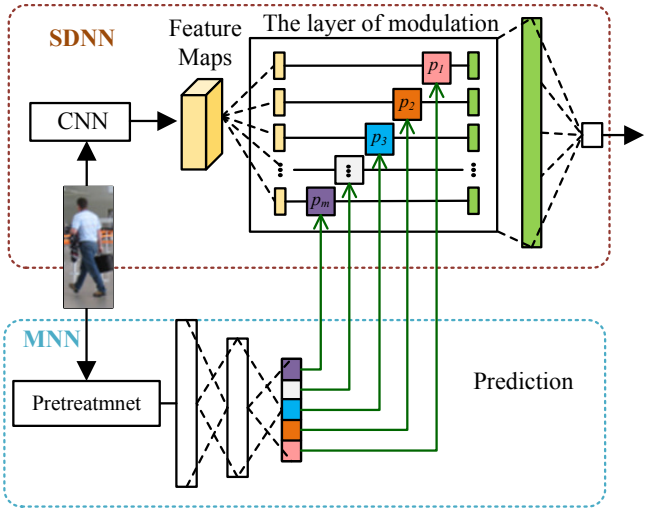


Fig. 1. The architecture of FMNN.

## I. Introduction

Recently, more and more attentions are payed on the domain adaptation of object detection. Namely, the distributions of the training (source) and test (target) samples are different, and the detection tasks are same. Meanwhile, as an important branch of object detection, pedestrian detection has been a hot research [1] [2] [3] for a long time owing to its great potential in various engineering fields, such as autopilot, intelligent surveillance and environmental perception. However, at present, the works about pedestrian detection based on domain adaptation are few. The existing methods can be divided into two kinds.

The first kind is the semi-supervised method. In this kind of methods, some labeled target samples are needed. Its basic idea is extracting domain-crossed features by exploiting some labeled target samples. For example, Sermanet *et al.* [4] pre-trained convolutional kernels on the source dataset, and then these trained kernels were fine-tuned based on some labeled target samples. Li *et al.* [5] proposed to reserve the domain-shared convolutional kernels and update the non-shared kernels of a Convolutional Neural Network (CNN) detector. With the help of cross-domain featuresthe mentioned methods above achieve good performance. However, these methods require that some target samples must be labeled to fine-tune the detector.

The second kind is the unsupervised method, which means that the labels of target samples are unavailable. Its basic idea is mining context information of the target domain to re-train the detector. The methods are typical detector-based methods. Nair *et al.* [6] proposed an online method which learned a classifier with automatically labeled samples by background subtraction. Following this work, Wang's team had made a serial of works [7]–[10] on transferring detector to specific scenes. In [8] a framework of pedestrian detection was proposed for traffic scene which automatically mined confident positive and negative examples in the target domain to adapt a pre-trained generic pedestrian detector. The works in [7], [9] exploited the target-context information to mine more reliable target-scene samples. And a deep model detector was developed in [10] which mined multi-scale scene-specific features and visual patterns in the target domain through a reconstruction layer and a cluster layer respectively. Since the context information was absorbed, the re-trained detectors presented good performance. Although some improvements have been made, these methods need to reserve source samples in the detection phase which are very unsuitable for practical applications. Moreover, this kind of methods are not suitable for the scenarios with dynamical background.

It is not hard found out that the methods above have a problem that source samples are kept or some labeled target

*Authors contributed equally
Song Tang, Jinpeng Mi, Jianwei Zhang, Qingdu Li are with TAMS, Department of Informatics, University of Hamburg, Hamburg 22527, Germany
Lijuan Chen, Mao Ye are with School of Computer Science and Engineering, School of Mathematical Science, Center for Robotics, Key Laboratory for NeuroInformation of Ministry of Education, University of Electronic Science and Technology of China, Chengdu 611731, China
†Correspondence author:cvlab.uestc@gmail.com

samples are required in the detection phase. In fact, in real scenarios, this requirement can not be well-satisfied. For example, the robots working in dynamical environment. In these cases, the robots are ignorant for the future working circumstance. This situation make it impossible that the robots keep the source sample and obtain labeled sample from the scenes changed all the time.

For solving the problem, Tang *et al.* [11] proposed a new classifier based solution. Concretely, combining the deep convolutional network and the idea of neural modulation, a dynamical classifier is designed which is adaptively adjusted by a neural network. Since that each candidate proposal is classified by a sample based classifier, the method presents good detection performance of transfer. However, this method has poor adaptability for different task of computer vision. It is only available for classification task.

It is well known that the accuracy of classification depends on two aspects: feature and classifier. To be different from the work of [11], this paper intends to solve the problem above from another view, i.e. the feature representation.

At present, the deep learning based transfer method has attracted many attentions. However, for the detection applications with one-class, since that the deep models are always trained on a big dataset with multi-class, the feature maps from the well-trained deep models can not be directly used. In order to obtain better performance, the feature maps should be selected properly. Inspired by this idea, we propose a new domain adaptation method, named Features Modulation Neural Network (FMNN). In order to illustrate the idea of features modulation, in this paper, we first attempt the simplest scheme, i.e, dynamically weighting the feature maps. The experiments show that this simple approach is feasible.

Our contributions can be summarized as follows.

1) The first feature-based unsupervised framework is proposed for the domain adaption of pedestrian detection. In the detection phase, the feature extraction method are dynamical changed. In this way, for each candidate proposal, the proposed method will adaptively generate unique feature, which is well-compatible for the current status of the dynamical environment. Moreover, this framework can be easily extended to other tasks of computer vision. The two properties make our method is completely different from previous works.

2) A new scheme of weighting feature maps is proposed to implement the dynamical feature extraction. In the corresponding training method, we not only design a new object function with a sparse constraint, but also propose some training skills, for example the learning-rate controlling.

## II. Overview of the Feature Modulation Neural Network

The architecture of FMNN is presented in Fig. 1. The top row is the Simple Dynamical Neural Network (SDNN) which includes three parts. The first part is a CNN that is taken as the feature extractor. For an image with size $u_1 \times u_2$, the CNN will convert it to $m$ feature maps with size $v_1 \times v_2$. Followed the extractor, there is the layer of modulation, in which the $k^{th}$ input feature map is weighted by the parameter $p_k \in R$, for $k = 1, \cdots, m$. In the end, one fully connected layers are attached as the classifier.

The bottom row is the Modulating Neural Network (MNN) including two components. The one is the part of pretreatment which is used to filter the noise in the image. The another one is a neural network with three fully connected layers. Its construction is $m_1 - m_2 - m$, the active function of the hidden layer and output layer are PReLU and sigmoid respectively. The output of MNN is taken as the predicted values of the parameters in the layer of modulation. In this way, this layer is controlled by MNN.

## III. Training Method of FMNN

Suppose that there are source and target domain. The source domain consists of $N$ labeled samples that include $N_p$ positive samples and $N_q$ negative samples. These samples and the corresponding labels are denoted by $x^i \in R^{m \times n}$ for $i = 1, \cdots, N$ and $y^i \in R$ for $i = 1, \cdots, N$ respectively. For the $i^{th}$ source sample $x^i$, its label is $y^i$. The target domain only includes a serial of images containing pedestrians.

We train FMNN end-to-end by minimizing the following object function:

$$L = \frac{1}{2N} \sum_{i=1}^{N} \left\| z^i(\theta) - y^i \right\|^2 + \alpha \|\mathbf{p}\|_1 = L_1 + \alpha L_2 \quad (1)$$

where $z^i(\theta)$ is the predicted label of $x^i$, $\theta$ represents the parameters of MCNN, $\alpha$ is a regularization constant, $\mathbf{p} = (p_1, \cdots, p_m)$ is the predicted weighting parameters and $\|\cdot\|_1$ means L1-norm.

In Eq. 1, the first item $L_1$ represents the constraint of accurate classification. Since that a lot of works have proved that the sparse is a common characteristic for neural connections. The sparse constraint to the predicted weights vector is additionally introduced by the second item $L_2$. It is known that the L1-norm is not differentiable at 0, and hence poses a problem for gradient-based methods. To solve this problem, we use the following differentiable equivalent [12].

$$\|\mathbf{p}\|_1 = \sum_{k=1}^{m} \sqrt{(p_k)^2 + \varepsilon}$$

where $\varepsilon$ is a small positive constant.

To solve the minimal optimization problem in Eq. 1, we employ the cross iteration algorithm to jointly train the parameters of both the SDNN and MNN. The concise presentation of the training method is given in Algorithm 1. In the steps, the 7th step can be easily implemented by feeding forward MNN. And the other steps and the training skills will be introduced in the remainder of this section respectively.

### A. Training SDNN

According to Algorithm 1, when the dynamical layer in SDNN is initiated, it is an ordinary CNN with one pooling layer and one fully connected layer . Therefore, we can train it using the standard error Back Propagation (BP) algorithm.

**Algorithm 1** The training method of FMNN.

**Require:**

Source samples: $X = \left\{ x^i \middle| i = 1, \cdots, N \right\}$;
Labels of $X$: $Y = \left\{ y^i \middle| i = 1, \cdots, N \right\}$;
Learning-rate of SDNN: $r_{SDNN}$;
Basic-learning-rate of MNN: $r_{MB} = \frac{1}{a} r_{SDNN}$;
A constant: $\beta$;

**Ensure:**

The parameters of FMNN: $\theta$;

**while** $L$ does not attach to the convergence **do**
  **if** $L > \beta$ **then**
    Learning-rate of MNN $r_M$=2$r_{MB}$
  **else**
    Learning-rate of MNN $r_M$=$r_{MB}$
  **end if**
  Predicting weights by the modulating network: $\mathbf{p}$;
  Taking $\mathbf{p}$ as the parameters of the dynamical layer;
  Fixing MNN and training SDNN;
  Fixing SDNN and training MNN.
**end while**
**return** $\theta$;

### *B. Training MNN*

MNN is a special BP network. In this network, the error signals in the output layer are back propagated from SDNN. In the following, the details of updating the parameters are presented.

We first introduce the training method of connection parameters from the hidden layer to the output layer. Suppose that the output of the $i^{th}$ hidden-neuron is $h_i^2$ for $i = 1, \cdots, m_2$. The input and output of the $k^{th}$ output-neuron are $g_k^3$ and $h_k^3$ for $k = 1, \cdots, m$ respectively. They satisfy $h_k^3 = \varphi\left(g_k^3\right)$ where $\varphi(\cdot)$ is the active function. The connection between the $i^{th}$ hidden-neuron and the $k^{th}$ output-neuron is $w_{ik}^3$ for $i = 1, \cdots, m_2$ and $k = 1, \cdots, m$. According to the gradient descent method, $w_{ik}^3$ are updated by the following rule,

$$w_{ik}^3(n+1) = w_{ik}^3(n) + r_M \frac{\partial L}{\partial w_{ik}^3} \tag{2}$$

where $n$ is the index of iteration, $r_M$ is the learning-rate. By the chain rule, $\partial L / \partial w_{ik}^3$ is obtained.

$$\begin{aligned} \frac{\partial L}{\partial w_{ik}^3} &= \frac{\partial L}{\partial p_k} \frac{\partial p_k}{\partial w_{ik}^3} \\ &= \left( \frac{\partial L_1}{\partial p_k} + \alpha \frac{\partial L_2}{\partial p_k} \right) \frac{\partial p_k}{\partial w_{ik}^3} \\ &= \left( \frac{\partial L_1}{\partial p_k} + \alpha \frac{\partial L_2}{\partial p_k} \right) \frac{\partial p_k}{\partial h_k^3} \frac{\partial h_k^3}{\partial g_k^3} \frac{\partial g_k^3}{\partial w_{ik}^3} \end{aligned} \tag{3}$$

For the formula above, the key is the computation of $\partial L_1 / \partial p_k$ which describes how the errors back propagate from DDCNN to the modulating network. Its details are presented as follows.

Suppose that the $k^{th}$ input and output feature map of the dynamical layer are $A^k \in R^{v_1 \times v_2}$ and $B^k \in R^{v_1 \times v_2}$ respectively. $\sigma^k$ are the local gradients in the $k^{th}$ output feature map. In order to compute derivatives using the BP algorithm, here, the process of weighting the feature maps is regarded as a special pooling which can be presented by

$$A^k = C^k \odot B^k$$

where $\odot$ means element-time, $C^k$ is the pooling matrix whose elements $C_{ij}^k = p_k$ for $i = 1, \cdots, v_1$ and $j = 1, \cdots, v_2$. By the BP algorithm, the partial derivatives of $L_1$ with respect to $C^k$ is obtained.

$$D^k \equiv \frac{\partial L_1}{\partial C^k} = A^k \odot \sigma^k$$

We deem the $L_1$ as the function of $C_{ij}^k$ for $i = 1, \cdots, v_1$ and $j = 1, \cdots, v_2$. Correspondingly, $\partial L_1 / \partial p_k$ can be presented as

$$\begin{aligned} \frac{\partial L_1}{\partial p_k} &= \sum_{i=1}^{v_1} \sum_{j=1}^{v_2} \frac{\partial L_1}{\partial C_{ij}^k} \frac{\partial C_{ij}^k}{\partial p_k} \\ &= \sum_{i=1}^{v_1} \sum_{j=1}^{v_2} D_{ij}^k \end{aligned} \tag{4}$$

Combining Eq.4 and the following relationship

$$\begin{aligned} &\frac{\partial L_2}{\partial p_k} = p_k \left( \varepsilon + (p_k)^2 \right)^{-\frac{1}{2}} \\ &\frac{\partial p_k}{\partial h_k^3} = 1, \frac{\partial h_k^3}{\partial g_k^3} = \varphi^{'}\left(g_k^3\right), \frac{\partial g_k^3}{\partial w_{ik}^3} = h_i^2, \end{aligned}$$

according to the manner of the BP algorithm, Eq. 2 can be re-written as follows.

$$w_{ik}^3(n+1) = w_{ik}^3(n) - r \delta_k^3 h_i^2$$

where $\delta_k^3$ is the local gradients in the output layer and expressed by

$$\delta_k^3 = \left( \sum_{i=1}^{v_1} \sum_{j=1}^{v_2} D_{ij}^k + \alpha p_k \left( \varepsilon + (p_k)^2 \right)^{-1/2} \right) \varphi^{'}\left(g_k^3\right) \tag{5}$$

As for the connection parameters from the input layer to the hidden layer, they are updated by the rule similar to Eq. 2. Since the local gradients in the output layer are given (Eq. 5), the partial derivatives of $L$ with respect to the parameters can be computed using the standard BP algorithm.

### *C. Training skills*

As shown in Algorithm 1, two learning-rate skills are used in the training process. The first one is the learning-rate matching (i.e. $r_{MB} = \frac{1}{a} r_{SDNN}$). Since that FMNN is a heterogeneous network, the errors in SDNN and MNN do not match. The mismatch will cause MNN fall into the saturation situation. This is observed in the experiments. When the error directly back-propagates from SDNN to MNN, the output of MNN will easy to be 1 or 0. This will lead to the vanishment of error in the modulating network. To avoid this problem, this skill is introduced. In practice, $r_{SDNN} = 0.1$ and $a = 10000$ according to experience.

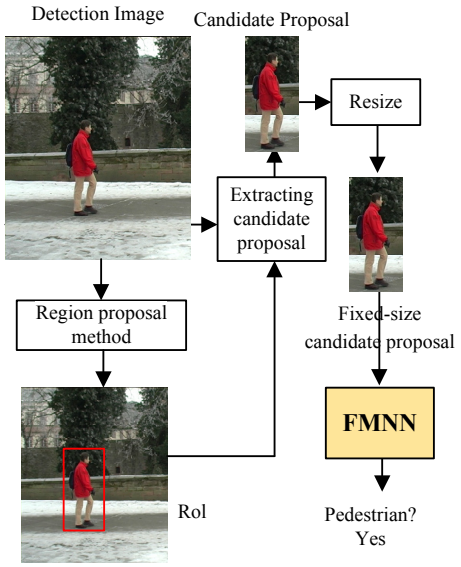


Fig. 2. The overview of the region-based detection procedure.

The second one is the learning-rate adjusting skill which is used in the iterations (step 2-6 in Algorithm 1). In order to highlight the importance of hard samples, we introduce the constraint. Based on the deep features, the most training samples are classified rightly while a few hard samples that are classified wrongly. If we adopt same learning rate, the MNN will tend to remember the parameter pattern of the samples classified rightly. In this way, MNN hardly predict weights for the hard sample. In practice, $\beta = 0.01$.

## IV. Detecting based on FMNN

Inspired by [13], [14], in this paper, our model is applied in the region-based detection framework for pedestrian detection. Fig. 2 gives the overview of our region-based detection procedure. At first, we employ the region proposal method to propose some regions of interest (RoIs). Then, we extract candidate proposals in accordance with the location of each RoI. Since that the size of each RoI is not fixed, all candidate proposals are resized to $160 \times 48$ pixels. At last, the fixed-size candidate proposals are input into our feature modulation model.

## V. Experiments

In this section, we firstly introduce the experiment setting. And then, the effectiveness of the proposed method is presented from two aspects by experiments.

### A. Experiment setting

**Source and target domains**: In this paper, we adopt the dataset proposed by [11] as the source domain, which totally includes 30825 images with size $160 \times 64$ including 12825 pedestrian and 18000 background images. And the CUHKsquare [7] dataset, including training and test sub-dataset, and TUDpedestrians [15] are taken as the target domain.

**Evaluation criterions**: In the evaluation experiments, we adopt the commonly used PASCAL rule. In addition, as done in [16], we make the evaluation on TUPpedestrians through drawing Precision versus Recall curves and calculating the Area Under Curve (AUC) measure. According to previous works [7], Detection Rate versus False Positive Per Image (FPPI) is used as the evaluation metric on the CUHKsquare.

**Model setting**: In experiment, considering the computational efficiency and characteristics of pedestrian, we respectively select the AlexNet [17] and the HOG [18] as the feature extractor and pretreatment in our model. Meanwhile, the structure parameters of FMNN are respectively set as $u_1 = 160$, $u_2 = 64$, $v_1 = 10$, $v_2 = 4$, $m_1 = 3348$, $m_2 = 1500$ and $m = 256$.

In the detection procedure, since our goal is to detect pedestrians, we prefer to select ACF [19] as our region proposal method other than class-agnostic methods, such as PRN [20].

### B. The experiment I

In the experiment, we present the experiment results compared with the previous transfer methods. First, we present the experimental results on CUHKsquare dataset. To prove the effectiveness, 10 representative detection approaches are taken as comparisons, namely CNNDAC [11], RCNN [13], FAST-RCNN [14], FUOLF [6], TGSVM [7], AGPD [8], CSCNN [10], ASVM [21], CDSVM [22] and CovBst [23].

These methods can be divided into 4 kinds. The first kind, including [13] [14], is deep model based methods. [13] transfer the deep features obtained by the well-trained deep CNN to new detection task by fine-tuning on the new domain. [14] is a faster version. The second kind, including [21] [22] [23], is HOG feature based semi-supervised methods. In order to transfer the detector, they require some manually labeled target samples for training. Concretely, analyzes the score distributions of the existing classifiers and transfers the existing classifiers to the target domain by learning a delta function. [22] adapts a pre-trained SVM by learning a new decision boundary with almost no additional computational cost. [23] shifts the selected features to the most discriminative locations and scales, and selects the related samples from source dataset by changing the weighting coefficients. The third kind, including [6] [7] [8] [10], is unsupervised methods. As introduced in Section 1, for transferring, they absorb the information from target domain. The fourth kind, including [11], is based on the idea of neural controlling, which is similar to our work.

Fig. 3(a) and 3(b) show the ROC curves of the above mentioned methods on CUHKsquare train and test sets respectively. Our method obtains the second best result on these two datasets. Except CNNDAC, our method is much better than other comparisons. Meanwhile, it is very close to the best comparison, i.e. CNNDAC. In Fig. 4, the first and second row represent some typical detection result on CUHKsquare train and CUHKsquare test respectively.

Second, we present the comparative experimental results on TUDpedestrians dataset. For proving the effectiveness,

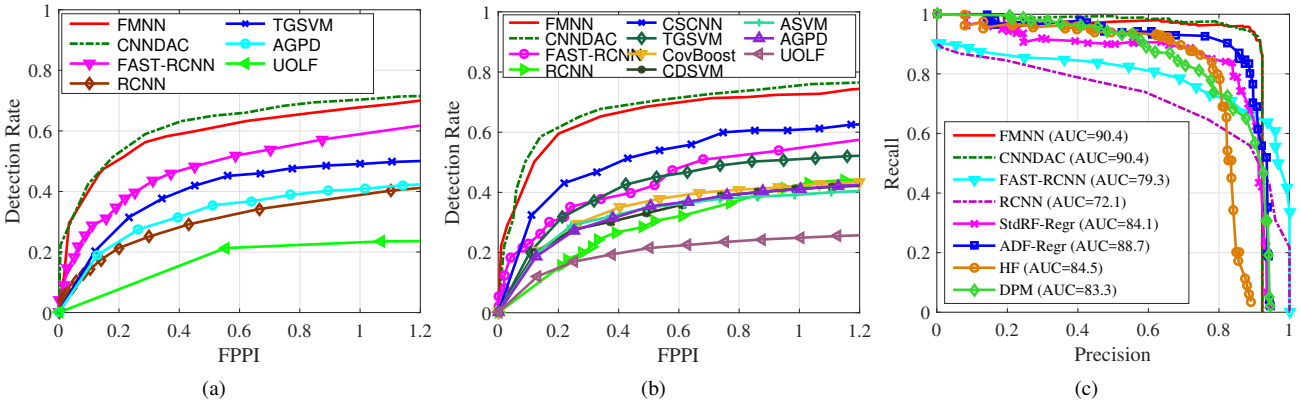


Fig. 3. The detection results of MCNN and the comparison methods on the two datasets. (a) result on CUHKsquare train; (b) result on CUHKsquare test; (c) result on TUDpedestrian.

we compare our method with 7 state-of-the-art detection approaches, including CNNDAC [11], RCNN [13], FAST-RCNN [14], StdRF-Regr [16], ADF-Regr [16], Hough Forests(HF) [24] and DPM [18].

Among them, StdRF-Regr, ADF-Regr and HF are based on the random forest framework. HF proposes a new object representation which regards the object as a set of small patches connected to a reference point. ADF-Regr and StdRF-Regr train a joint model to simultaneously predict the object probability and its aspect ratio. DPM is a part-based multi-component model which achieves good results on many datasets.

Fig. 3(c) gives ROC curves of the methods above. Similarly, our method obtains the best result. The AUC of our method is 90.4 that is same to CNNDAC. Compared with the third best method ADF-Regr, there is an improvement of 1.7 in the AUC measure. Some typical detection results on TUDpedestrians are represented in the third row of Fig. 4.

Fig. 4. Some typical detection result of FNMM on CUHKsquare train, CUHKsquare test and TUDpedestrians

In conclusion, our method obtain competitive results on the two target domains. In our opinion, the main reason is that by modulating feature map weights, the harmful feature maps are adaptively depressed while the helpful feature maps are adaptively preserved. Therefore, the new deep features are more suitable for the target detection task.

It is also noted that CNNDAC is slightly better than our method. There are two reasons. 1) Compared with feature modulation adopt by this paper, the classifier modulation adopt by CNNDAC is more directly for task of classification. 2) CNNDAC introduces a new regularization to make the dynamical classifier only sensitive to the hard samples. Compared with the similar skill used in FMNN, i.e. learning-rate controlling, it is a more natural way.

### C. The experiment II

In this experiment, we present that the predicted weights is dynamical. To prove the dynamicity, the predicted weights of test samples from MITpedestrian dataset [25] are investigated. For convenience of observation, we randomly select 8 example samples, denoted respectively by $s_1$, $s_2$, $s_3$, $s_4$, $s_5$, $s_6$, $s_7$, $s_8$, as shown in Fig. 5. For clarity, we divide them into 4 pairs as $a = (s_1, s_2)$, $b = (s_3, s_4)$, $c = (s_5, s_6)$, $d = (s_7, s_8)$ and visualize the difference value of the corresponding weights. As shown in Fig. 6, the predicted weights indeed vary with the change of testing samples. This indicates that the proprietary prediction is effective.

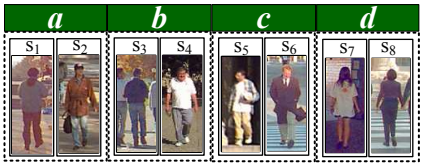


Fig. 5. The 8 example samples from MIT dataset.

## VI. Conclusion

In this paper, we propose a new modulated CNN architecture for the problem of pedestrian detection based on domain adaptation. The modulated CNN has a feature-map-weight layer whose parameters are controlled by another modulating

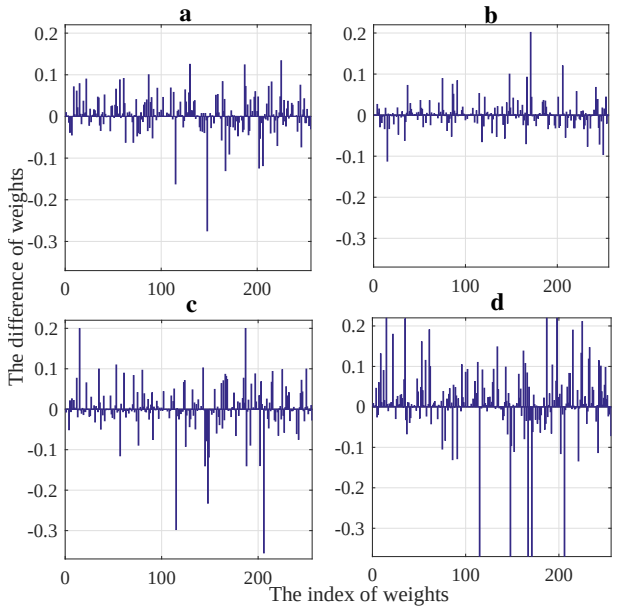


Fig. 6. The predicted weights of 4 sample-pairs from MITpedestrian.

network. By the dynamical weight layer, the modulated CNN can adaptively generate deep proprietary feature for every detection candidate. The experiments show that our method is effective. In addition, to be different from the most existing methods, our method does not keep source samples and label target samples. This property makes our method very suitable for real applications.

Moreover, the model is a general transfer framework, which can be directly extend to other tasks of computer vision, for example the task of scenes segmentation. How to extend the proposed network to different applications will be the focus of our future work.

## VII. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (61375038) and Applied Basic Research Programs of Sichuan Science and Technology Department (2016JY0088).

## References

[1] H. Hattori, V. N. Boddeti, K. Kitani, and T. Kanade, "Learning scene-specific pedestrian detectors without real data," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2015.

[2] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2015.

[3] K. Liu, B. Ma, W. Zhang, and R. Huang, "A spatio-temporal appearance representation for viceo-based pedestrian re-identification," in *IEEE international conference on computer vision(ICCV)*. IEEE, 2015.

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

[5] X. Li, M. Ye, M. Fu, P. Xu, and T. Li, "Domain adaption of vehicle detector based on convolutional neural networks," *International Journal of Control, Automation and Systems(IJCAS)*, vol. 13, no. 4, pp. 1020–1031, 2015.

[6] V. Nair and J. J. Clark, "An unsupervised, online learningframework for moving object detection," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2004, pp. 317–324.

[7] M. Wang, W. Li, and X. Wang, "Transferring a generic pedestrian detector towards specific scenes," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2012, pp. 3274–3281.

[8] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2011, pp. 3401–3408.

[9] X. Wang, M. Wang, and W. Li, "Scene-specific pedestrian detection for static video surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 36, no. 2, pp. 361–374, 2014.

[10] X. Zeng, W. Ouyang, M. Wang, and X. Wang, "Deep learning of scene-specific classifier for pedestrian detection," in *European Conference on Computer Vision(ECCV)*. Springer, 2014, pp. 472–487.

[11] S. Tang, M. Ye, C. Zhu, and Y. Liu, "Adaptive pedestrian detection using convolutional neural network with dynamically adjusted classifier," *Journal of Electronic Imaging*, vol. 26, no. 1, p. 013012, 2017.

[12] http://deeplearning.stanford.edu/wiki/index.php/Sparse_Coding:_Autoencoder_Interpretation.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2014, pp. 580–587.

[14] R. Girshick, "Fast R-CNN," in *IEEE international conference on computer vision(ICCV)*. IEEE, 2015.

[15] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2009, pp. 1014–1021.

[16] S. Schulter, C. Leistner, P. Wohlhart, P. M. Roth, and H. Bischof, "Accurate object detection with joint classification-regression random forests," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2014, pp. 923–930.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems(NIPS)*, 2012, pp. 1097–1105.

[18] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 32, no. 9, pp. 1627–1645, 2010.

[19] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 36, no. 8, pp. 1532–1545, 2014.

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[21] J. Yang, R. Yan, and A. G. Hauptmann, "Cross-domain video concept detection using adaptive svms," in *ACM international conference on Multimedia*. ACM, 2007, pp. 188–197.

[22] W. Jiang, E. Zavesky, S.-F. Chang, and A. Loui, "Cross-domain learning methods for high-level visual concept classification," in *IEEE International Conference on Image Processing(ICIP)*. IEEE, 2008, pp. 161–164.

[23] J. Pang, Q. Huang, S. Yan, S. Jiang, and L. Qin, "Transferring boosted detectors towards viewpoint and scene adaptiveness," *IEEE transactions on image processing(TIP)*, vol. 20, no. 5, pp. 1388–1400, 2011.

[24] J. Gall and V. Lempitsky, "Class-specific hough forests for object detection," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 2009.

[25] M. Oren, C. Papageorgiou, P. Sinha, E. Osuna, and T. Poggio, "Pedestrian detection using wavelet templates," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*. IEEE, 1997, pp. 193–199.