Cho and Hwang *EURASIP Journal on Image and Video Processing* (2015) 2015:16
DOI 10.1186/s13640-015-0074-5

EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

High-performance on-road vehicle detection with non-biased cascade classifier by weight-balanced training

Hanmin Cho and Sun-Young Hwang*

Abstract

In this paper, we propose a cascade classifier for high-performance on-road vehicle detection. The proposed system deliberately selects constituent weak classifiers that are expected to show good performance in real detection environments. The weak classifiers selected at a cascade stage using AdaBoost are assessed for their effectiveness in vehicle detection. By applying the selected weak classifiers with their own confidence levels to another set of image samples, the system observes the resultant weights of those samples to assess the biasing of the selected weak classifiers. Once they are estimated as biased toward either positive or negative samples, the weak classifiers are discarded, and the selection process is restarted after adjusting the weights of the training samples. Experimental results show that a cascade classifier using weak classifiers selected by the proposed method has a higher detection performance.

Keywords: Vehicle detection; AdaBoost; Cascade classifier; Weak classifier; Biased classifier

1 Introduction

Vehicle detection is a binary classification problem that distinguishes vehicles of different colors and shapes from cluttered backgrounds. Vehicle detection is employed in driver assistance systems and in autonomous vehicles [1, 2]. In surveillance systems that perform object detection using images from a static camera, differential images are used to locate the region of interest (ROI). In on-road detection environments, the background of images captured in a moving vehicle is not fixed and changes continuously. The ROI cannot be identified and the whole region of an image should be searched to detect vehicles. This means that a much larger number of operations is required. In applications related to on-road vehicle and transportation, driver safety is as important as convenience; thus, a robust and real-time detection system should be capable of providing an instant alarm to the driver or system [1]. From the driver's viewpoint, a late alarm would, in effect, be equivalent to a detection error. The system should give a warning to the driver as early as possible and needs to detect vehicles in the distance. This requires increased image resolution and makes the detection system more complex.

Radar sensors have been used for vehicle detection [1, 3]. They can detect objects without complex computation, even under poor illumination conditions. However, due to interference problems among sensors and poor lateral resolution [4], passive sensors such as cameras have been commonly employed for vehicle detection. Recently, the part-based model approach proposed by Felzenszwalb et al. [5] was applied to vehicle detection systems [6–8]. By detecting the parts of vehicles from images, their model combines the detected parts to detect vehicles [6]. While the part-based vehicle model improves detection performance, its computation complexity increases due to its algorithmic structure. Neural networks have emerged as a powerful machine learning model and have shown outstanding performance in detection systems [9, 10]. However, they require a tremendous amount of computing time, which makes it difficult to apply them to the detection of multiple moving objects in real time.

The cascade classifier proposed by Viola and Jones [11] has been commonly employed for real-time vehicle detection. The cascade classifier achieves both high processing speed and detection performance by employing

* Correspondence: hwang@sogang.ac.kr
Department of Electronic Engineering, Sogang University, 35 Baekbeom-ro, Mapo-gu, Seoul 121-742, Korea

simple classifiers at early stages to reject non-objects and complex classifiers at later stages. However, the detection rate is decreased as the stage proceeds in the cascade, and there is a large gap between the observed detection rate and the theoretical one. This is caused by biased classifiers. A classifier may be biased by unbalanced training samples due to a disparity between the difficulty of positive samples and that of negative ones. Each stage of a cascade classifier is trained by AdaBoost with training samples prepared using bootstrapping. Bootstrapping collects samples misclassified at a previous stage. It makes negative samples more difficult than positive ones as the stage proceeds; thus, training comes to focus more on negative samples.

Several methods have been proposed to improve the detection performance of cascade classifiers by assigning larger weights to positive samples. Most of these studies assumed that AdaBoost has a faulty weight update and modified it to assign larger weights to positive samples. These works improved detection performance for face detection; however, they failed to provide the weight values to be assigned to the positive samples.

In this paper, a new cascade classifier is proposed, in which the weak classifiers selected at each stage are confirmed for their effectiveness in the detection process using another sample set (reservoir set). The weights of these samples are updated by applying the selected weak classifiers using the same process in AdaBoost. The disparity, between the total weight assigned to the positive and negative samples in the reservoir set, is regarded as the degree of weight unbalance in the positive/negative training samples and bias in the selected weak classifiers. To generate non-biased classifiers, the disparity should be reduced. If the initial weight of the training samples is adjusted prior to training, the unbalance in the training samples is expected to be reduced and the weak classifiers selected after retraining are expected to be less biased. To determine the initial weight, the ratio of total weight assigned to the positive to the negative samples in the reservoir set is used as a reference value. This process is continued at each stage until non-biased weak classifiers are selected.

The rest of this paper is organized as follows: Section 2 presents the AdaBoost algorithm, the conventional cascade classifier, and asymmetric boosting. Section 3 describes the proposed system and its algorithmic flow. Experimental results showing the performance of the proposed system are presented in Section 4. Conclusions are drawn in Section 5.

2 Background and related works

In this section, a background is provided that is required for the understanding of the proposed system. The AdaBoost and conventional cascade learning algorithms are briefly described. Asymmetric boosting is then presented together with related research works.

2.1 AdaBoost and cascade learning

AdaBoost is known to provide a principled and highly efficient mechanism for feature selection [12, 13]. In AdaBoost, a strong classifier is constructed by iterating rounds of weak classifier selection and sample weight updating. The weights are decreased for correctly classified samples and increased for incorrectly classified ones so that AdaBoost can focus on difficult samples [14, 15]. Freund and Schapire proved that the training error of a strong classifier approaches zero exponentially in the number of rounds [16, 17].

Let $x_i \in \boldsymbol{X} = \{x_1, x_2, \ldots, x_N\}$ be an image in image set $\boldsymbol{X}$, and $y_i \in \boldsymbol{Y} = \{+1, -1\}$ be a class label of x_i, where N is the number of images. To initialize the weight of x_i prior to training, it is assigned as in (1)

$$w_i(0) = \begin{cases} 1/2 \cdot N_+ , & \text{if } y_i = +1 \\ 1/2 \cdot N_- , & \text{if } y_i = -1 \end{cases}, \tag{1}$$

where N_+ and N_- are the numbers of positive and negative samples in $\boldsymbol{X}$, respectively.

A weak classifier is selected from set $\mathbf{W}$ as in (2), where $w_i(t)$ denotes the weight of sample x_i at round t.

$$h_t = \arg\min_{h_j \in \boldsymbol{W}} \mathrm{err}_{h_j} , \text{ where } \mathrm{err}_{h_j} = \sum_{i=1}^{N} w_i(t) \cdot \left| y_i - h_j(x_i) \right| \tag{2}$$

The error of each weak classifier is calculated, and the one with the minimum error, h_t, is selected. After a weak classifier is selected, the weights of the samples are updated by applying it as in (3). Z_{t+1}, as shown in (4), is a normalization factor that makes the sum of $w_i(t+1)$ 1.

$$w_i(t+1) = \frac{w_i(t) \cdot \exp(-y_i \cdot h_t(x_i))}{Z_{t+1}} \tag{3}$$

$$Z_{t+1} = \sum_{i=1}^{N} w_i(t) \cdot \exp(-y_i \cdot h_t(x_i)) \tag{4}$$

A strong classifier is constructed through a linear combination of the selected weak classifiers as shown in (5), where T and a_t are the numbers of weak classifiers and the confidence level of $h_t(x_i)$, respectively.

$$H(x_i) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x_i) \right) \tag{5}$$

In the conventional cascade classifier, simple classifiers are employed at early stages to reject the negative samples, whereas complex classifiers are used at later stages to achieve a high detection rate [18, 19]. The structure

of the cascade reflects the fact that within any single image, the overwhelming majority of sub-windows are negative [11]. The bootstrapping procedure [20] is employed to collect negative samples in the training cascade. Each stage uses misclassified samples at its precedent stage by bootstrapping its training samples. Later stages are trained by more difficult negative samples, and the classifiers become more complex.

Given a trained cascade, its detection rate (D) and false positive rate (F) are calculated as in (6) and (7), respectively [11]. Here, k denotes the number of stages in a cascade. d_i and f_i are the detection rate and false positive rate of the i-th stage, respectively.

$$D = \prod_{i=1}^{k} d_i \tag{6}$$

$$F = \prod_{i=1}^{k} f_i \tag{7}$$

For example, to achieve a detection rate of 0.95 and a false positive rate of 6×10^{-6} by a 10-stage cascade, the performance goals of each stage include a detection rate of 0.995 ($0.995^{10} \approx 0.95$) and a false positive rate of 0.3 ($0.3^{10} \approx 6 \times 10^{-6}$). The amount of operations in a detection cascade is estimated by the number of features (n_i) and the positive rate (p_i) of the i-th stage as shown in (8).

$$N = n_1 + \sum_{i=2}^{k} \left(n_i \prod_{j<i} p_j \right) \approx n_1 + \sum_{i=2}^{k} \left(n_i \prod_{j<i} f_j \right) \tag{8}$$

The positive rate (p_j), which is the rate of true and false positives against the number of samples, can be approximated to f_j, as positive samples are extremely rare in detection environments [11].

2.2 Asymmetric boosting

Asymmetric boosting has been proposed by several researchers [12, 17, 21–24]. It assigns larger weights to positive samples than negative ones to overcome the problem that false positives and false negatives are treated without distinction in AdaBoost. According to Viola and Jones, one limitation of AdaBoost arises in the rare occurrence of positive samples [9]. The impact of misclassified positive samples is much higher than that of misclassified negative ones, and rejected positive samples cannot be restored in a cascade classifier. This necessitates that false negatives be minimized. Various researchers have proposed methods to improve the performance of the AdaBoost algorithm through asymmetric weight assignment. The weight update of AdaBoost was modified so that false negatives are assigned larger weight than false positives [12, 17, 21, 22, 24].

Viola and Jones [12] applied asymmetry to AdaBoost by assigning k times larger weight to false negatives than false positives. Fan et al. [21] proposed AdaCost, a cost-sensitive extension of AdaBoost, in which false negatives are assigned larger weights than false positives and true negatives are assigned smaller weights than true positives. Ting [22] proposed a method that assigns larger weights to false negatives based on a cost function that reflects the importance of misclassified samples. While these methods achieve higher detection performance, the weight update is heuristic, making it difficult to predict performance prior to in-field use.

Landesa-Vázquez and Alba-Castro [23] showed that asymmetric boosting can be achieved by assigning a larger initial weight to positive samples. Given an asymmetry parameter of γ, the initial weight of the i-th sample, w_i, is calculated as in (9).

$$w_i = \begin{cases} \gamma / N_+ & , \text{ if } y_i = +1 \\ (1-\gamma) / N_- & , \text{ if } y_i = -1 \end{cases} \tag{9}$$

They did not suggest any method that derives an optimal asymmetry parameter of γ for performance improvement.

3 Proposed system

In the cascade learning process, samples are prepared at each stage with bootstrapping. The bootstrapping procedure collects samples, which consist of misclassified examples of its previous stage as well as additional ones prepared for training. It makes negative samples more difficult. The training cascade is set to focus on negative samples as the stage goes on, and the selected weak classifiers become well-trained to negative samples rather than positive ones. This fact can be confirmed by observing the detection rate at each cascade stage of an object detector based on the Viola and Jones algorithm, as illustrated in Fig. 1. Although each stage was trained to achieve the performance goal for detection rate (0.998), there is a gap in the observed detection rate, and the gap becomes larger as the stage proceeds. This is due to the increased difficulty of classifying negative samples. To balance the difficulty between the positive and negative samples, larger weights should be assigned to the positive samples. In this work, the weights are adjusted so that the total weight of the positive samples is comparable to that of the negative samples.

The proposed system takes the form of the conventional cascade. The difference is in the structure and operation of the constituent stages, especially in the training process. Each stage of a conventional cascade performs feature (weak classifier) selection. Additionally, the proposed cascade performs an evaluation of the selected weak classifiers to assess their effectiveness. Although the selected weak classifiers might achieve the performance goal, they

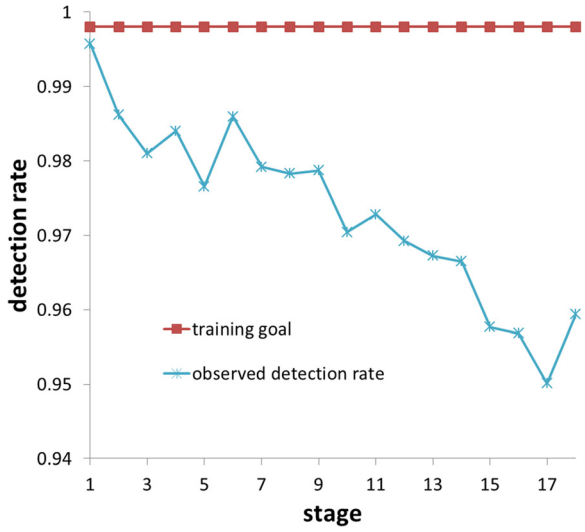

Fig. 1 Training goal and observed detection rate of each stage

were trained using the samples collected by bootstrapping. Therefore, the weak classifiers tend to become biased as unbalance between the positive and negative samples in the training set increases. In detection environments, false negatives are much more critical than false positives, while objects are less frequent than non-objects. Thus, non-biased weak classifiers are desired. The estimated bias in the selected weak classifiers is used to adjust the weights in the training samples. By adjusting the initial weight of the training samples prior to training, the unbalance between positive and negative samples in the training set is reduced, and the weak classifiers selected by using weight-adjusted samples are less biased. In the proposed system, this is handled as follows: By applying weak classifiers to another set of samples, *the reservoir set*, the weights of these samples are updated with the same process that updates the weights of the samples in training. The degree of weight disparity in the positive and negative samples is estimated with the weight ratio, which is the ratio of the total weight assigned to positive to negative samples. In this research, the weight ratio is used as a reference to estimate to what extent the selected weak classifiers are biased to positive or negative samples. The training is then restarted using the training samples whose weights are adjusted by the weight ratio. If the positive and negative samples in the training set are balanced, the weight ratio converges to 1. This process prevents the detection rate by the selected weak classifiers from decreasing in later stages; thus, the detection performance of the cascade classifier improves.

Figure 2 shows the algorithmic flow of the proposed system for training a cascade stage. First, the weights of the samples in the training set are initialized and a cascade stage is trained using the provided samples. In training, weak classifiers that can achieve the given performance goal are selected. The selected weak classifiers may satisfy the performance goal for the training set, but they may not be efficient when used for detecting vehicles. The samples in the training set are collected with bootstrapping, and the training samples are unbalanced. In the proposed system, to estimate bias, the efficiency of the selected weak classifiers is assessed by applying them to different samples called the reservoir set. The weights of the samples in the reservoir set are initialized and updated by applying the selected weak classifiers. The weights updated by applying $(t+1)$-th weak classifier, $w_j'(t+1)$ and y_j' denote the j-th sample and its class sample in the reservoir set, respectively.

$$w_j'(t+1) = w_j'(t)\cdot \exp\left(-y_j'\cdot h_t\left(x_j'\right)\right) \tag{10}$$

M denotes the number of weak classifiers selected at a cascade stage, and the updated weight of j-th sample in the reservoir set, w_j', is calculated as (11).

$$w_j' = w_j'(0)\cdot \exp\left(-y_j'\cdot \sum_{t=0}^{M-1} h_t\left(x_j'\right)\right) \tag{11}$$

Equation (12) shows that the sums of the weights assigned to positive and negative samples are S_+ and $S\_$, respectively, where N' denotes the number of samples in the reservoir set.

$$\begin{aligned} S_+ &= \sum_{j=1}^{N'} w_j' \quad \text{if } y_j' = +1 \\ S\_ &= \sum_{j=1}^{N'} w_j' \quad \text{if } y_j' = -1 \end{aligned} \tag{12}$$

The weight ratio is obtained as in (13).

$$\mathrm{WR} = {}^{S_+}\!/_{S\_} \tag{13}$$

If WR converges to 1, the training will be terminated and proceed to a subsequent stage. Otherwise, the weights of the samples in the training set are adjusted by WRcp, the cumulated product of WR, using (14), and all the selected weak classifiers will be discarded.

$$w_i(0) = \begin{cases} {}^{\mathrm{WRcp}}\!/_{\left(1+\mathrm{WR_{cp}}\right)\cdot N_+}, & \text{if } y_i = +1 \\ {}^{1}\!/_{\left(1+\mathrm{WR_{cp}}\right)\cdot N\_}, & \text{if } y_i = -1 \end{cases} \tag{14}$$

The training for the corresponding stage is restarted. This process is repeated until the weight ratio converges to 1 or there is no further improvement.

Given:

- Training samples $(x_0, y_0), \ldots, (x_{N-1}, y_{N-1})$, where image $x_i \in \boldsymbol{X}$, class label $y_i \in \boldsymbol{Y} = \{+1, -1\}$, and N is number of samples.
- Reservoir samples $(x'_0, y'_0), \ldots, (x'_{N'-1}, y'_{N'-1})$, where image $x'_j \in \boldsymbol{X}'$, class label $y'_j \in \boldsymbol{Y} = \{+1, -1\}$, and N' is number of samples.
- All weak classifiers $\boldsymbol{H} = \{ h_0, \ldots h_{M-1} \}$ at present stage, where M is number of weak classifiers.
- Target detection rate and target false positive rate for a stage are D_{target} and F_{target}, respectively.
- Iteration count, *count*, is initialized to 0 at the beginning of a stage.
- The weight ratio and the cumulated product of weight ratio are initialized to 1 at the beginning of a stage. (weight ratio = sum of weight of positive samples over negative samples)

Step 1. Initialize

Initialize the weight of each sample in training set , $\boldsymbol{X}$

for i=0,…,N-1 do

$$w_i(0) = \begin{cases} 1/2 \cdot N_+ & , if\ y_i = +1 \\ 1/2 \cdot N_- & , if\ y_i = -1 \end{cases}$$

, where N_+ and N_- are the numbers of positive and negative samples in $\boldsymbol{X}$, respectively.

Step 2. Select weak classifiers (conventional cascade stage)

Generate weak classifier set $\boldsymbol{H}$ whose application to $\boldsymbol{X}$ can achieve performance goals D_{target} and F_{target}.

Step 3. Assess effectiveness of selected weak classifiers

Initialize and update the weights of samples in reservoir set, $\boldsymbol{X}'$ using $\boldsymbol{H}$ and calculate weight ratio.

3-1. Initialize the weight of each sample in $\boldsymbol{X}'$

for j=0,…, N'-1 do

$$w'_j(0) = \begin{cases} 1/2 \cdot N'_+ & , if\ y'_j = +1 \\ 1/2 \cdot N'_- & , if\ y'_j = -1 \end{cases}$$

, where N'_+ and N'_- are the numbers of positive and negative samples in $\boldsymbol{X}'$, respectively.

3-2. Apply $\boldsymbol{H}$ to each sample in $\boldsymbol{X}'$ and update the weights of samples, w'_j.

for j=0,…, N'-1do

$$w'_j = w'_j(0) \cdot exp\left(-y'_j \cdot \sum_{t=0}^{M-1} h_t(x'_j) \right)$$

3-3. Calculate the weight ratio by sum of weights of positive and negative samples in $\boldsymbol{X}'$.

sum of the weights of positive samples, $S_+ = \sum_{j=0}^{N'-1} w'_j \ \ if\ y'_j = +1$

sum of the weights of negative samples, $S_- = \sum_{j=0}^{N'-1} w'_j \ \ if\ y'_j = -1$

weight ratio, $WR = S_+ / S_-$

cumulated product of weight ratio, $WR_{cp} = WR_{cp} \cdot S_+ / S_-$

3-4. If WR converges to 1, exit.

If *count* exceeds predetermined iteration count, $\boldsymbol{H} \leftarrow \boldsymbol{H}_{\text{save}}$ and exit.

Else go to Step 4

Step 4. Adjust weights of training samples

If WR is better than saved WR_{save}, $WR_{save} = WR$ and $\boldsymbol{H}_{\text{save}} \leftarrow \boldsymbol{H}$.

Adjust weights of samples in $\boldsymbol{X}$ and initialize $\boldsymbol{H}$ to empty set.

for i=0,…,N-1 do

$$w_i(0) = \begin{cases} WR_{cp} / \left(1 + WR_{cp}\right) \cdot N_+ & , if\ y_i = +1 \\ 1 / \left(1 + WR_{cp}\right) \cdot N_- & , if\ y_i = -1 \end{cases}$$

count ++

go to Step 2

Fig. 2 Algorithmic flow of the proposed system

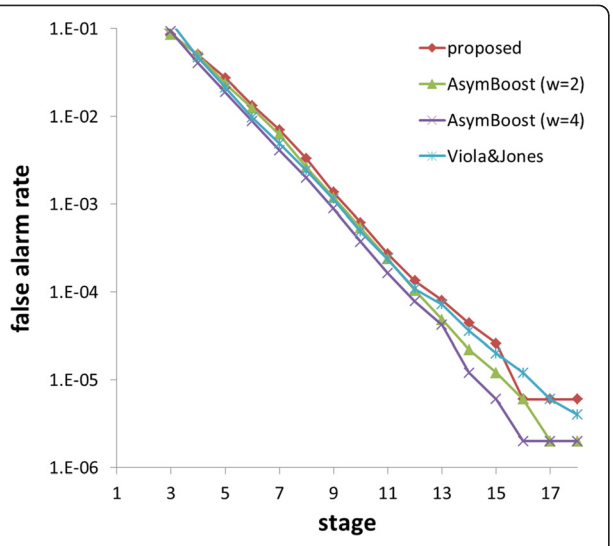

Fig. 3 False alarm rate of classifiers at the end of each stage

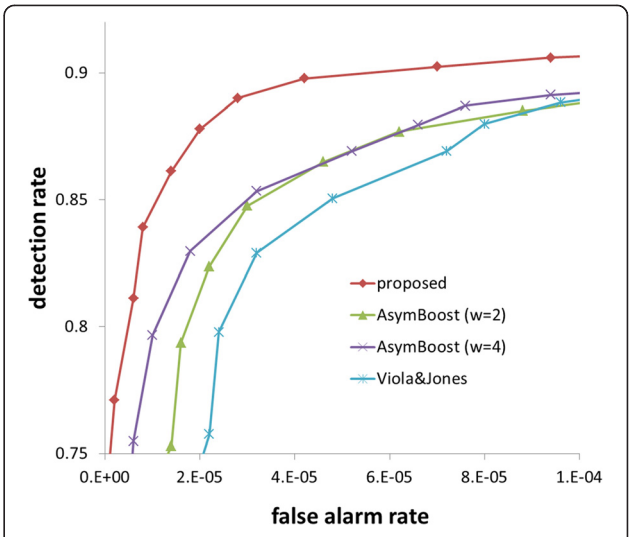

Fig. 5 ROC curves of classifiers trained by different algorithms

4 Experimental results

To show the effectiveness of the proposed system, a series of experiments were performed. For the positive samples, 26,000 vehicle images were obtained by cropping the images captured on the road with a camera mounted on vehicle. All of the vehicle samples were resized to 20 × 20 pixels. For each stage of the cascades, 6,000 samples were used for training. A half of remaining 20,000 samples were used as the reservoir set, and the other half were used to evaluate the detection performance of the weak classifiers employing the proposed system. Over 500,000 negative samples were prepared by randomly cropping 1,085 background images that did not contain any vehicle objects. These samples were included in the training set, the test set, and the reservoir set.

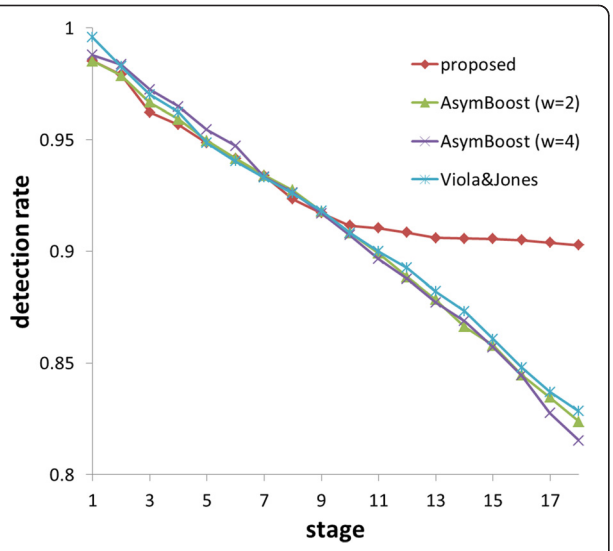

Fig. 4 Detection rate of classifiers at the end of each stage

Vehicle detection systems were trained by three different approaches: the conventional cascade learning approach, the asymmetric boosting approach, and the proposed system. Each stage in the cascade classifiers was trained to achieve a detection rate of 0.998 and a false alarm rate of 0.5. The detection system employing the asymmetric boosting approach was trained by assigning the initial weights to the training samples such that the sum of the positive sample weights is two times and four times greater than that of the negative sample weights. In the proposed system, the weight is assigned by extracting the latent asymmetry at each stage in the cascade classifier. The termination condition of each stage is set to have the weight ratio of 1.

Figures 3 and 4 show the detection rate and the false alarm rate of the cascade classifiers trained employing the different approaches evaluated at the end of each stage, respectively. The cascade classifier trained employing the conventional cascade learning approach is labeled "Viola&Jones", and those trained employing the asymmetric boosting approach with the asymmetric weights of 2 and 4 are labeled "AsymBoost($W = 2$)" and "AsymBoost($W = 4$)", respectively. Generally, cascade classifier characteristics entail that the detection rate and false alarm rate decrease as the stage moves forwards. The

Table 1 Comparison of number of operations

Methods	Number of operations	Comparison w/ Viola&Jones
Viola&Jones	1,049,758	–
AsymBoost($W = 2$)	904,741	−13.8 %
AsymBoost($W = 4$)	945,914	−9.9 %
Proposed	859,821	−18.1 %

detection rate and false alarm rate of the cascade with 15 stages was approximately 0.97 ($0.998^{15} \approx 0.97$) and 3×10^{-5} ($0.5^{15} \approx 3 \times 10^{-5}$), respectively. While all the approaches achieved the performance goal for false alarm rate, they failed to achieve the detection rate goal. The detection rate of the proposed system decreased slowly as the stages proceeded when compared with the others. With less stages, the cascade classifier based on the proposed system can achieve the required detection performance.

Figure 5 shows the receiver operating characteristic (ROC) curves of different classifiers trained in the cascade consisting of 12 stages. The cascade classifier employing the proposed system shows significantly improved performance when compared with the others.

Table 1 shows the number of operations required for rejecting 100,000 negative sub-windows. It is calculated by using the false alarm rate and number of features, as shown in (8). The number of operations is also reduced in the proposed system compared to the other approaches. This is due to the fact that the number of features used in the early cascade stages is smaller, even though more features are used in all the cascade stages of the proposed system in the detection environments.

5 Conclusions

To obtain a high-performance cascade classifier, weak classifiers constituting each stage should be selected considerately, even with increased training time. Weak classifiers not biased toward either positive or negative samples are desired. This is due to the fact that false negatives are much more critical than false positives while objects are less frequent than non-objects in detection environments. The selected weak classifiers in conventional cascade classifiers are biased, and they lead to a gap between performance goal and observed performance at each stage.

In this paper, a cascade classifier is proposed in which the selected weak classifiers are confirmed for their effectiveness in detection process. The main factor contributing to biased weak classifiers is unbalanced training samples prepared by the bootstrapping procedure. To assess the bias of the selected weak classifiers, they are applied to the samples in the reservoir set to update their weights. The disparity between the total weight of positive and negative samples in the reservoir set indicates the degree of unbalance in the positive and negative training samples. If they are found to be unbalanced, the weights of the training samples are adjusted and the weak classifier selection process is restarted. Experimental results confirm the effectiveness of the proposed cascade classifier in vehicle detection by showing an improved performance over conventional ones.

Competing interests
The authors declare that they have no competing interests.

Received: 25 February 2015 Accepted: 2 June 2015
Published online: 06 June 2015

References

1. Z Sun, G Bebis, R Miller, On-road vehicle detection: a review. IEEE Trans. Pattern Anal. Mach. Intell. **28**(5), 694–711 (2006)
2. C Papageorgiou, T Poggio, A trainable system for object detection. Int. J. Comput. Vis. **38**(1), 15–33 (2000)
3. Z Sun, G Bebis, R Miller, On-road vehicle detection using Gabor filters and support vector machines, in *Proceedings of 14th Int. Conf. Digital Signal Processing* (IEEE, Santorini, Greece, 2002), pp. 1019–1022
4. J Cui, F Liu, Z Li, Z Jia, Vehicle localization using a single camera, in *Proceedings of IEEE Intelligent Vehicle Symposium* (IEEE, San Diego, CA, 2010), pp. 871–876
5. PF Felzenszwalb, RB Girshick, D McAllester, D Ramanan, Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010)
6. Y Li, B Tian, B Li, G Xiong, F Zhu, Vehicle detection with a part-based model for complex traffic conditions, in *Proceedings of 14th IEEE Int. Conf. Vehicular Electronics and Safety* (IEEE, Dongguan, China, 2013), pp. 110–113
7. LC Leon, R Hirata, Vehicle detection using mixture of deformable parts models: static and dynamic camera, in *Proceedings of 25th SIBGRAPI Conf. Graphics, Patterns and Images* (IEEE, Ouro Preto, Brazil, 2012), pp. 237–244
8. S Sivaraman, MM Trivedi, Real-time vehicle detection using parts at intersections, in *Proceedings of 15th Int. IEEE Conf. Intelligent Transportation Systems* (IEEE, Aleutian, Alaska, 2012), pp. 1519–1524
9. PM Daigavane, PR Bajaj, Vehicle detection and neural network application for vehicle classification, in *Proceedings of Int. Conf. Computational Intelligence and Communication Networks* (IEEE, Gwalior, India, 2011), pp. 758–762
10. O Ludwig, U Nunes, Improving the generalization properties of neural networks: an application to vehicle detection, in *Proceedings of 11th Int. IEEE Conf. Intelligent Transportation Systems* (IEEE, Beijing, China, 2008), pp. 310–315
11. P Viola, M Jones, Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
12. P Viola, M Jones, Fast and robust classification using asymmetric AdaBoost and a detector cascade, in *Proceedings of Advances in Neural Information Processing Systems* (NIPS, Vancouver, Canada, 2001), pp. 1311–1318
13. Y Freund, RE Schapire, A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Sys. Sci. **55**(1), 119–139 (1997)
14. R Duda, P Hart, D Stork, *Pattern classification*, 2nd edn. (Wiley-Interscience, New York, 2001)
15. Y Freund, RE Schapire, A short introduction to boosting. J. Jpn. Soc. Artif. Intell. **14**(5), 771–780 (1999)
16. P Viola, M Jones, Rapid object detection using a boosted cascade of simple features, in *Proceedings of IEEE Computer Society Conf. Computer Vision and Pattern Recognition* (IEEE, Kauai, HI, 2001), pp. 511–518
17. H Masnadi-Shirazi, N Vasconcelos, Asymmetric boosting, in *Proceedings of Int. Conf. Machine Learning* (ACM, Corvallis, OR, 2007), pp. 609–619
18. R Xiao, L Zhu, HJ Zhang, Boosting chain learning for object detection, in *Proceedings of IEEE Int. Conf. Computer Vision* (IEEE, Nice, France, 2003), pp. 709–715
19. J Wu, SC Brubaker, MD Mullin, JM Rehg, Fast asymmetric learning for cascade face detection. IEEE Trans. Pattern Anal. Mach. Intell. **30**(3), 369–382 (2008)
20. KK Sung, T Poggio, Example-based learning for view-based human face detection. IEEE Trans. Pattern Anal. Mach. Intell. **20**(1), 39–51 (1998)
21. W Fan, SJ Stolfo, J Zhang, PK Chan, AdaCost: misclassification cost-sensitive boosting, in *Proceedings of 16th Int. Conf. Machine Learning* (ICML, Bled, Slovenia, 1999), pp. 97–105
22. KM Ting, A comparative study of cost-sensitive boosting algorithms, in *Proceedings of 17th Int. Conf. Machine Learning* (ICML, Stanford, CA, 2000), pp. 983–990
23. I Landesa-Vázquez, JL Alba-Castro, Shedding light on the asymmetric learning capability of AdaBoost. Pattern Recognit. Lett. **33**(3), 247–255 (2012)
24. Y Sun, MS Kamel, AKC Wong, Y Wang, Cost-sensitive boosting for classification of imbalanced data. IEEE Trans. Pattern Anal. Mach. Intell. **40**(12), 3358–3378 (2007)