An Efficient Tree Classifier Ensemble-Based Approach for Pedestrian Detection

Yanwu Xu, *Member, IEEE*, Xianbin Cao, *Member, IEEE*, and Hong Qiao, *Senior Member, IEEE*

IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 41, NO. 1, FEBRUARY 2011

Outline

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Outline

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Introduction

- A practical pedestrian detection system(PDS) seeks for high detection accuracy and a low false positive rate and high detection speed simultaneously.
- At present, classification has become a predominant technique for the PDS owing to its accuracy and speed compared with techniques based on image processing.

- Most of the existing classification algorithms adopted in the PDS mainly concentrate on classification/detection accuracy and do not purposely optimize the detection/training speed.
- In this paper, we focused on how to speed up detection without degrading accuracy.

Outline

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Related Work

- Feature per object (fpo), a criterion independent from execution, is adopted to measure the speed in addition to the fps.
- Classification approaches based on a sliding window have been proven to outperform others and have become the predominant methods at present [48].

[48] X. Y. Wang, T. X. Han, and S. Yan, "A HOG-LBP human detector with partial occlusion handling," in *Proc. ICCV*, 2009.

- Although HOG [13] and HOG-like [33] features have achieved great success on pedestrian detection in recent years, they cannot be used in real-time onboard detection due to their high computational complexity.
- The Adaboost-based approach has the fastest detection speed and comparable accuracy without the time constraint.

[13] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. CVPR*, 2005, pp. 886–893.

[33] S. Maji, A. C. Berg, and J. Malik, “Classification using intersection kernel support vector machines is efficient,” in *Proc. CVPR*, 2008, pp. 1–8.

- Single Classifiers
  - Some researchers aim to train a single classifier with both high accuracy and high speed. However, these two demands might conflict with each other and can hardly be satisfied simultaneously.
  - Most single-classifier-based detection techniques concentrate on how to improve detection accuracy. The common strategy is to adopt more and better features, but this will lead to a low speed.

- Classifier Ensembles
  - Serial ensemble
    - The serial classifier ensemble uses a linear chain structure to arrange several single classifiers that are usually used in a cascaded way.
    - detection techniques based on serial-ensemble classification can get a higher detection speed and a lower detection rate comparatively.
  - Parallel ensemble
    - The parallel classifier ensemble arranges several single classifiers in a parallel way.
    - Although parallel-ensemble-based approaches might achieve higher accuracy, one serious drawback is that they usually consume much more time than single-classifier-based techniques.

- Classifier Ensembles
  - Mix ensemble
    - Combination of serial-ensemble and parallel ensemble classifiers might achieve more balanced and better overall performance.
    - in our previous work [49], a tree classifier was proposed, and higher detection accuracy was obtained, while the detection speed was almost the same as that of a serial-ensemble Adaboost classifier.

[49] C. Wei, X. Cao, Y. Xu, H. Qiao, and F. Wang, “The treelike assembly classifier for pedestrian detection,” in *Proc. PAISI*, 2009, pp. 232–237.

Outline

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Efficient Pedestrian Detection Approach Based On The Tree Classifier Ensemble

- *Brief Introduction of the Detection Procedure*
  - It takes a fixed 32 $\times$ 16 (pixel $\times$ pixel) sliding window as the classification unit.
  - Intercept a single frame from the video flow, and set the initial value of the loop counter N to be 0.
  - Zoom the original frame with a predefined scale stride Z ($0 < Z < 1$) in each round N
  - For each window region, a tree classifier ensemble is used to identify whether it contains a pedestrian by taking depth first searching. A window region is regarded as containing a pedestrian only when there is a positive path from the root to a leaf of the tree classifier.
  - If $N < 7$, increase N by 1 and go to step 2; else end detection of the present frame, output the detection result, and go to step 1 until the detection is terminated.

N=0

N=1

N=2

N=3

N=4

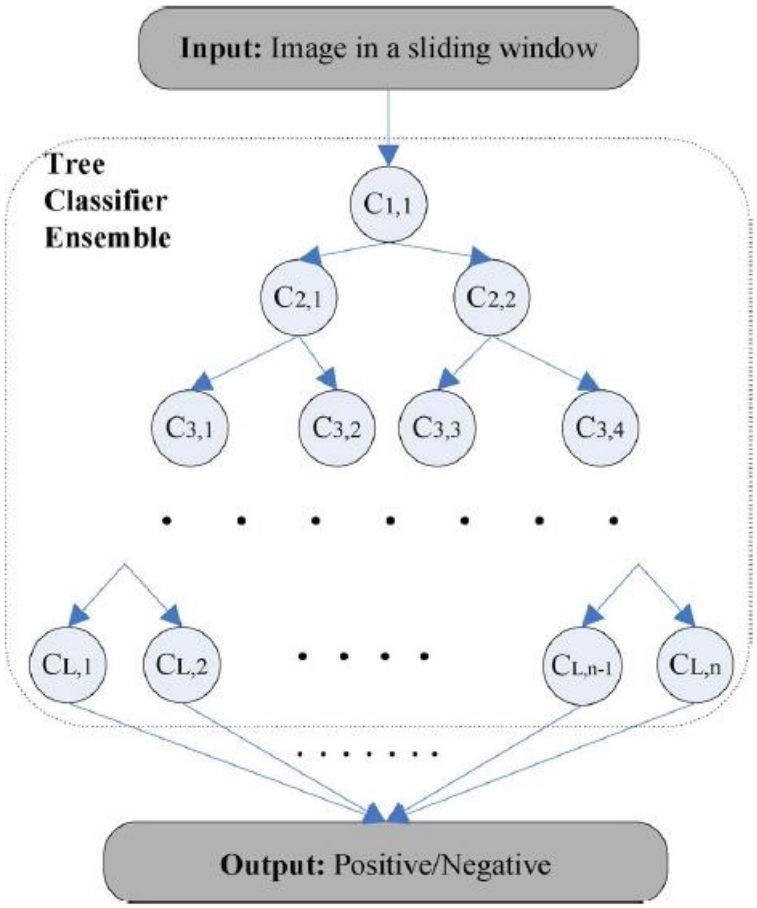
N=5

N=6

N=7

- *Execution-Independent Speed Metric*
  - The advantages of the proposed feature per object (fpo) metric can be concluded as follows.
    - Since fpo is independent from execution, the detection speed measured by fpo could be calculated and optimized directly and theoretically, whereas fps cannot.
    - Fpo can be used as a supplementary speed evaluation criterion to fps. It can be used to evaluate and compare some linear classifier-based approaches based on the same feature type in order to eliminate the influence of execution.

- *Tree Classifier Ensemble*
  - For an obvious nonpedestrian region, only a few single classifiers in top levels are used to reject it in order to get higher speed; for a pedestrian-like region, more single classifiers in more levels are used to increase the accuracy.



- *Performance Formulations and the Fpo Optimization Problem*
  - When an *L*-level tree classifier is used to detect pedestrians in one frame, for a positive object,

$$
\begin{aligned}
\mathrm{fpo}_p &= \sum_{l=1}^{L}\sum_{k=1}^{v_l} 1_{\{C_{l,k}\text{ is performed}\}} n_{l,k} \\
&= n_1 + (1 - fn(1))n_2 \cdot v_2 + (1 - fn(1,2))n_3 \cdot v_3 \\
&\quad + \cdots + (1 - fn(l|1,\ldots,l-1;\mathbf{n}))n_l \cdot v_l \\
&= n_1 + \sum_{l=2}^{L} v_l n_l (1 - fn(l|1,\ldots,l-1;\mathbf{n})) \qquad (2)
\end{aligned}
$$

$$
\begin{aligned}
\mathrm{fpo}_n &= \sum_{l=1}^{L}\sum_{k=1}^{v_l} 1_{\{C_{l,k}\text{ is performed}\}} n_{l,k} \\
&= n_1 + fp(1)n_2 \cdot v_2 + fp(1,2)n_3 \cdot v_3 \\
&\quad + \cdots + fp(l|1,\ldots,l-1;\mathbf{n})n_l \cdot v_l \\
&= n_1 + \sum_{l=2}^{L} v_l n_l fp\,(l|1,\ldots,l-1;\mathbf{n})) \qquad (3)
\end{aligned}
$$

- $v_l = 2l-1$ is the number of single classifiers in level l
- $C_{l,k}$ denotes the *k*th single classifier in the *l*th level
- n_l denotes the feature number of a single classifier in level *l*
- f_n denotes the probability of a positive sample being classified as *false* by at least one single classifier.
- *fp* denotes the robability of a negative sample being classified as *true.*

$$
\begin{aligned}
\mathrm{fpo} &= N_p \cdot \mathrm{fpo}_p + N_n \cdot \mathrm{fpo}_n \\
&= N_p \left(n_1 + \sum_{l=2}^{L} v_l n_l \left(1 - fn(l|1,\ldots,l-1;\mathbf{n})\right) \right) \\
&\quad + N_n \left(n_1 + \sum_{l=2}^{L} v_l n_l fp(l|1,\ldots,l-1;\mathbf{n}) \right) \\
&= n_1 + N_p \sum_{l=2}^{L} v_l n_l \left(1 - fn(l|1,\ldots,l-1;\mathbf{n})\right) \\
&\quad + N_n \sum_{l=2}^{L} v_l n_l fp(l|1,\ldots,l-1;\mathbf{n})
\end{aligned} \tag{4}
$$

$$
\begin{aligned}
&\arg\min_{n_1,\ldots,n_L} n_1 + N_p \sum_{l=2}^{L} v_l n_l \left(1 - fn(l|1,\ldots,l-1;\mathbf{n})\right) \\
&\quad + N_n \sum_{l=2}^{L} v_l n_l fp(l|1,\ldots,l-1;\mathbf{n}) \\
&s.t. \begin{cases} fp(l|1,\ldots,L;\mathbf{n}) \le \mu \\ 1 - fn(l|1,\ldots,L;\mathbf{n}) \ge \rho \\ 0 < n_l < N_l (l = 1,2,\ldots,L) \end{cases}
\end{aligned} \tag{5}
$$

- μ is the upper bound of the FPR, ρ is the lower bound of the detection rate, and N_l is the maximum feature number of the single classifier in the lth level.

- *Solution to Obtain Key Parameters*
  - there are only a few pedestrians in each frame in urban traffic.
  - For each single-classifier training, its false-negative rate is set to be a very small value in order to ensure that the tree classifier can get a high detection rate.

$$\arg\min_{n_1,\ldots,n_L}, n_1 + \sum_{l=2}^{L} v_l n_l fp(l|1,\ldots,l-1;\mathbf{n})$$

$$s.t. \begin{cases} fp(l|1,\ldots,L;\mathbf{n}) \leq \mu \\ 0 < n_l < N_l (l = 1,2,\ldots,L). \end{cases} \quad (6)$$

- *Solution to Obtain Key Parameters*
  - Radial basis function (RBF) neural networks, which can perform efficient nonlinear fitting, are introduced to construct the mapping from **n** to **fp**.

$$
\begin{aligned}
&fp(1) \xleftarrow{RBF_1} n_1 \\
&fp(1,2) \xleftarrow{RBF_2} (n_1, n_2) \\
&\cdots \qquad \cdots \\
&fp(1,2,\cdots,L) \xleftarrow{RBF_L} (n_1, n_2, \cdots, n_L)
\end{aligned} \tag{8}
$$

  - The optimal **n** and *L* can guarantee the minimum fpo, which represents the highest detection speed.

- *Classifier Ensemble Training With Optimal Parameters*
  - The single classifiers in the L-level tree are trained level by level, from root to leaf. In our experiments, a six-level full binary tree classifier ensemble is trained.
  - Each single classifier in level-l has n_l features.

- *Classifier Ensemble Training With Optimal Parameters*
  - Training samples of each single classifier are selected according to the following rules.
    - *Positive samples*: The positive samples of a parent node are equally divided into two sets. Each child node randomly chooses one as its positive training sample.
    - *Negative samples*: If a single classifier is the root node, its negative samples are randomly selected from the huge negative sample set; otherwise, its negative samples are chosen from the false-positive outputs of all its ancestor nodes. Both child nodes of a parent node share the same negative samples.
  - The training of a single classifier in level-l ($1 \leq l \leq 6$) terminates when its FPR reaches $fp(1, 2, \ldots, l)$, and its false-negative rate is not greater than 0.1%.

Outline

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Experimental Result

- All the experiments are carried out on an Intel 1.6-GHz (dual core) computer with 1-GB DDRII667 RAM. The implementation is based on Visual C++ 6.0. For training the tree classifier ensemble, Haar-like features [47] and the Adaboost algorithm are adopted in the experiments.
- Six thousand positive samples and more than 600 000 negative samples are used to train the classifiers.

[47] P. Viola, M. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, Jul. 2005.

- *RBF Fitting*

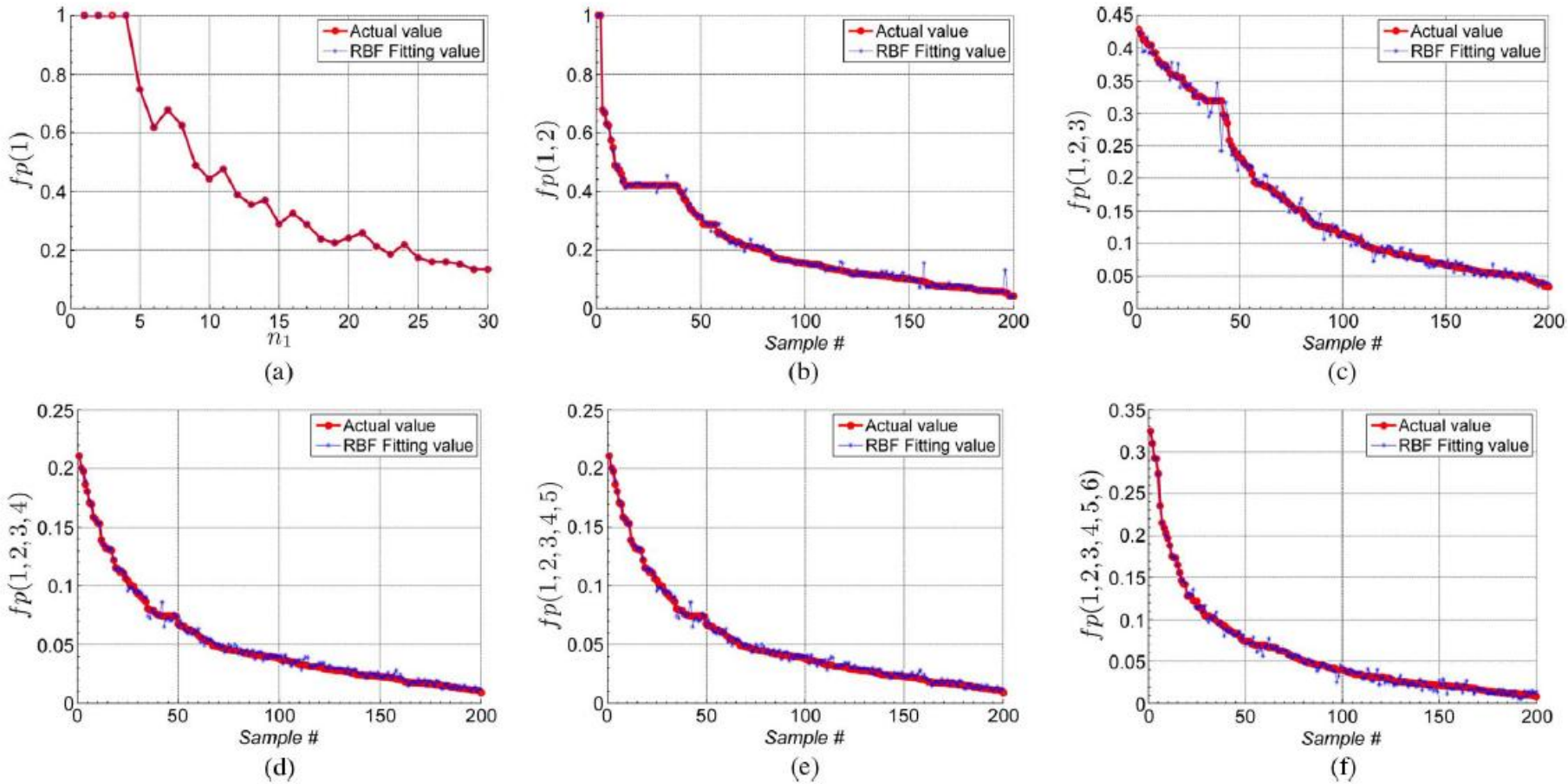


Fig. 4. RBF neural network fitting curves of testing samples. The mean average absolute fitting error of the six RBF neural networks is $\overline{\varepsilon} = 0.0040$. (a) $l = 1$, average absolute fitting error $\varepsilon_1 = \overline{|\Delta fp(1)|} = 0$. (b) $l = 2$, average absolute fitting error $\varepsilon_2 = \overline{|\Delta fp(1,2)|} = 0.0057$. (c) $l = 3$, average absolute fitting error $\varepsilon_3 = \overline{|\Delta fp(1,2,3)|} = 0.0061$. (d) $l = 4$, average absolute fitting error $\varepsilon_4 = \overline{|\Delta fp(1,\ldots,4)|} = 0.0043$. (e) $l = 5$, average absolute fitting error $\varepsilon_5 = \overline{|\Delta fp(1,\ldots,5)|} = 0.0046$. (f) $l = 6$, average absolute fitting error $\varepsilon_6 = \overline{|\Delta fp(1,\ldots,6)|} = 0.0031$.

- *Optimal Parameters*

TABLE II
FPO ERROR OF THE TWO OPTIMAL TREE CLASSIFIER ENSEMBLES

$\mu = 0.01$			
l	n_l	Expected $fp(1, 2, \cdots, l)$	Actual $fp(1, 2, \cdots, l)$
1	26	0.15944	0.159439
2	64	0.05463	0.054540
3	65	0.03106	0.033319
4	43	0.01824	0.018554
5	12	0.01020	0.009797
6	1	-	-
		Expected fpo 74.7988	Actual fpo 75.6127
		$\lvert\Delta fpo\rvert = 0.8139$	
$\mu = 0.005$			
l	n_l	Expected $fp(1, 2, \cdots, l)$	Actual $fp(1, 2, \cdots, l)$
1	28	0.151800	0.151804
2	67	0.049171	0.051284
3	66	0.030718	0.029165
4	53	0.014091	0.014877
5	23	0.010359	0.008730
6	11	0.004991	0.004688
		Expected fpo 83.1269	Actual fpo 82.7944
		$\lvert\Delta fpo\rvert = 0.3325$	

- *Classification Performance and Comparisons*
  - AdaBoost [9], [44], [47] and SVM [12], [25], [42], [46], [51]

TABLE III
CLASSIFICATION PERFORMANCE COMPARISON BETWEEN APPROACHES BASED ON DIFFERENT CLASSIFICATION ALGORITHMS

Classification Algorithm	Detection Accuracy		Detection Speed	
	DR(%)	FPR(%)	DS(fps)	DS(fpo)
Ours	92.34	0.72	16.50	83.20
Adaboost	88.57	3.24	14.13	87.46
	84.26	1.14	12.68	98.35
	78.15	0.81	10.94	112.43
SVM	79.20	44.45	0.172	100
	83.80	0.92	0.068	300
	86.80	0.13	0.044	500

[9] X. B. Cao, H. Qiao, and J. Keane, “A low-cost pedestrian-detection system with a single optical camera,” *IEEE Trans. Intell. Transp. Syst.*, vol. 9, no. 1, pp. 58–67, Mar. 2008.

[44] A. Shashua, Y. Gdalyahu, and G. Hayun, “Pedestrian detection for driving assistance systems: Single-frame classification,” in *Proc. IEEE Intell. Veh. Symp.*, 2004, pp. 1–6.

[47] P. Viola, M. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” *Int. J. Comput. Vis.*, vol. 63, no. 2, pp. 153–161, Jul. 2005.

[12] H. Cheng, N. Zheng, and J. Qin, “Pedestrian detection using sparse Gabor filter and support vector machine,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2005, pp. 583–587.

[25] G. Grubb, A. Zelinsky, L. Nilsson, and M. Rilbe, “3D vision sensing for improved pedestrian safety,” in *Proc. IEEE Intell. Veh. Symp.*, Jun. 2004, pp. 19–24.

[42] S. Schauland and A. Kummert, “Implementation and optimization of wavelet and symmetry features for vision-based pedestrian detection,” in *Proc. IASTED Int. Conf. Graph. Vis. Eng.*, 2007, pp. 19–24.

[46] Q. Tian, H. Sun, Y. P. Luo, and D. C. Hu, “Nighttime pedestrian detection with a normal camera using SVM classifier,” in *Proc. Int. Symp. Neural Netw.*, 2005, pp. 189–194.

[51] F. Xu, X. Liu, and K. Fujimura, “Pedestrian detection and tracking with night vision,” *IEEE Trans. Intell. Transp. Syst.*, vol. 6, no. 1, pp. 63–71, Mar. 2005.

Outline

- Introduction
- Related Work
- Proposed Method
- Experimental Result
- Conclusion

Conclusion

- The tree classifier ensemble is introduced to utilize the advantages of existing learning theory methods, which guarantees both high accuracy and speed for approaches based on simple features and classification algorithms.
- Fpo is introduced as a supplementary metric to measure the DS independently. A practical solution based on RBF neural networks is adopted to solve the optimization problem quickly and accurately.
- The optimal parameters can be obtained to instruct training a rapid and accurate tree classifier ensemble.