TRAINING A MULTI-EXIT CASCADE WITH LINEAR ASYMMETRIC CLASSIFICATION FOR EFFICIENT OBJECT DETECTION

Peng Wang[1], Chunhua Shen[2,3], Hong Zheng[1], Zhang Ren[1]

[1] Beihang University, Beijing 100191, China
[2] NICTA, Canberra Research Laboratory, Canberra ACT 2601, Australia
[3] Australian National University, Canberra ACT 0200, Australia

ABSTRACT

Efficient visual object detection is of central interest in computer vision and pattern recognition due to its wide ranges of applications. Viola and Jones' detector has become a *de facto* framework [1]. In this work, we propose a new method to design a cascade of boosted classifiers for fast object detection, which combines linear asymmetric classification (LAC) into the recent multi-exit cascade structure. Therefore, the proposed method takes advantages of both LAC and the multi-exit cascade. Namely, (1) the multi-exit cascade structure collects all the scores of prior nodes for decision making at the current node, which reduces the loss of decision information; (2) LAC considers the asymmetric nature of the node training. We also show that the multi-exit cascade better meets the assumption of LAC learning than the standard Viola-Jones' cascade, both theoretically and empirically. Experiments confirm that our method outperforms existing methods such as Viola and Jones [1] and Wu *et al.* [2] on the MIT+CMU test data set.

***Index Terms*—** Face detection, Boosting, Linear Asymmetric Classifier, Cascade Classifier

1. INTRODUCTION

Real-time object detection [1, 3–7] has many applications in computer vision such as video surveillance, visual teleconference and image analysis. The cascaded boosting framework was firstly introduced to object detection by Viola and Jones [1, 8], which has been viewed as a significant progress on face/object detection. Other successful detectors include neural network based methods [3] and support vector machine [5, 9]. We build our work upon the general detection framework of [1].

Generally speaking, there are three aspects that lead to the great success of Viola-Jones face detector: Haar feature, AdaBoost and the cascade classifier structure. Haar feature is a simple but informative descriptor for appearance-based image. Owing to the usage of the concept of "integral image", it costs only a few operations to calculate a single Haar feature, which results in an extremely fast detection speed. The set of Haar features is over-complete but involving all aspect ratios and locations, which make it appropriate for scaled sub-window scanning.

In Viola-Jones framework, AdaBoost is used for both feature selection and ensembled classifier learning. AdaBoost is an aggressive and effective mechanism for feature selection. During the course of learning, it iteratively selects weak classifiers, which corresponds to features when decision stumps[1] are used as weak classifiers. Finally, a small number of discriminative features are selected from the huge amount of candidate pool, and only those selected features will be evaluated at the test phase. At the same time, AdaBoost assigns weights with respect to weak classifiers to form a strong classifier, using line search.

The cascade structure used by Viola and Jones can be seen as a degenerate decision tree, which is made up of a sequence of boosting classifiers (called "nodes", see Fig. 1(a)). A positive decision made by one node will trigger the evaluation of next node while a negative decision made by any node will result in a rejection. Only those sub-windows passing through all nodes would be treated as positive detections. In practice, most of sub-windows are rejected by early nodes, which make the face detector much more efficient. On the other hand, equipped with bootstrapping, the cascade structure is used to cope with the asymmetry of the highly-skewed face and non-face data. Bootstrapping makes it possible to utilize a large number of negative non-face data for training. Since the number of negative examples is greatly large than positive data, we can not collect sufficient negative examples at one time. Before adding a new node, negative examples rejected by current cascade are removed from training set, and new negative examples misclassified by the current cascade are loaded from pool for training next node. Thus, the actual number of negative examples taken into consideration can be extremely large.

Much incremental work has been advocated to improve the performance of Viola-Jones standard framework. Roughly, there are two important types of motivations among those methods. One is introducing asymmetric learning mechanism into node learning [2, 8]. Although the cascade structure has alleviated the asymmetry problem to some extend, our goal of training a node in the cascade is still highly imbalanced: we wish to achieve a extremely high detection rate (*e.g.* 99.9%) and a moderate false positive rate (*e.g.* around 50%). However, most boosting algorithms are not designed for this learning goal [10]. One approach is to modify the example weighting strategy within boosting algorithms as [8] did. Another way is separating the node learning process into two stages (feature selection and strong classifier learning) and then apply asymmetric learning method on either (or both) of those two stages [2, 11]. LAC is one of the typical examples along this line [2].

The other motivation is to utilize scores made by previous nodes for the current node prediction. In Viola-Jones standard cascade structure, the information derived by previous nodes is discarded when it passes to the next node. Work has been done to use these information to obtain a better performance and higher detection speed. Empirical results indicate that these methods can obtain better ROC curves, and sometimes fewer features are needed [6, 12–15]. Pham *et al.*'s multi-exit cascade is considered the state-of-the-art [12].

Directly inspired by the work of LAC and the multi-exit cascade, we introduce a new boosting cascade framework for face detection. We have designed a *simplified* multi-exit cascade for training a de-

NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Center of Excellence program. The first author's contribution was made when visiting NICTA, Canberra Research Laboratory and Australian National University.

[1] Decision stumps are weak classifiers that use only one individual feature.

tector. As for node training, we make use of LAC as an alternative for ensemble classifier learning. Through the analysis on experimental data, we find that the multi-exit cascade structure better exploit the potential of linear asymmetric classifiers. We also compare our approach with other relevant approaches on the standard MIT+CMU data set, which shows our the presented framework achieves better performances than others.

The main contributions of this work are two-fold. (1) To our knowledge, it is the first time to combine the linear asymmetric classifier (LAC) and the multi-exit cascade structure for detection, and we achieve a better detection performance than existing work. (2) We have also analyzed the condition that makes the validity of LAC, and explained why the multi-exit cascade structure is more suitable for LAC rather than Viola-Jones standard cascade structure.

2. ALGORITHMS

2.1. Linear Asymmetric Classification

Wu *et al.* [2] have proposed a post-processing step for training nodes in the cascade framework, called linear asymmetric classification (LAC). LAC is guaranteed to get an optimal solution under the assumption of a specific Gaussian data distribution.

Suppose that we have a linear classifier $\boldsymbol{H}(\boldsymbol{z}) = \mathbf{sign}(\boldsymbol{a}^T\boldsymbol{z} - b)$, if we want to find a combination of $\boldsymbol{a}$ and b with a very high classification accuracy on positive class $\boldsymbol{x}$ and a moderate accuracy on negative class $\boldsymbol{y}$, which is expressed as the following problem:

$$\max_{\boldsymbol{a}\neq 0, b} \Pr_{\boldsymbol{x}\backsim(\overline{x},\sum_x)} \{\boldsymbol{a}^T\boldsymbol{x} \geq b\}, \text{ s.t. } \Pr_{\boldsymbol{y}\backsim(\overline{y},\sum_y)} \{\boldsymbol{a}^T\boldsymbol{y} \geq b\} = \beta, \quad (1)$$

where $\boldsymbol{x} \backsim (\overline{x}, \sum_x)$ denotes the class with fixed mean $\overline{x}$ and covariance $\sum_x$; likewise for $\boldsymbol{y}$. If we prescribe β to 0.5 and assume that for any $\boldsymbol{a}$, $\boldsymbol{a}^T\boldsymbol{x}$ is Gaussian and $\boldsymbol{a}^T\boldsymbol{y}$ is symmetric, then a close-formed optimal solution is guaranteed:

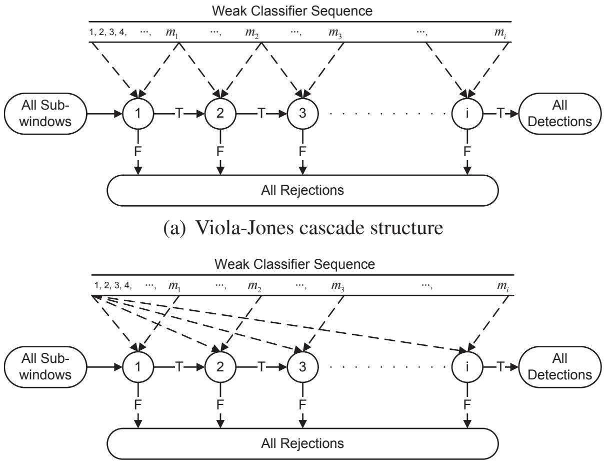
$$\boldsymbol{a}^\star = \textstyle\sum_{\boldsymbol{x}}^{-1}(\overline{\boldsymbol{x}} - \overline{\boldsymbol{y}}), \quad b^\star = \boldsymbol{a}^{\star T}\overline{\boldsymbol{y}}. \quad (2)$$

On the other hand, each node in cascaded boosting classifiers has the following form:

$$\boldsymbol{H}(\boldsymbol{z}) = \mathbf{sign}(\boldsymbol{a}^T\boldsymbol{h}(\boldsymbol{z}) - b), \quad (3)$$

where $\boldsymbol{h}(\cdot)$ denotes the output vector of all weak classifiers. We can cast each node as a linear classifier over the feature space constructed by the binary outputs of all weak classifiers. For each node in cascade classifier, we wish to maximize the detection rate as high as possible (*e.g.* 99.9%), and meanwhile keep the false positive rate to an acceptable level (*e.g.* 50.0%). That is to say, the node learning goal coincides with the problem (1). Therefore, we can use boosting algorithms (*e.g.* AdaBoost) as feature selection methods, and then use LAC to learn a linear classifier over those binary features chosen by boosting.

However, there is a precondition of LAC's validity. That is, for any $\boldsymbol{a}$, $\boldsymbol{a}^T\boldsymbol{x}$ is Gaussian and $\boldsymbol{a}^T\boldsymbol{y}$ is symmetric. In the case of boosting classifiers, $\boldsymbol{a}^T\boldsymbol{x}$ and $\boldsymbol{a}^T\boldsymbol{y}$ can be expressed as the margin of positive data and negative data. From empirical data, Wu *et al.* [2] verified that $\boldsymbol{a}^T\boldsymbol{x}$ is Gaussian approximately for a cascade face detector. Shen *et al.* [16] theoretically proved that under the assumption that weak classifiers are independent, the margin of AdaBoost follows the Gaussian distribution, as long as the number of weak classifiers is *sufficiently large*. Here we verify this theoretical result by performing the normality test on nodes with different number of weak classifiers in the experiment section.



(a) Viola-Jones cascade structure

(b) The multi-exit cascade structure

Fig. 1: A description of the two cascade structures. In (a), the ith node of Viola-Jones cascade exclusively use weak classifiers from index $m_{i-1}+1$ to m_i; For the multi-exit cascade in (b), each node has shared weak classifiers with previous nodes.

2.2. The multi-exit cascade

Pham *et al.* [12] introduced a generalized cascade framework, termed the multi-exit cascade (See Fig. 1(b)). The multi-exit cascaded boosting classifier is expressed as follow:

$$F(\boldsymbol{z}) = \begin{cases} +1 & \text{if } \sum_{t=1}^m w_t h_t(\boldsymbol{x}) + \theta_m \geq 0, \forall m \in \mathcal{M}; \\ -1 & \text{otherwise.} \end{cases} \quad (4)$$

In the multi-exit framework, the ith node combines the scores from the first weak classifier to the ith exit. Note that the soft cascade [6] and the dynamic cascade [13] can be viewed as special cases of the multi-exit cascade, in which $\mathcal{M} = \{1, \ldots, M\}$. The desirable property of the multi-exit cascade is that, each node takes into account the historical information. Consequently, given the same number of weak classifiers, the multi-exit cascade can utilize more information than the Viola-Jones standard cascade. Fig. 1 demonstrates the difference between Viola-Jones standard cascade and the multi-exit cascade structure. To simplify the training process, we pre-set the number of weak classifiers of each node before training, rather than determine the number during training.

2.3. Combining LAC into the multi-exit cascade

In this section, we combine LAC into the multi-exit cascade for a better performance. Algorithm 1 shows the procedure of training a multi-exit cascade with LAC.

The final classifier of the "multi-exit plus LAC" framework can be expressed as:

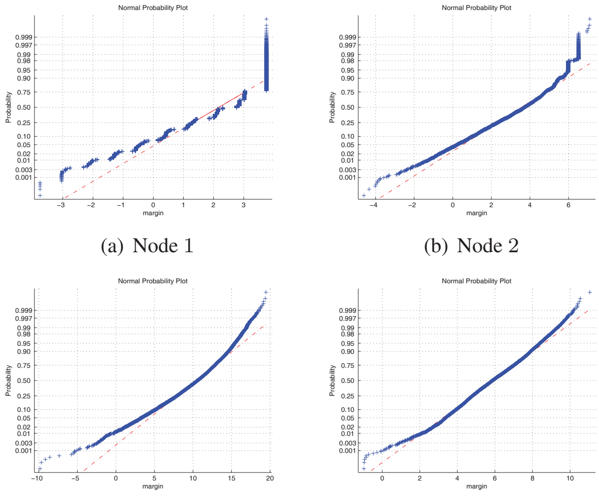
$$F(\boldsymbol{z}) = \begin{cases} +1 & \text{if } \sum_{t=1}^m w_{mt} h_t(\boldsymbol{z}) + \theta_m \geq 0, \forall m \in \mathcal{M} \\ -1 & \text{otherwise.} \end{cases} \quad (5)$$

The major difference between (4) and (5) is that, nodes in the multi-exit cascade share both the weak classifiers and the corresponding weights; while for the multi-exit cascade with LAC, nodes only share the weak classifiers but have their own sequence of weights with respect to weak classifiers.

In our multi-exit plus LAC framework, boosting algorithms are used as feature selection methods and LAC is used for training the final ensemble classifier. LAC re-assign weights corresponding to

Algorithm 1 The procedure for training a multi-exit cascade with LAC.

```
Input:
    • A training set with $N_x$ positive and $N_y$ negative examples;
    • $D_{\min}$: minimum acceptable detection rate per node;
    • $F_{\rm target}$: target overall false positive rate.
Initialize: $i = 0$; $M = 0$; $D_i = 1$; $F_i = 1$;
while $F_{\rm target} < F_i$ do
    $i = i + 1$; $f_i = 1$;
    while $f_i > F_{\max}$ do
        1. $M = M + 1$.
        2. Feature selection: train a new weak classifier $\boldsymbol{h}_M(\cdot)$
        using AdaBoost.
        3. Ensemble classifier learning: train LAC over sequence
        $\boldsymbol{h} = [h_1, h_2, \cdots, h_M]$.
        $\overline{\boldsymbol{x}} = \frac{\sum_{n=1}^{N_x} \boldsymbol{h}(x_n)}{N_x}, \quad \overline{\boldsymbol{y}} = \frac{\sum_{n=1}^{N_y} \boldsymbol{h}(y_n)}{N_y},$
        $\sum_{\boldsymbol{x}} = \frac{\sum_{i=1}^{N_x} (\boldsymbol{h}(\boldsymbol{x}_n) - \overline{\boldsymbol{x}})(\boldsymbol{h}(\boldsymbol{x}_n) - \overline{\boldsymbol{x}})}{N_x},$
        then apply (2) to get
        $\boldsymbol{a}_M = \sum_{\boldsymbol{x}}^{-1} (\overline{\boldsymbol{x}} - \overline{\boldsymbol{y}}), \quad b_M = \boldsymbol{a}_M{}^T \overline{\boldsymbol{y}}.$
        4. Adjust threshold $b_M$ of current boosted classifier such
        that $d_i = 0.5$.
        5. Update $f_i$ using this classifier threshold.
    $D_{i+1} = D_i \times d_i$; $F_{i+1} = F_i \times f_i$; $m_i = M$;
    Remove correctly classified negative samples from negative
    training set.
    if $F_{\rm target} < F_i$ then
        Evaluate the current cascaded classifier on the negative
        images and add misclassified samples into the negative
        training set;
Output:
    • A sequence of $M$ weak classifiers;
    • A cascade of boosting classifiers with same entrance index (1) and
      different exit indices $(m_1, m_2, \ldots, m_i)$;
```



(a) Node 1

(b) Node 2

(c) Node 3

(d) Node 22

Fig. 2: Normality test for the margin distribution of node 1, 2, 3 and 22 in the multi-exit cascade with LAC, which has 7, 22, 52 and 2932 weak classifiers respectively. An exact Gaussian distribution forms a straight line. Close to a straight line indicates close to a Gaussian.

weak classifiers, introducing asymmetric learning into node classifiers. Note that other feature selection methods like fast-forward selection [2] and greedy sparse Fisher discriminant analysis [11] may also be adopted. The proposed multi-exit plus LAC framework has the following benefits: (1) The multi-exit cascade structure collect scores of previous nodes, which discarded by Viola-Jones standard cascade, such that it avoids the loss of information. (2) LAC makes the node learning goal asymmetric, which is more suitable for highly skewed data. (3) Unlike the soft cascade structure [6], in which bootstrapping is performed after learning every weak classifiers, the multi-exit cascade only requires bootstrapping after each exit is trained. Therefore, it refrains from the extra computational burden for bootstrapping.

3. EXPERIMENTS

In our experiments, we obtain 9832 mirrored faces images with size 24×24 used in [1]. The background images are the same as those used by Wu *et al.* [2], which is a collection of 7323 large images without any face.

We use 10000 examples for training boosting algorithm, which is made up of 5000 faces and 5000 non-faces. The 5000 faces are randomly sampled from those 9832 faces, and the remaining 4832 images are used as the validation set. During the whole cascade training process, the positive training set and validation set keep unchanged. For the negative training set, the initial 5000 non-faces are randomly cropped from background images collection. After training each node, only negative examples misclassified by current cascade are retained, and new negative examples are replenished from patches in background images collection which are classified as faces. 7 basic types of Haar-like features are calculated, which are the same as those used by Viola and Jones [1], generating 162336 features for 24×24 images. For each round of weak classifier training, we uniformly sub-sample 10% number of total features.

We evaluate four types of cascade frameworks, which are Viola-Jones cascade, Viola-Jones cascade plus LAC, the multi-exit cascade and the multi-exit cascade plus LAC. For simplicity and fair evaluation, we prescribe both the number of nodes and weak classifiers per node in training. All the four types of cascade classifiers have 22 nodes. For Viola-Jones cascade, we use 7 weak classifiers in the first node to speed up the detection process, and the last node is restricted to has maximum 200 weak classifiers. Totally, 2932 weak classifiers are used for Viola-Jones cascade. For the multi-exit cascade, the number of weak classifiers in the ith node is equal to the total number from 1th to ith node in Viola-Jones cascade. Thus, same number of weak classifiers are used for the multi-exit cascade and Viola-Jones cascade. The training process is performed on a machine with 8 Intel Xeon E5520 CPUs and 32GB RAM, which takes less than three hours to train a multi-exit with LAC cascade.

Fig. 2 illustrates the normal probability plot of margins of positive training data, for node 1, 2, 3 and 22 in the multi-exit plus LAC framework. We can find that the larger number of weak classifiers used, the more closely the margin follows Gaussian distribution. In other words, LAC will obtain more desirable performance, if a larger number of weak classifiers are used; while the performance is poor when the number is too small. For that reason, we do not apply LAC in the first two nodes, due to the margin distribution is far from a Gaussian distribution. Fig. 3 shows false negative rates of 22 nodes in the multi-exit framework and the multi-exit plus LAC framework. We can see that, the multi-exit plus LAC framework has lower false negative rates than the multi-exit framework*over all nodes* (except first two nodes, since we do not apply LAC on these nodes), Moreover, the reduction effect is more explicit when nodes have more weak classifiers. Compared with results in [2], we also find that *the multi-exit cascade with LAC achieves better results than Viola-Jones*

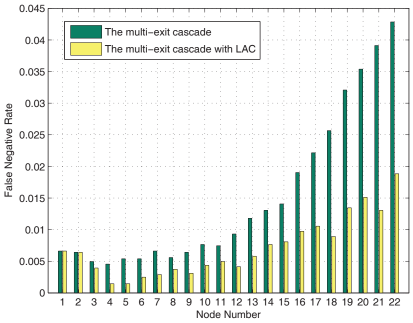


Fig. 3: Comparison of false negative rates between nodes in the multi-exit cascade and the multi-exit cascade with LAC. The x-axis is the number of nodes; the y-axis shows the false negative rate on training set with $\beta = 0.5$.

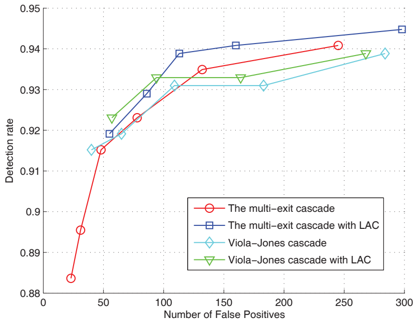


Fig. 4: Comparison of ROC curves with different cascaded boosting frameworks. The x-axis is the number of false positives and the y-axis is the detection rate.

cascade with LAC on reducing false negative rate (52.8% vs 36.1%). It can be interpreted by that the number of weak classifiers of nodes in the multi-exit cascade are larger than those in Viola-Jones cascade, which makes the margin distribution closer to Gaussian. The multi-exit cascade structure reinforce the effectiveness of LAC.

For testing the performance, we use the MIT+CMU frontal face test set, which totally contain 130 images with 507 frontal faces. A detection will be treat as true positive, if its variation of shift and scale from the ground-truth is less than 50%. The post-processing strategy for combining overlapping detections is the same as [1]. The scale factor is set to 1.2 and the stride step is set to 1 pixel. Fig. 4 shows ROC curves of Viola-Jones cascade, Viola-Jones cascade with LAC, the multi-exit cascade, the multi-exit cascade with LAC on the MIT+CMU data set. Overall, the performances from "good" to "poor" are the multi-exit cascade with LAC, the multi-exit cascade, Viola-Jones cascade with LAC and Viola-Jones cascade. *Our method outperforms those only take advantage of LAC or multi-exit.* We show some face detection results of the multi-exit plus LAC framework in Fig. 5.

4. CONCLUSION

In this work, we have presented an alternative method for training a cascade of boosting classifiers. The node training process is decoupled into feature selection and ensemble classifier training, and LAC is used as the strategy for the latter step. A more efficient cascade structure, namely multi-exit cascade, is adopted. These two technique are combined to achieve a better performance. Since LAC has a close-formed solution, the time for training a LAC classifier can be neglected compared with the whole training time. When testing, our framework just introduces a few extra addition and multiplication operations, which also has a small influence on running time. Since Viola and Jones's method has already got an excellent performance for face detection, the improvement of our algorithm is not very significant.

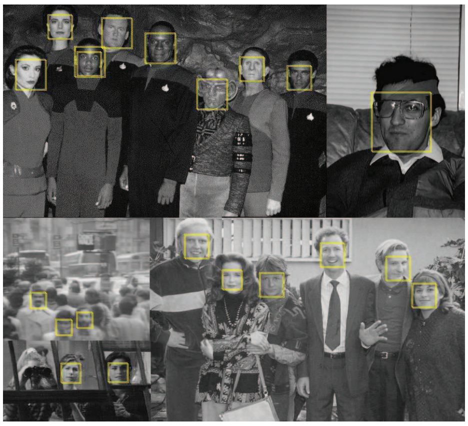


Fig. 5: Face detection examples using the multi-exit plus LAC framework on the MIT+CMU data set.

We have also discussed the validity of LAC. LAC is more effective on nodes with more weak classifiers. In other words, *the multi-exit cascade structure is desirable for applying LAC*, which utilize more information without additional weak classifiers. From these results, we can conclude that the presented framework benefit from both the multi-exit cascade structure and LAC. More importantly, the improvements of multi-exit and LAC are *not overlapped*, whilst they reinforce each other.

References

[1] P. Viola and M. Jones, "Robust real-time face detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2001, vol. 2, pp. 747–747.

[2] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 3, pp. 369–382, 2008.

[3] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23–28, 1998.

[4] C. Shen, S. Paisitkriangkrai, and J. Zhang, "Face detection from few training examples," in *Proc. IEEE Int. Conf. Image Process.*, San Diego, California, USA, 2008.

[5] W. Kienzle, G. Bakir, M. Franz, and B. Schölkopf, "Face detection—efficient and rank deficient," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 673–680.

[6] L. Bourdev and J. Brandt, "Robust object detection via soft cascade," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2005, vol. 2, pp. 236–243.

[7] M.-T. Pham and T.-J. Cham, "Fast training and selection of Haar features using statistics in boosting-based face detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, Rio de Janeiro, Brazil, 2007.

[8] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," *Proc. Adv. Neural Inf. Process. Syst.*, vol. 14, pp. 1311–1318, 2002.

[9] S. Romdhani, P. Torr, B. Schölkopf, and A. Blake, "Computationally efficient face detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, Vancouver, 2001, vol. 2, pp. 695–700.

[10] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[11] S. Paisitkriangkrai, C. Shen, and J. Zhang, "Efficiently learning a detection cascade with sparse eigenvectors," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009.

[12] M. Pham, V. D. Hoang, and T. Cham, "Detection with multi-exit asymmetric boosting," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2008, pp. 1–8.

[13] R. Xiao, H. Zhu, H. Sun, and X. Tang, "Dynamic cascades for face detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2007, pp. 1–8.

[14] R. Xiao, L. Zhu, and H. Zhang, "Boosting chain learning for object detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2003, pp. 709–715.

[15] C. Zhang and P. Viola, "Multiple-instance pruning for learning efficient cascade detectors," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007.

[16] C. Shen and H. Li, "On the dual formulation of boosting algorithms," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2010.