Journal of Computers Vol. 30 No. 2, 2019, pp. 240-251
doi:10.3966/199115992019043002022

A Real-time Pedestrian Detection Model Adopted by High-Performance ANS-DPM

Ai-Ying Guo[1], Feng Ran[2*], Jian-Hua Zhang[1], Mei-Hua Xu[3],
Lu-Ming Gong[2], Hua-Ming Shen[3]

[1] Key Laboratory of Advanced Display and System Applications Ministry of Education, Shanghai, China
{gayshh, jhzhang}@shu.edu.cn

[2] Microelectronic Research and Development Center, Shanghai, China
{ranfeng, gluminous}@shu.edu.cn

[3] School of Mechatronics Engineering and Automation, Shanghai University, Shanghai, China
{mhxu, hmshen}@shu.edu.cn

Received 1 August 2017; Revised 7 January 2018; Accepted 1 February 2018

Abstract. In order to fulfill the real-time pedestrian detection system, it is difficult to balance the higher detection rate and greater detection speed at the same time. Therefore, this paper proposed an Adaptive Neighborhood Selection-Deformable Part Model (ANS-DPM) as a novel pedestrian feature operator to solve this problem. ANS-DPM adopts the adaptive feature zones to extract the pedestrian features based on the relationship between features ex-traction score with the experience threshold to decrease the whole calculation and upgrate. Meanwhile, the inner-layer and inter-layer constraint are introduced into the ANS-DPM to deal with the model of robustness. Finally, ANS-DPM with Latent-SVM constructs the pedestrian detection system and experiment result shows that the ANS-DPM can improve the feature rate, while the pedestrian detection system can detect 30 fps to satisfy the real-time detection system.

Keywords: adaptive, ANS-DPM, feature extraction, Latent-SVM, pedestrian detection

1 Introduction

Pedestrian detection is the key technology and places an important role in many applications, such as: video surveillance, assistance driving vehicle devices, human-computer interaction system, and so on [1]. Nowadays, the research of pedestrian detection aims to promote the detection rate and processing speed. Although the deep learning (un-supervision machine learning) is the hottest and reliable model to improve the detection rate, the huge hardware-cost is still the biggest difficulty for this detection model to be transplanted to on the vehicle device [2-5]. This means that the pedestrian detection based on the supervision algorithm is the ideal technology on the vehicle device.

On the supervision pedestrian detection, the methods can be divided into two parts: feature extraction and recognition, which are the key modules to improve the performance of pedestrian detection. Based on the different feature description, the feature of pedestrians can be divided into two kinds: scalar and vector features.

The Haar-like is the typical of scale feature, which was firstly put forward by Papageorgiou, Viola et al [6-7]. Haar-like features are confined with the integral image and Ada-boost classification algorithm. Meanwhile, this feature can real-time detect pedestrians by describe the gray-level pixels. But the detection rate is low and can't be proper to utilize in the pedestrian detection on vehicle device. Then, Liu proposed color channel aided-by the Haar-like feature algorithm to optimize the detection rate [8], but the result isn't still satisfy the demand of pedestrian detection system. In 2016, Zhou combined the Local Binary Pattern (LBP) algorithm with Haar as the new feature, this feature can effectively improve the

* Corresponding Author

detection rate, but the detection result shows not so good on the real-time performance [9].

Compared with scalar features, vector features can promote the detection rate, because this type feature have some certain invariant and show stronger description ability. David Lowe develops the Scale Invariant Feature Transform (SIFT) as an image descriptor, this description is suitable for matching between two images and isn't proper the description for the pedestrian detection [10]. In 2005, Histogram of Orientation Gradient (HOG) is formed by Dalal and Triggs [11]. HOG is the milestone algorithm in the whole development of pedestrian detection. Based on the HOG, scholars have carried out many other relative HOG, such as Co-HOG, 3D extension of Gradient Location and Orientation Histograms (3D GLOH) and so on. Although the detection rate can be improved, the long vector needs the huge amount of calculation [12-13]. However, the real-time of detection system is limited because of the high dimensionality from pedestrian feature. In order to promote the detection rate, Felzenszwalb brought forward DPM on the basis of HOG features which enhanced the description of features by a low resolution global model and a high-resolution part model [14]. Based on fast feature pyramid, Yang combined DPM with Aggregated Channel Features (ACF) [15]. Cheng advanced the automatic selection and placement of Mixture of Deformable Part Model (MDPM) [16]. Although the detection accuracy has been improved, it still cannot meet the real-time requirements.

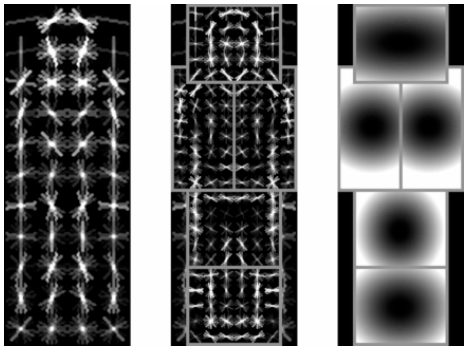
About these problems, this paper proposed an Adaptive Neighborhoods Selection-Deformable Part Module (ANS-DPM) as the new feature operator, which is based on the relationship between filter scores and threshold to decrease the computation and upgrade the processing speed. The ANS-DPM is able to adaptively select the filter zones by comparing the root filter score on low resolution with the empirical threshold. Meanwhile, in order to increase the robustness of features, the inner and inter constrains between layers are added into the ANS-DPM. Finally, the ANS-DPM cascade Latent SVM constructs the new pedestrian detection model. The experimental results from the pedestrian detection system show that ANS-DPM with the Latent SVM can process one frame using less than 20ms and satisfy the real time requirement of pedestrian detection system.

2 Relative Theory

In order to describe the theory of ANS-DPM, this section firstly analyses the original theory of filters in DPM model. Based on the analysis, it is found that the most time, which is used to extract DPM model, is the calculation of each filter in DPM not the matching calculation between root filter and relative part filters in one detection window. This conclusion can be used to design the ANS-DPM operator in the next section.

2.1 DPM Theory

In one detection window, the overall scores of DPM in one detection window is obtained by a low-resolution root filter, which describes the whole contour of the pedestrian, adding the partial filters, which exhibit the detail information of pedestrian, subtracting the deformation cost, which explains the object deformation between the low-resolution and higher-resolution. Fig. 1(a) stands for the one root filter in the HOG pyramid, Fig. 1(b) stands for the several part-filters in HOG pyramid, and Fig. 1(c) stands for the deformation cost between different resolutions layers.



(a) (b) (c)

Fig. 1. DPM model based on the HOG pyramid

Meanwhile, the scores of all filters in one detection window, which are obtained through filter parameters dot HOG features in feature pyramid, can be expressed by function. Firstly, let's suppose that this DPM is composed by N filters, then we define F_0 as a root filter and $(P_1,\cdots,P_n)$ as a series of part filters. F_i stands for the *i*-th part filter, $\boldsymbol{v}_i$ stands for a 2D vector and defines the anchor position coordinates of *i*-th part (this position before the deformation), which position respects to the coordinates of the root filter. d_i is a 4D vector and defines the deformation cost, which relatives to the anchor position after the parts are deformed. Second, let H be a HOG feature pyramid. Then, $p_i=(x_i,y_i,l_i)$ stands for one cell, which located in the *l*-th layer of the pyramid and $z=(p_1,\cdots,p_n)$ stands for the position in the HOG pyramid. Third, let $\phi(H,p_i,w,h)$ stand for the vector obtained by concatenating the HOG feature vectors in $w\times h$ rectangular region of H with top-left corner at p_i in a fixed order. When $i=0$, this model is a root filter. Based on the above definition, the total score of DPM in one detection window can be calculated by function (1):

$$\begin{cases} score(p_0,L,p_n)=\underbrace{\sum_{i=0}^{n}F_i\times f(H,p_i)}_{Root\text{-}score+Part\text{-}score}-\underbrace{\sum_{i=1}^{n}d_i\times f_d(dx_i,dy_i)}_{Deformation\ \cos t}+b \\ \begin{cases}\phi_d(dx,dy)=(dx,dy,dx^2,dy^2)\\ (dx_i,dy_i)=(x_i,y_i)-(2(x_0,y_0)+\boldsymbol{v}_i)\end{cases}\end{cases} \tag{1}$$

Besides, b stands for the offset, $\phi_d(dx,dy)$ stands for the deformation feature, and (dx_i,dy_i) for the part *i* position relative to the anchor. At the same time, the final score of each root filter is obtained from the optimal position score based on the part filters, as the following function express:

$$score(p_0)=\max_{p_1,\cdots,p_n} score(p_1,\cdots,p_n) \tag{2}$$

Before matching the root filter with the part filters, the first step is to calculate the response score of each filter. The score of part file *i* in the *l*-th layer of pyramid is $R_{i,l}(x,y)=F_i\cdot\phi(H,(x,y,l))$, and then the spatially transformed for each part filter can be expressed:

$$D_{i,l}(x,y)=\max_{dx,dy}\left(R_{i,l}(x+dx,y+dy)-d_i\cdot\phi_d(dx,dy)\right) \tag{3}$$

Considering on the deformation cost, this transformation also stands for the matching calculation between part filters and root filter. $D_{i,l}(x,y)$ is calculated by $R_{i,l}$ in linear time through the distance transform algorithm. Therefore, the total score of root filter in each layer can be expressed by the score root filter plus the transformed part filter score:

$$score(x_0,y_0,l_0)=R_{0,l_0}(x_0,y_0)+\sum_{i=1}^{n}D_{i,l_{0-\lambda}}(2(x_0,y_0)+v_i)+b \tag{4}$$

In function (4), the λ stands for required go down layers, which stands for the 2D resolution in someone layer. In addition, it is possible to simultaneously calculate the best part position during the during the processing of calculating $D_{i,l}$, as the following function express:

$$P_{i,l}(x,y)=\arg\max_{dx,dy}\left(R_{i,l}(x+dx,y+dy)-d_i\cdot\phi_d(dx,dy)\right) \tag{5}$$

According to the above functions, the highest score of root filter can be obtained, then the optimal position of part filters also can be calculated.

Analysis of Computational Cost

Suppose that the total pixels of one detected image are A, all part filters can be scaled and rotated. To calculate all the possible positions of one part filter in one detection window, DPM operator needs to multiply A with different coefficients, which will spend a large amount of calculation. Even if the DPM adopts the simplest tree topology between root filter and part filters, the total calculation still reach to $O\left(PA^2\right)$, which can't represent all the calculation of all parts in one detection window. Main reasons are explained as following:

(1) $O\left(PA^2\right)$ only stands for the total calculation, which match the optimal position of the part-filters and ignore the matching calculation of each part in one detected image, as the Fig. 2 showed. However, the part-matching usually requires the relative calculation of the possible position from each part filter. Suppose that the filter dimension is D, so the relative calculation is $O(D)$ and the total calculation is $O\left(PA(D+A)\right)$.

Fig. 2. Part matching in one detected image

Otherwise, if the spatial quantization step is α, then the part filter only can be located in the discrete position in the detected image, so the calculation of possible location is $O\left(\frac{A}{\alpha^2}P\left(D+\frac{A}{\alpha^2}\right)\right)$.

(2) It is necessary to calculate the micro-deformations from part filters. Thus, considering all possible positions of each part and suppose that there are $1/\sigma$ parts in one detected image may be deformed, the total calculation can reach up to the $O\left(\frac{A}{\alpha^2}P\left(D+\frac{A}{\alpha^2\sigma}\right)\right)$.

From the above analysis, the conclusion is that the biggest calculation of one DPM is the $O(D)$, which stands for the whole filters response calculation, includes root filter and all part filters. This conclusion can be used to design the ANS-DPM operator.

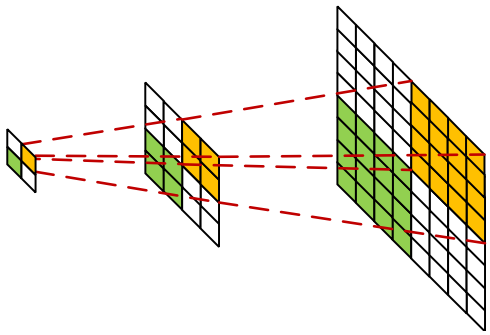
3 ANS-DPM Operator

ANS-DPM operator is achieved by comparing the score of the root filter in low-resolution layer with the experience threshold, and the automatic selection of the feature extraction area of filters in higher resolution layer. This can be used to reduce the computation. Secondly, in order to upgrade the robustness of ANS-DPM operator, the inner-layer and inter-layer constraint are introduced to enhance the feature description.

3.1 Threshold Division and Neighborhood Selection

DPM is a multi-resolution pedestrian feature extraction. In order to decrease the calculation, simplifying feature extraction, which is according to the inheritance relationship between pixels under different resolutions, is necessary. This kind of inheritance relationship includes two aspects. One aspect is that, in the image feature pyramid, two different objects in the image should occupy different image space; the other is that there are parent-child relations at the same pixel between different resolutions. For example,

one pixel in top layer of pyramid is the parent node of the middle layer, and that middle layer is the parent node of the lowest layer as the Fig. 3 showed.



(a) Top layer (b) Middle layer (c) Low layer

Fig. 3. Parent-child relationship of different resolution layers

According to the parent-child relationship between different resolution layers, the middle layer and lower layer can be selected the calculation area by calculation numeric on the top layer. So, the proposed the CtF DPM theory as the Fig. 4 showed [17]. The middle layer can be chosen by the top layer through the root score and experience threshold. The chosen area is the 3×3 neighborhood area in the different resolution layer. But the biggest drawback is that the CtF DPM operator isn't sensible to feature density.

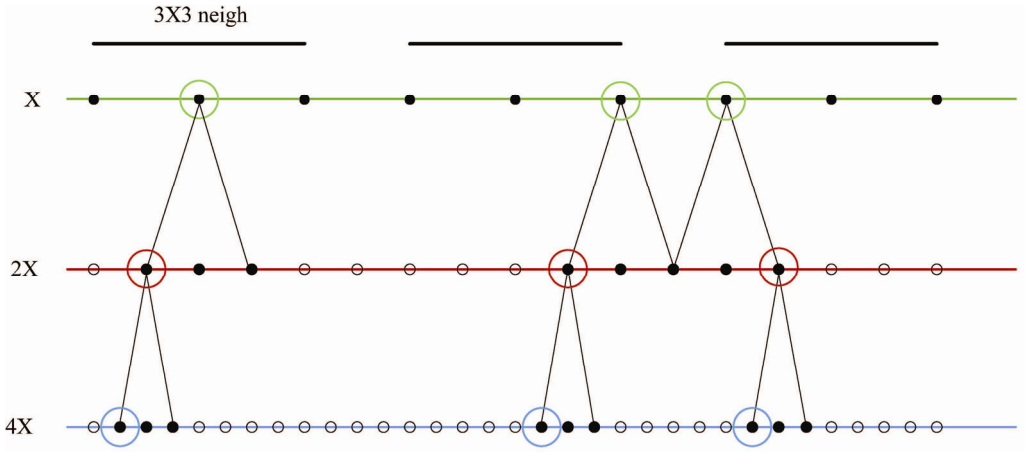


Fig. 4. Parent-child neighborhood in different resolution

So, in order to decrease the calculation and upgrade the sensitivity of feature density, we give the different experience thresholds on the different resolution and recording the relative detection accuracy as the Table 1.

Table 1. Influence of different threshold and neighborhood on detection accuracy

Size	Threshold	Accuracy /%
5×5	0.1	82
	0.3	88
4×4	0.3	87
	0.5	90
	0.7	91
3×3	0.7	90
	0.9	91

Meanwhile, according to the experience threshold and accuracy in the Fig. 5, it is found that when the score of root filter is 0.5, the accuracy of final detection results can reach more than 85%. Therefore,

three thresholds, T_1=0.3, T_2=0.5 and T_3=0.7 are taken to divide the root filter score curve into four parts, and the part filter calculation area is adaptively selected.

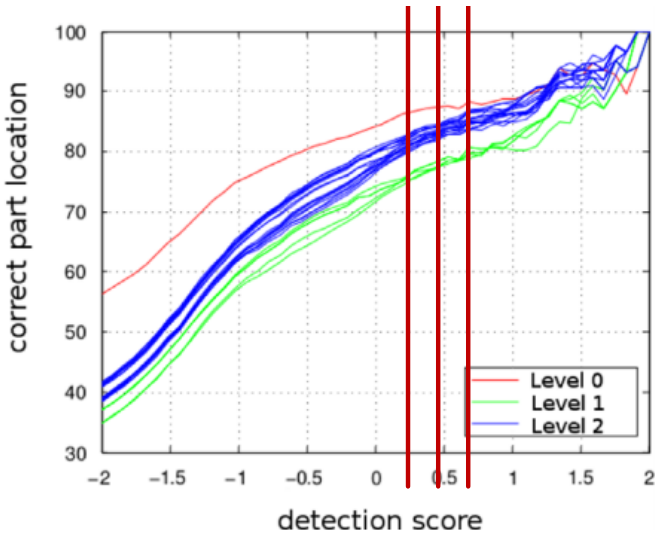


Fig. 5. Threshold and detection accuracy in different resolution

So based on the Table 1 and Fig. 5, the neighborhoods of ANS-DPM are 3×3 neighborhoods, 4×4 neighborhoods and 5×5 neighborhood, specific adaptive neighborhood selection is shown in Fig. 6. The ANS-DPM can adaptively select the neighborhood according to the relation between scores and T_1, T_2, T_3.

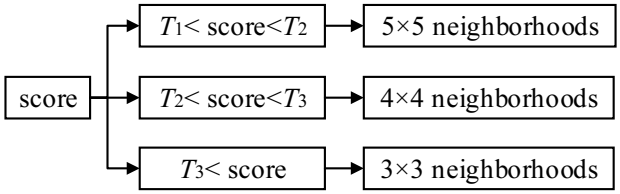


Fig. 6. Adaptive selection neighborhood in different resolutions

3.2 Inner-Layer and Inter-Layer

Similar to DPM, the ANS-DPM operator also includes some implicit relative, especially the inner-layer relationship and inter-layer relationship. Suppose that the ANS-DPM is used as the pedestrian operator and there is only one 31D HOG filter, which contains $w\times h$ cells in the layer (top layer). Based on this, in the middle layer (2 X resolution), one root filter will be divided into four filters, which are four 31D HOG filters. Then, at the low layer, the root filter will be divided into sixteen 31D HOG filters, each containing $w\times h$ cells.

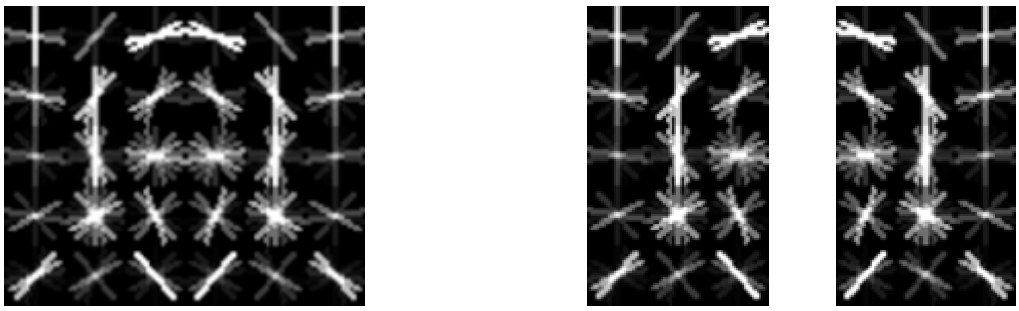
Based on the above analysis, let's suppose that $Y_i, i=1,2,\cdots,P$ is the position of the P parts and $Y_i=(x_i, y_i)$. Then the scores of the part position Y can be regarded as the sum of all the implicit item scores and the deformation item scores:

$$S(Y;X,w)=\sum_{i=1}^{P} S_{H_i}(Y_i;X,w)+\sum_{(i,j)\in F} S_{F_{ij}}(Y_i,Y_j;w)+\sum_{(i,j)\in P} S_{P_{ij}}(Y_i,Y_j;w) \quad (6)$$

Where F stands for the parent-child relationship, P is the implicit relationship in the same layer, and $\boldsymbol{w}$ is the model parameter vector used for training. $S_{H_i}(\boldsymbol{Y}_i;\boldsymbol{X},\boldsymbol{w})$ can be calculated through linear-filter, it is means that $S_{H_i}(\boldsymbol{Y}_i;\boldsymbol{X},\boldsymbol{w})=H(\boldsymbol{Y}_i;\boldsymbol{X})\cdot M_{H_i}(\boldsymbol{w})$, while $H(\boldsymbol{Y}_i;\boldsymbol{X})$ stands for the HOG at $\boldsymbol{Y}_i$ location in image X; $M_{H_i}(\boldsymbol{w})$ is a filter parameter in the i-th filter. $S_{F_{ij}}(Y_i,Y_j;w)$ stands for the offset in the location

Y_j, which compared to the parent node location Y_i; $S_{F_{ij}}\left(Y_i, Y_j; w\right)$ can be expressed to $S_{F_{ij}}\left(\boldsymbol{Y}_i, \boldsymbol{Y}_j; \boldsymbol{w}\right)=D\left(2\boldsymbol{Y}_i, \boldsymbol{Y}_j\right)\cdot M_{F_i}\left(\boldsymbol{w}\right)$, while $M_{F_i}\left(\boldsymbol{w}\right)$ stands for the deformation coefficient from the parameter vector $\boldsymbol{w}$. $S_{P_{ij}}\left(\boldsymbol{Y}_i, \boldsymbol{Y}_j; \boldsymbol{w}\right)=D\left(\boldsymbol{Y}_i, \boldsymbol{Y}_j\right)\cdot M_{P_{ij}}\left(\boldsymbol{w}\right)$ stands for the inner-layer deformation, while $S_{P_{ij}}\left(\boldsymbol{Y}_i, \boldsymbol{Y}_j; \boldsymbol{w}\right)=D\left(2\boldsymbol{Y}_i, \boldsymbol{Y}_j\right)\cdot M_{P_{ij}}\left(\boldsymbol{w}\right)$. Others, deformation coefficient is mainly from the parent-child relationship.

If there is no inner-layer relationship in the ANS-DPM operator, the model will be distorted as the Fig. 7 shows.



(a) inner-layer constraint (b) without inner-layer constraint

Fig. 7. Inner-layer relative and without inner-layer relative

About the area waiting for calculation in different resolution, the usually method is cascaded program or dynamic real-time calculation. In the ANS-DPM, the cascaded program is introduced into the ANS-DPM as the Fig. 8 showed. According to the different thresholds, the different layers can be cascade to combine the ANS-DPM to deal with the inter-layer relationship.

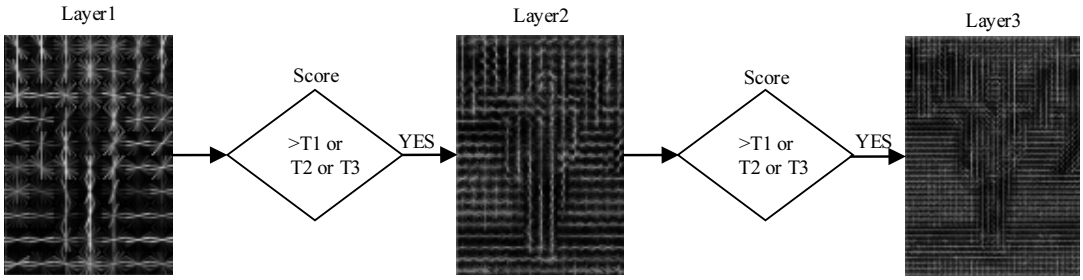


Fig. 8. Cascaded in different layers

4 Performance Testing and Analysis

In this section, the performance of ANS-DPM operator is tested mainly in the self-built SHU-Dataset [18]. Firstly, this section gives the whole pedestrian detection model using ANS-DPM. Secondly, the theory of ANS-DPM is analysis and shows the result. Thirdly, the ANS-DPM with Latent-SVM is tested based on the SHU-Dataset and compared with the other typical methods. Finally, the ANS-DPM is tested with the other datasets to show the different experiment results.

4.1 Real-time Pedestrian Detection Aided by ANS-DPM

The pedestrian detection system aided by ANS-DPM is combined with two parts: real-time detection module and offline training module as the Fig. 9 shows. In the real-time detection module, the image is inputted from the video, the ANS-DPM is used to extract the pedestrian feature. Meanwhile, the cascaded program in ANS-DPM is introduced to deal with the different thresholds in inter-layers. After the ANS-

DPM, the features are input the Latent-SVM as the classify module to determine the pedestrian or not. Other, the Latent-SVM is offline trained by SHU-Dataset.

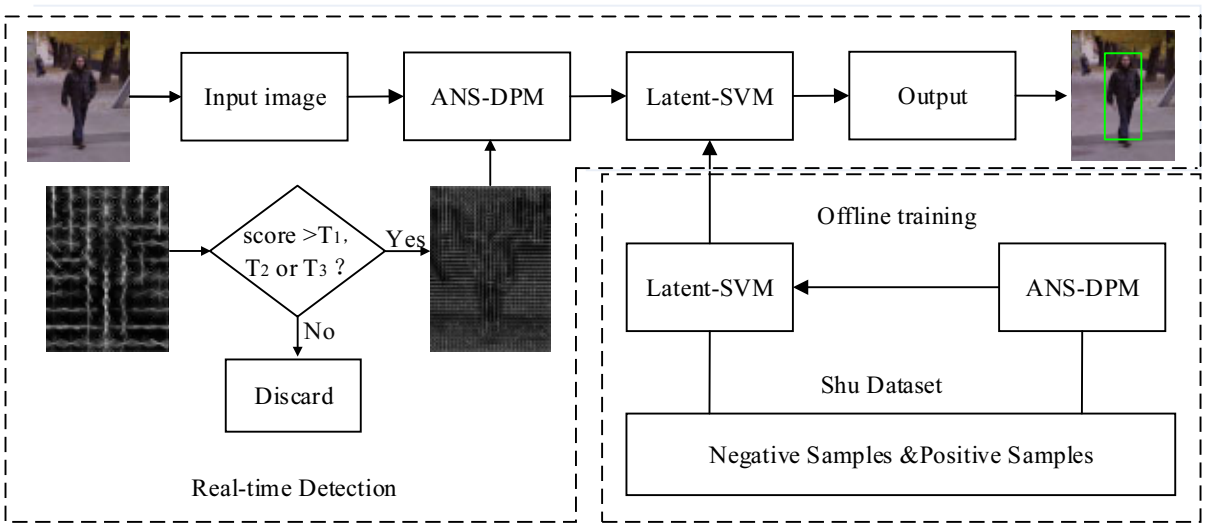


Fig. 9. Flow of pedestrian detection aided by ANS-DPM

4.2 Performance Test

About the ANS-DPM operator, the results in different layer are showed in the Fig. 10. Fig. 10(a) is the one original frame to be detected. Fig. 10(b), Fig. 10(c) and Fig. 10(d) stands for the HOG pyramid in top layer, middle layer and low layer. In the Fig. 10(f), Fig. 10(g) and Fig. 10(e) stands for the different selection area in the top layer, middle layer and low layer. According to the Fig. 11, we use the different color lines to stand for the filters in top layer (red line), middle layer (green line) and low layer (blue line). From the Fig. 10, it can be seen that the ANS-DPM can be used for efficiently feature extraction.

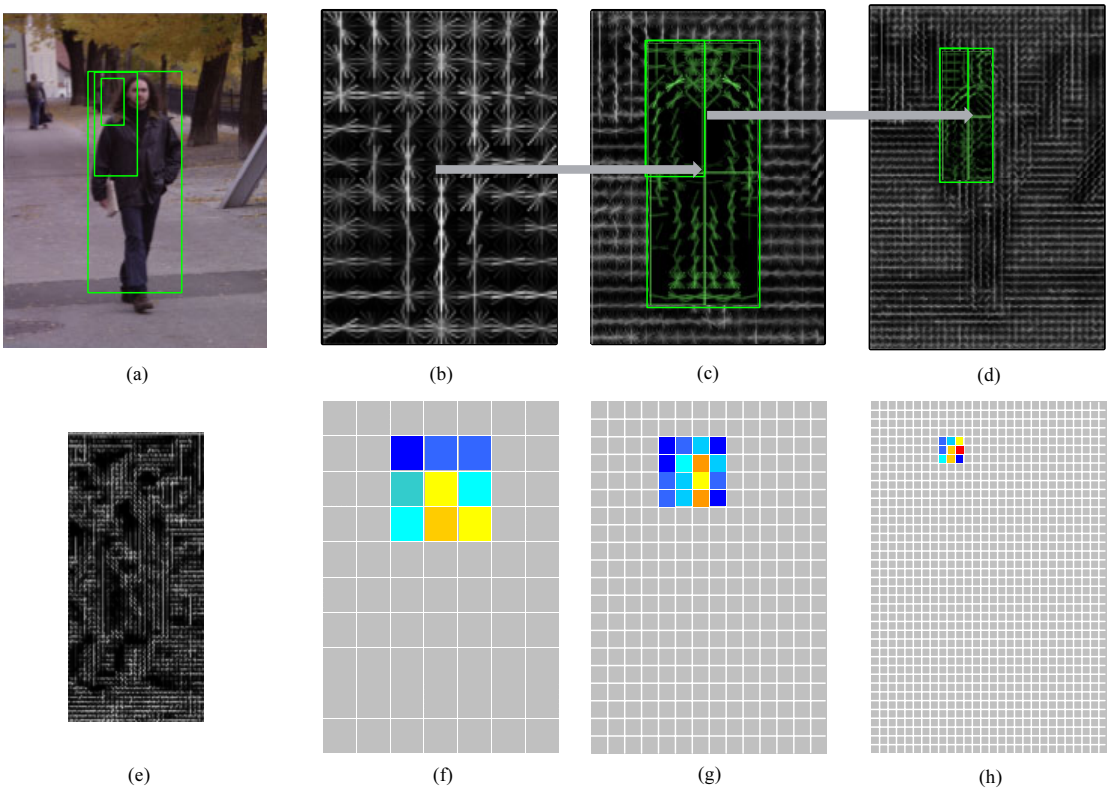


Fig. 10. Actual feature extraction process of ANS-DPM

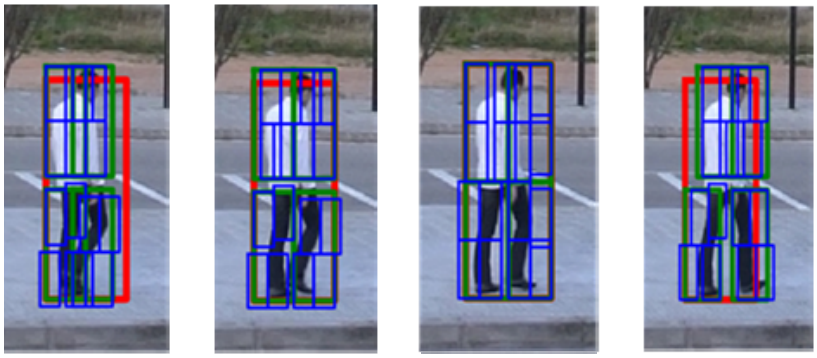


Fig. 11. ANS-DPM feature extraction area for each layer

In ANS-DPM, it is necessary to introduce the inner-layer constraint to upgrade the robustness. Therefore, DPM, ANS-DPM with or without inner-layer constraint are used to detect the pedestrians and recorded the data. The comparison is shown in Fig. 12. The abscissa represents the score of the ANS-DPM, and the ordinate represents the score of the traditional DPM. In the Fig. 12, the green line indicates the score in the ideal case, and the yellow line indicates the actual score. It can be seen that the performance of the ANS-DPM does improve the performance significantly after introducing the inner-layer constraint.

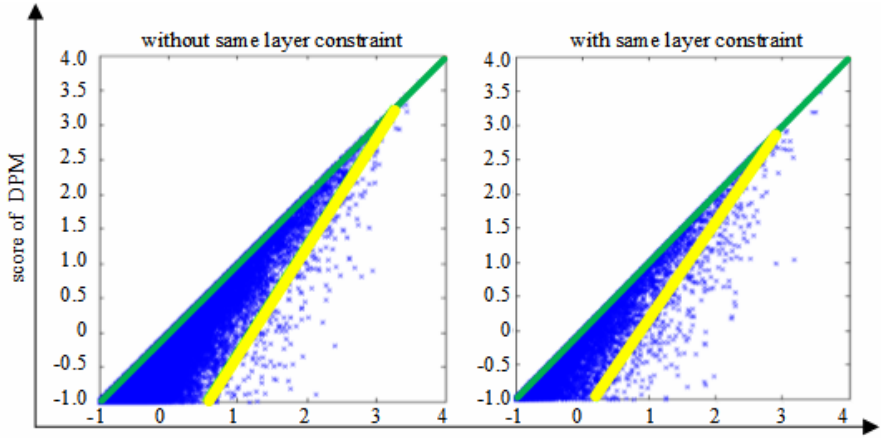


Fig. 12. Score of with or without inner-layer constraint

Table 2. Time-consuming of ANS-DPM with inner-layer constraint or not

Operator	Time-consuming/s
ANS-DPM (without inner-layer constraint)	0.014
ANS-DPM (with inner-layer constraint)	0.017

Others, in order to evaluate the performance of ANS-DPM, this paper introduces VJ [19], HOG +linear SVM [11], HOG LBP [20], LatSVM-V1 [21], LatSVM-V2 [22], FPDW [23], Crosstalk [24] and other classify algorithms to compare based on the FPPI. These experiments are achieved in the Visual Studio 2012, the PC configuration of detection is as follows: HP p6709cx Desktop PC; Dual Core Pentium E5700 @ 3.00GHz; 4G memory. The number of images are 2000, which is from SHU-Dataset. The FPPI performance curve is shown in Fig. 13, and it can be seen that the ANS-DPM has some advantages over other feature extraction algorithms.

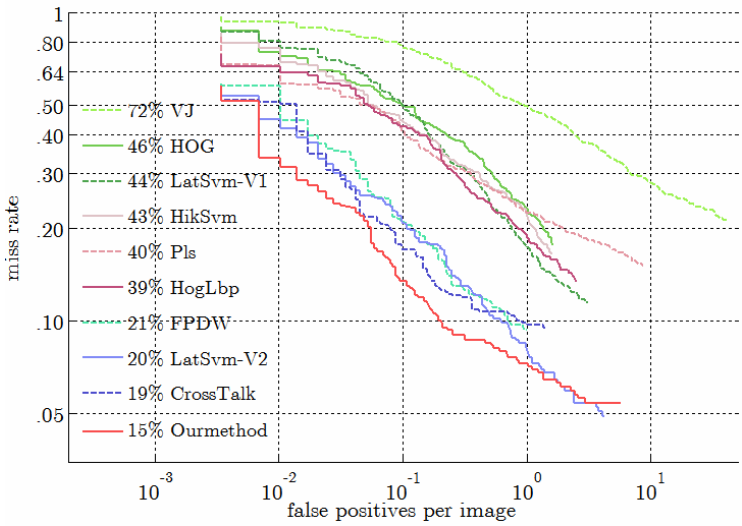


Fig. 13. FPPI performance with classic algorithms

4.3 ANS-DPM under Actual Situation

In order to verify the pedestrian detection performance of the ANS-DPM, we establish the SHU Dataset as the dataset for training Latent-SVM and testing the performance of the whole system. Meanwhile, the SHU Dataset includes 4523 pedestrians and 40 minutes videos, which is 30fps and captured on the downtown area, campus and some other places in Shanghai. Fig. 14 shows the some detection results using ANS-DPM + Latent-SVM under different scene.

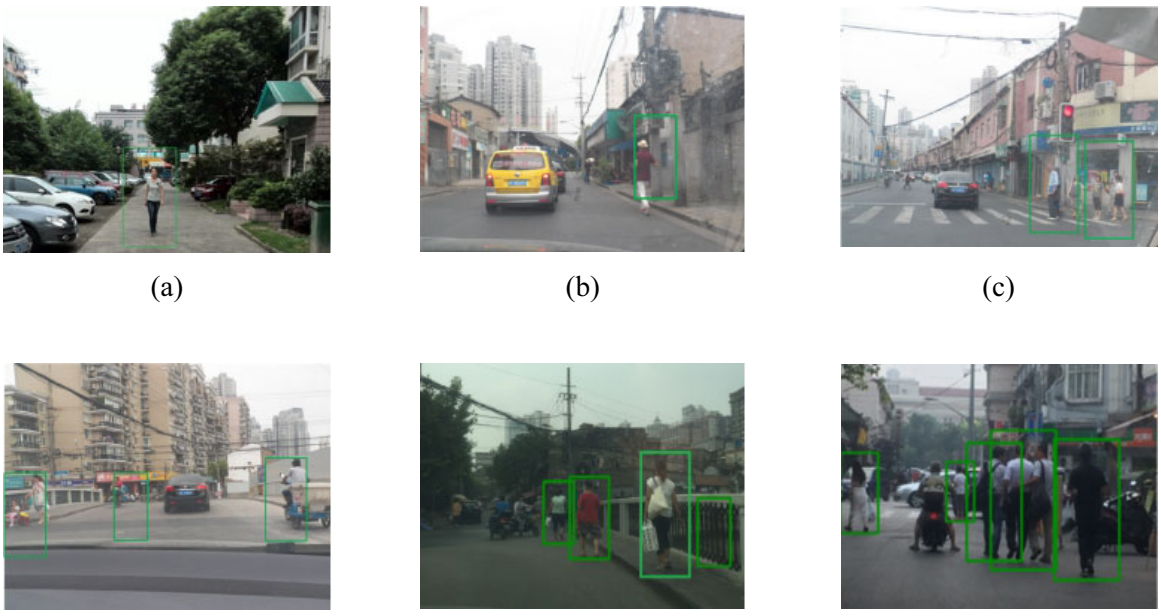


Fig. 14. Actual detection of ANS-DPM model

From the Fig. 14, the conclusion is that the ANS-DPM with Latent-SVM can detect the pedestrians very well when the pedestrian is slight blocked, as shown in Fig. 14(a) to Fig. 14(c). However, the ANS-DPM with Latent-SVM can effectively marks the pedestrians with obvious features, it still needs to promote the detection performance in pedestrian detection, as shown in Fig. 14(d) to Fig. 14(f).

Otherwise, this pedestrian detection system, which contains the ANS-DPM and Latent-SVM, is also used to detect pedestrians based on the CVC-02, Daimler and Caltech dataset. The Table 3 shows the detection results from the different dataset.

Table 3. Comparison of the accuracy of pedestrian detection in different datasets

Datasets	Detect Person	Total Person	Accuracy/%
Shu Dataset	1090	1957	83
CVC-02	1240	1736	81
Caltech	1360	1842	81
Daimler	2480	378	80

Analysis Table 3, the detection rate in Caltech is the minimum, because that the Daimler contains many obstacles and Data acquisition environment is bad. But the detection rate in Shu Dataset, CVC-02, Caltech and Daimler also show that ANS-DPM cascade Latent-SVM can be used as the real-time pedestrian detection system.

5 Conclusion

This paper introduces a real-time pedestrian detection model aided by an ANS-DPM operator. Firstly, the ANS-DPM is introduced based on the analysis about theory of DPM model. ANS-DPM operator exhibits the less-calculation and without sacrifice of detection rate compared with DPM. Secondly, in order to improve the robustness of ANS-DPM, the inner-layer and inter-layer are introduced into the ANS-DPM. Finally, the ANS-DPM is cascaded with Latent-SVM as the pedestrian detection system. The results from the theoretical analysis and actual road detection show the ANS-DPM with Latent-SVM can compress the detection speed less than the 20ms, and can be used as the real-time pedestrian detection system.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant 61376028 and 61674100), Open funding from the Key Laboratory of Advanced Display and System Applications Ministry of Education.

References

[1] J. Li, Q.L. He, L.Y. Yang, C.F. Sao, Pedestrian detection and counting in crowded scenes, in: Wang et al. (Eds.), Green Intelligent Transportation Systems, Springer Science+Business Media, Berlin, 2018, pp. 495-511.

[2] M. Bilal, A. Khan, M.U.K. Khan, C.M. Kyung, A low complexity pedestrian detection framework for smart video surveillance systems, IEEE Transactions on Circuits & Systems for Video Technology 27(10)(2017) 2260-2273.

[3] J.Y. Mao, T.T. Xiao, Y.N. Jiang, Z.M. Cao, What can help pedestrian detection?, Computer Vision and Pattern Recognition 1(2)(2017) 3.

[4] J.J. Wang, J.C. Yang, K. Yu, F.J. Lv, T. Huang, Y.H. Gong, Locality-constrained linear coding for image classification, in: Proc. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 2010.

[5] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, Microsoft COCO: common objects in context, in: D. Fleet, T. Pajadla, B. Schiele, T. Tuytelaars (Eds.), European Conference on Computer Vision, Springer, Cham, 2014, pp. 740-755.

[6] C. Papageorgiou, T. Poggio, A trainable system for object detection, International Journal of Computer Vision 38(1)(2000) 15-33.

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

[8] C. Liu, F. Chang, C. Liu, Cascaded split-level colour Haar-like features for object detection, Electronics Letters 51(25)(2015)

2106-2107.

[9] S. Zhou, G. Wang, Y. Xu, An improved HLBP texture feature method for pedestrian detection, Computer Engineering & Science 38(05)(2016) 960-967.

[10] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 60(2)(2004) 91-110.

[11] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, IEEE Computer Society Conference on Computer Vision & Pattern Recognition 1(2005) 886-893.

[12] C. Hans, C. McCollum, M. Bondesson, S. Shah, F. Merchant, Gradient weighted CO-HOG for analysis of caudal vein structural changes in toxin exposed zebrafish embryo, in: Proc. 2015 IEEE 12th International Symposium on Biomedical Imaging, 2015.

[13] A. Abdulmunem, Y.K. Lai, X. Sun, 3D GLOH features for human action recognition, in: Proc. International Conference on Pattern Recognition, 2016.

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

[15] J. Yang, G. Li, W. Wang, R. Wang, An empirical study of deformable part model with fast feature pyramid, in: Proc. 2016 23rd International Conference on Pattern Recognition, 2016.

[16] D. Cheng, Y. Gong, J. Wang, N. Zheng, Balanced mixture of deformable part models with automatic part configurations, in: Proc. IEEE Transactions on Circuits and Systems for Video Technology, 2016.

[17] M. Pedersoli, A. Vedaldi, J. Gonzalez, A coarse-to-fine approach for fast deformable object detection, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[18] A.Y. Guo, M.H. Xu, F. Ran, Q. Wang, Model of real-time pedestrian detection under vehicle environment based on CS-SD, Journal of Traffic and Transportation Engineering, Changan University Magazine 16(6)(2016) 132-139.

[19] P. Viola, M. Jones, Robust Real-time Face Detection, International Journal of Computer Vision 57(2)(2004) 137-154.

[20] X. Wang, T.X. Han, S. Yan, An HOG-LBP human detector with partial occlusion handling, in: Proc. International Conference on Computer Vision, 2010.

[21] W.C. Cheng, Y.Y. Cheng, Pedestrian detection in video using shape features and mixture of SVMs, in: Proc. International Congress on Image & Signal Processing, 2010.

[22] Y. Xu, C.N. Li, X.L. Xu, M. Jiang, J.G. Zhang, A two-stage hog feature extraction processor embedded with SVM for pedestrian detection, in: Proc. IEEE International Conference on Image Processing, 2015.

[23] Y. Li, E. Zhu, X. Zhu, J. Yin, J. Zhao, Counting pedestrian with mixed features and extreme learning machine, Cognitive Computation 6(3)(2014) 462-476.

[24] P. Dollár, R. Appel, W. Kienzle, Crosstalk cascades for frame-rate pedestrian detection, in: Proc. European Conference on Computer Vision, Springer-Verlag, 2012.