Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2008, Article ID 183804, 11 pages
doi:10.1155/2008/183804

Research Article

DOOMRED: A New Optimization Technique for Boosted Cascade Detectors on Enforced Training Set

Dong Woo Park[1] and Kyoung Mu Lee[2]

[1] *Information & Technology Laboratory, LG Electronics Institute of Technology, 16 Woomyeon-dong, Seocho-gu, Seoul 137-724, Korea*
[2] *Department of Electrical Engineering, ASRI, Seoul National University, 599 Gwanangno, Gwanak-gu, Seoul 151-742, Korea*

Correspondence should be addressed to Kyoung Mu Lee, kyoungmu@snu.ac.kr

Received 31 August 2007; Revised 27 December 2007; Accepted 19 February 2008

Recommended by Olivier Lezoray

We propose a new method to optimize the completely-trained boosted cascade detector on an enforced training set. Recently, due to the accuracy and real-time characteristics of boosted cascade detectors like the Adaboost, a lot of variant algorithms have been proposed to enhance the performance given a fixed number of training data. And, most of algorithms assume that a given training set well exhibits the real world distributions of the target and non-target instances. However, this is seldom true in real situations, and thus often causes higher false-classification ratio. In this paper, to solve the optimization problem of completely trained boosted cascade detector on false-classified instances, we propose a new base hypothesis weight optimization algorithm called DOOMRED (Direct Optimization Of Margin for Rare Event Detection) using a mathematically derived error upper bound of boosting algorithms. We apply the proposed algorithm to a cascade structured frontal face detector trained by AdaBoost algorithm. Experimental results demonstrate that the proposed algorithm has competitive ability to maintain accuracy and real-time characteristic of the boosted cascade detector compared to those of other heuristic approaches while requiring reasonably small amount of optimization time.

1. INTRODUCTION

Recently, the boosted cascade detector [1] became the most popular method for an object detection in computer vision. Due to its accuracy and real-time characteristic, many works have been proposed to enhance the original one [2–4]. However, most researches on the boosted cascade detector have concentrated on the learning problem for a fixed number of initial training data. The basic assumption made in the researches is that the distributions of the target and nontarget objects obtained from the given fixed number of initial training data are good enough to reflect the real distributions, which is seldom true in practice. This is because it is almost impossible to know the exact distribution of the target as well as nontarget instances in real situations. As a result, the detector trained with the fixed number of initial training data cannot work properly in the real applications.

The problem we would like to address in this paper is "what should be done to a completely trained object detector when false-classified instances occur in the real applications?" More specifically, the key issue can be stated as "how can we enhance the detection rate with the false-rejected instances while maintaining the false positive rate to be low?" We call this problem as the "optimization on the enforced training set." To the best of our knowledge, there has been no report in literature on this problem for the case of boosted detector. Note that, for the boosted cascade detector, this problem is not easy to solve for several reasons. First, in boosting algorithms, selection of the base hypothesis and its weights are executed in sequential fashion based on some implicit conditions, so there is no explicit rule to modify the completely trained detector. Second, because of the huge amount of computational time for the training, retraining the cascade detector is impractical. Third, in a cascade structured detector, most of the target instances should not be rejected by any single layer in it. So, in case false-rejected target instances are enforced, a simple heuristic solution such as lowering the threshold of each layer will increase the false positive rate at each layer exponentially,

resulting overall great amount of computational burden in real applications.

To overcome these difficulties, the optimization algorithm for a boosted cascade detector on an enforced training set needs the following three conditions:

(1) an explicit optimization rule guaranteed by the mathematical background,
(2) less optimization time than the time for the retraining,
(3) low false positive rate while maintaining the expected detection rate for a given target training set.

In this paper, we propose a fast algorithm called DOOMRED (direct optimization of margin for rare event detection) that optimizes the base hypothesis weight set of each single layer detector in a boosted cascade detector, especially when the false-rejected target instances are enforced. Note that, in a boosting algorithm, the base hypothesis selection procedure from the large candidate base hypothesis set usually demands high-computational cost. This is the reason why we focus on the optimization of the base hypothesis weight set for the performance enhancement of a boosted cascade detector. In this respect, DOOMRED may be categorized as a kind of back-fitting which is a well-known optimization algorithm in the machine learning field [5].

2. BOOSTING ALGORITHM

Boosting is a well-known machine learning method that constructs a binary classification rule from certain training data set. The basic idea of a boosting algorithm is somewhat simple though clear such that an ensemble combination of multiple base hypotheses makes one strong hypothesis. The base hypothesis means a classification rule which has slightly better accuracy than random choice on a given training data set, which has error slightly less than 0.5. Meanwhile, the strong hypothesis indicates a classification rule that has high accuracy on the given training data set. Because constructing a highly accurate classification rule at one try is hard, a boosting algorithm constructs one accurate classification rule by combining multiple classification results from several base hypotheses. At this point, two important issues will be what base hypotheses to select from abundant candidate base hypothesis set, and how to combine them to make one accurate classification rule. In a learning procedure, both the training data set S and the candidate base hypothesis set H should be predefined. Then, for T iterations, corresponding base hypotheses $h \in H$ are selected sequentially from the candidate base hypothesis set H by updating the weight distribution of the training instances contained in S using an implicit cost function. The merit of the boosting algorithm is that the selection procedure of the base hypothesis can compensate the performance of the previously selected base hypotheses. If an instance is classified correctly by the base hypothesis selected in previous iteration, its weight decreases and vice versa. By updating the importance of each instance for every iteration, a base hypothesis that has good accuracy on instances not correctly classified by previously selected base hypothesis is selected. After T iterations, T base hypotheses are selected and the final classification rule is obtained by a linear combination (or ensemble combination) of T base hypotheses. The final outcome of the boosting algorithm is a binary classification rule $f(x)$ on the test instance x, which is labeled by $y \in \{-1, 1\}$ as shown in (1) where $\sum_{i=1}^{T} w_i = 1, w_i > 0, h_i(x) \in \{-1, 1\}$. We limit our work in the range of boosting algorithms which deal with hypotheses h having only binary outputs -1 and 1:

$$f(x) = \sum_{i=1}^{T} w_i h_i(x) = \begin{cases} \geq \theta_T, & x \in \text{class 1 (target)}, \\ < \theta_T, & x \in \text{class 2 (non-target)}. \end{cases} \tag{1}$$

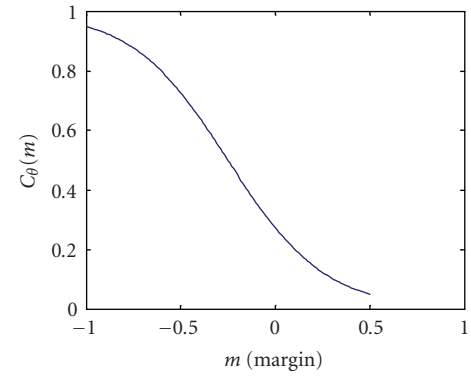
Note that each of the finally selected base hypothesis h_i corresponds to a basis of the feature space where instances are distributed, and the set of T base hypotheses' weight set is a gradient of a linear decision boundary. For this reason, the training procedure of a boosting algorithm can be interpreted as "data dimension reduction," that is selecting base hypotheses which makes the distribution of class 1 ($y = -1$) and class 2 ($y = 1$) instances in feature space separable with a linear decision boundary.

When $f(x)$ is used as a general binary classifier, the threshold $\theta_T = 0$. However, when $f(x)$ is used as a rare-event detector such as a frontal face detector, θ_T is usually set in the range $-1 < \theta_T \leq 0$. This is to guarantee the detection rate of $f(x)$ to be a specific goal value.

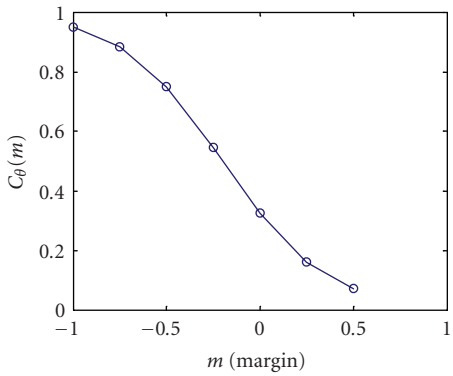
3. SINGLE LAYER DETECTOR OPTIMIZATION

3.1. Problem statement

Note that our problem is how to optimize the classification rule $f(x)$ when false classified instances in real application are added to the original training data set used in a forward training procedure. In detection problem, usually the number of nontarget object instances in real world is far larger than that of target object instances. So, the detection problem is often referred as "rare event detection" problem to indicate this situation. And, in general, a high detection rate results in a high false positive rate and vice versa. When constructing a detection rule, the way decreasing the false positive rate sacrificing detection is not used. This is because the detection rule with low detection rate is meaningless. As a result, the objective of constructing a detection rule may be defined as minimizing the false positive rate while fixing the detection rate to a specific goal value. In a cascade detector [1], usually, instances rejected by any subdetector in cascade are rejected forever for fast detection speed. Although the cascade detector is very appropriate for fast detection speed, this model is very hard to arrange when some false classified instances, especially false classified target instances, are enforced. To make a cascade detector to have a specific goal detection rate even when enforced target instances are added to the original training data set, every subdetector in cascade should be arranged not to lose enforced target-object instances. There are two heuristic solutions to this problem. First one is to simply retrain the whole cascade detector. The only remaining problem of it is that training cascade detector with



(a)

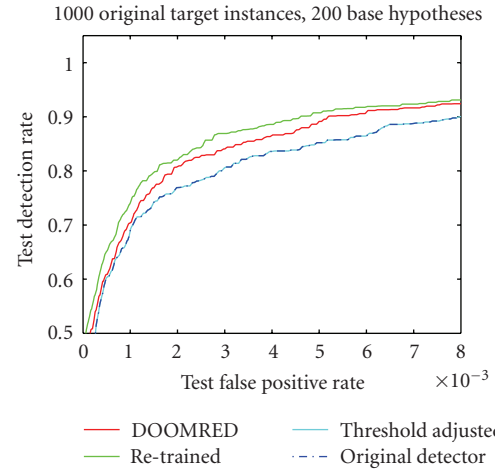


(b)

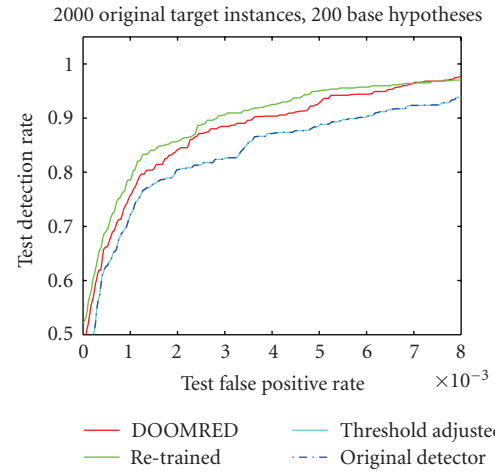
FIGURE 1: (a) Mean-shifted sigmoid function in (13) when $\theta = 0.5$. (b) Digitized version of (a) when the margin is segmented in the size 0.25.

a boosting algorithm when large training set is given requires substantially long training time. To train a boosted frontal face cascade detector, the size of training data set should be about several ten thousands because training set should include all instances that represents the face's large variance in appearance. For the frontal face detection problem, the training time for a cascade detector is minimally about several days using the fastest source codes such as OpenCV and maximally several weeks [1]. So, retraining cascade detector for every time when some instances are enforced is impractical. Second heuristic solution is to adjust the threshold θ_T of each subdetector to make each one satisfies the specific goal detection rate as in (1). One obvious drawback of this approach is that the solution also increases the false positive rate exponentially. This will result in decreasing the detection speed of the whole cascade detector in real applications.

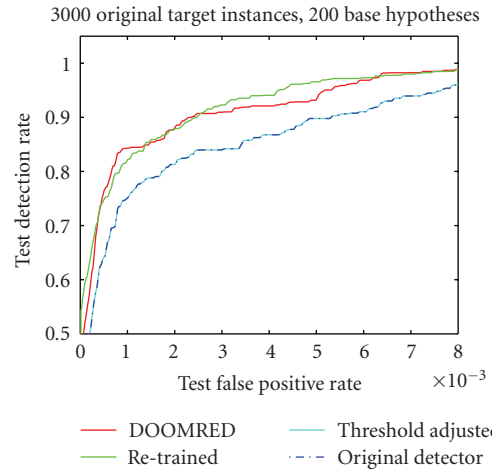
To overcome the shortcomings of the two heuristic solutions, in this work, we proposed a new optimization method for subdetectors in a boosted cascade detector that can minimize the false positive rate while maintaining the goal detection rate in reasonably small amount of



(a)



(b)



(c)

FIGURE 2: The ROC curves of the single layer detectors when the detector is initially trained with (a) 1000 (b) 2000 (c) 3000 target instances.

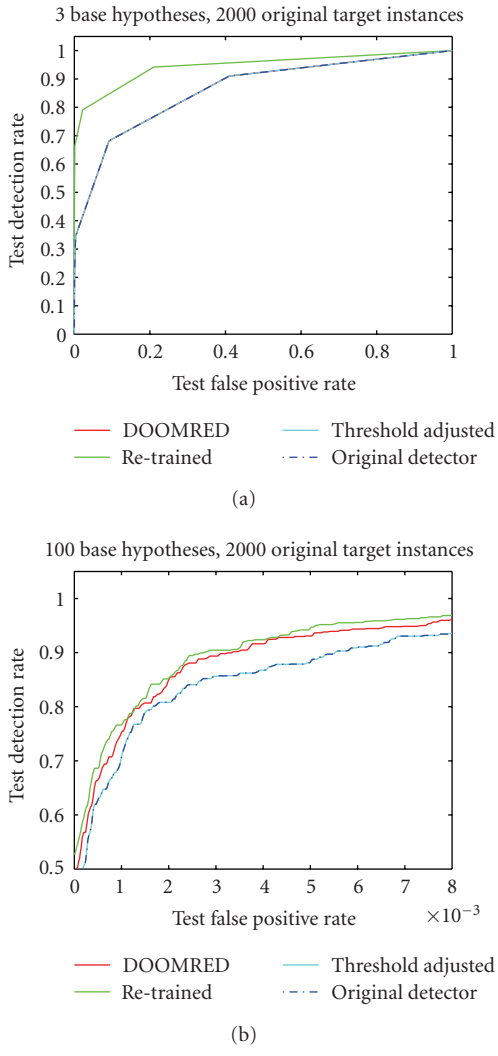


Figure 3: The ROC curves of the single layer detectors when the detector contains (a) 3 (b) 100 (c) 200 base hypotheses.

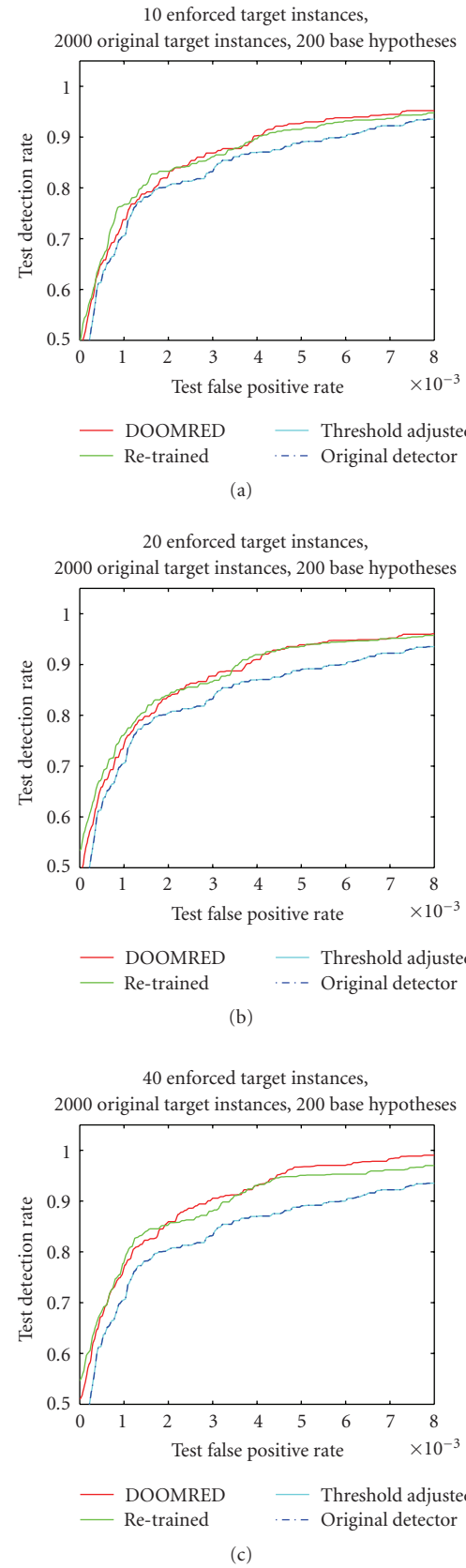


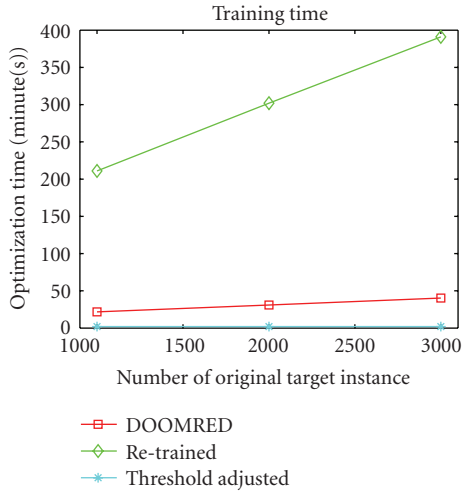
Figure 4: The ROC curves of the single layer detectors when the number of (a) 10 (b) 20 (c) 40 false rejected target instances are enforced.

optimization time. Our basic idea is to optimize the decision boundary of each subdetector only. The reason is that the most portion of training time of the boosting algorithm comes from base hypothesis selection procedure. Sections 3.2 and 3.3 describe a mathematically derived optimization rule and an optimization algorithm that we propose.

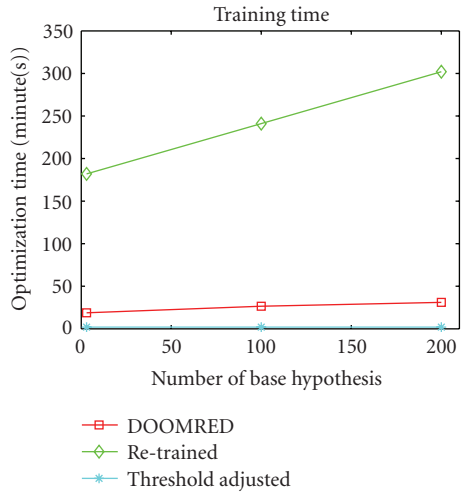
3.2. Optimization rule

In this section, an upper bound on test error of the classification rule $f(x)$ in (1) is derived in a more generalized form than the work in [6]. Based on the derived equation, the factors that affect the accuracy of the boosted detector may be extracted. Then, by adjusting the controllable factors, the boosted detector can be optimized on the enforced training set.

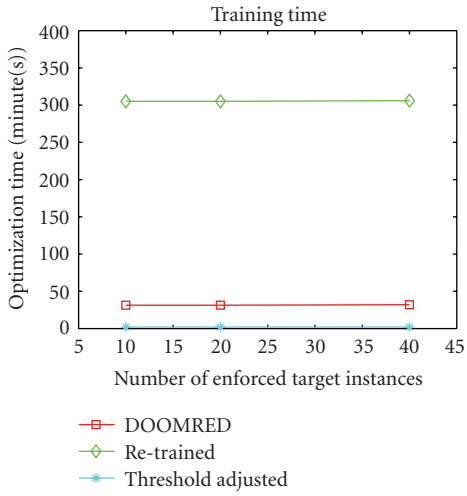
Since the AdaBoost algorithm was proposed [7], it has been shown from the subsequent experiments that the



(a)



(b)



(c)

Figure 5: The optimization time for the case in (a) Figure 2, (b) Figure 3, (c) Figure 4.

gap between the training error and test error decreases as the number of selected base hypotheses increases even after the training error reached to zero. These results show that the AdaBoost algorithm does not fit to the basic machine learning theory, Occam's razor, saying that the classification rule should be as simple as possible to minimize the gap between the training error and test error. To explain this phenomenon, the upper bound on error of voting methods such as the AdaBoost algorithm has been derived mathematically in [6].

However, the upper bound derived in [6] works only for the general binary classification problem, when $\theta_T = 0$ in (1). To make the upper bound on error of the boosting algorithm applicable even for the rare-event detection problem (when $-1 < \theta_T \leq 0$), we derive a more generalized error upper bound in Theorem 1. So, (2) can be used as an upper bound of the false positive rate in real applications when θ_T is tuned in the range $-1 < \theta_T \leq 0$ to get a specific goal detection rate. The proof is given after Theorem 1 following the similar procedure in [6].

Theorem 1. *Let P be a distribution over (x, y), $y \in \{-1, 1\}$, and let S be a set of k examples chosen independently at random according to P. Assume that the base hypothesis space H is finite, and let $\delta > 0$. Then with the probability of at least $1 - \delta$ over the random choice of the training set S, every function f made as a combination of $h \in H$ satisfies the following bound for all $-1 < \theta_T \leq 0$, $0 < \theta < 1$, and $0 < -\theta_T + \theta < 1$:*

$$Pr_P[yf(x) \leq -\theta_T] \leq Pr_S[yf(x) \leq -\theta_T + \theta] + O\left(\frac{1}{\sqrt{k}}\left(\frac{\log k \log |H|}{\theta^2} + \log\left(\frac{1}{\delta}\right)\right)^{1/2}\right). \tag{2}$$

Proof. For the sake of the proof, we define C_N to be the set of unweighted average over N elements from H:

$$C_N = \left\{ f : x \mapsto \frac{1}{N}\sum_{i=1}^{N} h_i(x) \mid h_i \in H \right\}. \tag{3}$$

We allow the same $h \in H$ to appear multiple times in the sum. This set will play the role of the approximating set in the proof. □

Any majority vote classifier $f \in C$ can be associated with a distribution over H as defined by the coefficients w_i. By choosing N elements of H independently at random according to this distribution, we can generate an element of C_N. Using such a construction, we map each $f \in C$ to a distribution Q over C_N. That is, a function $g \in C_N$ distributed according to Q is selected by choosing $h_1, \ldots, h_N$ independently at random according to the coefficients w_i and defining $g(x) = (1/N)\sum_{i=1}^{N} h_i(x)$.

Our goal is to upper bound the generalization error of $f \in C$. For any $g \in C_N$ and $\theta > 0$, we can separate this probability into two terms:

$$\begin{aligned}&\Pr_P[yf(x) \le -\theta_T]\\ &\le \Pr_P\left[yg(x) \le \frac{\theta}{2}\right] + \Pr_P\left[yg(x) > \frac{\theta}{2}, yf(x) \le -\theta_T\right].\end{aligned} \tag{4}$$

This holds because, in general, for two events A and B,

$$\Pr[A] = \Pr[B \cap A] + \Pr[\overline{B} \cap A] \le \Pr[B] + \Pr[\overline{B} \cap A]. \tag{5}$$

As (4) holds for any $g \in C_N$, we can take the expected value of the right-hand side with respect to the distribution Q and get

$$\begin{aligned}&\Pr_P[yf(x) \le -\theta_T]\\ &\quad\le \Pr_{P,g\sim Q}\left[yg(x) \le \frac{\theta}{2}\right]\\ &\qquad + \Pr_{P,g\sim Q}\left[yg(x) > \frac{\theta}{2}, yf(x) \le -\theta_T\right]\\ &\quad= E_{g\sim Q}\left[\Pr_P\left[yg(x) \le \frac{\theta}{2}\right]\right]\\ &\qquad + E_P\left[\Pr_{g\sim Q}\left[yg(x) > \frac{\theta}{2}, yf(x) \le -\theta_T\right]\right].\end{aligned} \tag{6}$$

We bound both terms in (6) separately, starting with the second term. Consider a fixed example (x, y) and take the probability inside the expectation with respect tothe random choice of g. It is clear that $f(x) = E_{g\sim Q}[g(x)]$. So, the probability inside the expectation is equal to the probability that the average over N random samples from a distribution over $\{-1, +1\}$ is larger than its expected value by more than $\theta/2$. The Chernoff bound yields

$$\Pr_{g\sim Q}\left[yg(x) > \frac{\theta}{2} \mid yf(x) \le -\theta_T\right] \le \exp\left(-\frac{N\theta^2}{8}\right). \tag{7}$$

To upper bound the first term in (7) we use the union bound. That is, the probability over the choice of S that there exists any $g \in C_N$ and $\theta > 0$ for which

$$\Pr_P\left[yg(x) \le \frac{\theta}{2}\right] > \Pr_S\left[yg(x) \le \frac{\theta}{2}\right] + \varepsilon_N \tag{8}$$

is at most $(N + 1)|C_N| \exp(-2m{\varepsilon_N}^2)$. The exponential term $\exp(-2m{\varepsilon_N}^2)$ comes from the Chernoff bound which holds for any single choice of g and θ. The term $(N + 1)|C_N|$ is an upper bound on the number of such choices where we have used the fact that, because of the form of functions in C_N, we need only to consider values of θ of the form $2i/N$ for $i = 0, \ldots, N$. Note that $|C_N| \le |H|^N$.

Thus, if we set $\varepsilon_N = \sqrt{(1/2m)\ln((N + 1)|H|^N/\delta_N)}$, and take expectation with respect to Q, we get with probability at least $1 - \delta_N$,

$$\Pr_{P,g\sim Q}\left[yg(x) \le \frac{\theta}{2}\right] \le \Pr_{S,g\sim Q}\left[yg(x) \le \frac{\theta}{2}\right] + \varepsilon_N \tag{9}$$

for every choice of θ and every distribution Q.

To finish the argument we relate the fraction of the training set on which $yg(x) \le \theta/2$ to the fraction on which $yf(x) \le -\theta_T + \theta$, which is the quantity that we measure. Using (5) again, we have that

$$\begin{aligned}&\Pr_{S,g\sim Q}\left[yg(x) \le \frac{\theta}{2}\right]\\ &\quad\le \Pr_{S,g\sim Q}[yf(x) \le -\theta_T + \theta]\\ &\qquad + \Pr_{S,g\sim Q}\left[yg(x) \le \frac{\theta}{2}, yf(x) > -\theta_T + \theta\right]\\ &\quad= \Pr_S[yf(x) \le -\theta_T + \theta]\\ &\qquad + E_S\left[\Pr_{g\sim Q}\left[yg(x) \le \frac{\theta}{2}, yf(x) > -\theta_T + \theta\right]\right].\end{aligned} \tag{10}$$

To bound the expression inside the expectation we use the Chernoff bound as we did for (7) and get

$$\Pr_{g\sim Q}\left[yg(x) \le \frac{\theta}{2} \mid yf(x) > -\theta_T + \theta\right] \le \exp\left(-\frac{N\theta^2}{8}\right). \tag{11}$$

Let $\delta_N = \delta/(N(N + 1))$ so that the probability of failure for any N will be at most $\sum_{N\ge1}\delta_N = \delta$. Then combining (6), (7), (9), (10), and (11), we get that, with probability at least $1 - \delta$, for every $\theta > 0$ and every $N \ge 1$,

$$\begin{aligned}\Pr_P[yf(x) \le -\theta_T] \le \Pr_S[yf(x) \le -\theta_T + \theta] + 2\exp\left(-\frac{N\theta^2}{8}\right)\\ + \sqrt{\frac{1}{2m}\ln\left(\frac{N(N+1)^2|H|^N}{\delta}\right)}.\end{aligned} \tag{12}$$

Finally, the statement of the theorem follows by setting $\lceil N = (4/\theta^2)\ln(m/\ln|H|)\rceil$.

$\Pr_P[A]$ and $\Pr_S[A]$ denote the probabilities of the event A when (x, y) is chosen according to P and uniformly at random from the set S, respectively. Above Theorem 1 verifies that factors that affect the upper bound on test error does not vary when $\theta_T = 0$ or $-1 < \theta_T \le 0$. Now, the four variables that can affect the upper bound on test error of $f(x)$ can be summarized as follows:

$|H|$: the size of the candidate base hypothesis set,
k: the size of the training data set,
$\Pr_S[yf(x) \le -\theta_T + \theta]$: the portion of nontarget training instances whose margin is under θ,
θ: the goal marginal value.

Let us examine how those four factors affect the upper bound on error of the initially trained $f(x)$ when some false-classified training instances are enforced. First, $|H|$ is unchanged. Second, k is increased resulting in the lower upper bound on test error. Finally, $\Pr_S[yf(x) \le -\theta_T + \theta]$ and θ remain as two controllable factors. Thus, an optimization rule can be derived as a conclusion: to minimize the test false positive rate when θ_T is adjusted to set the detection rate to a specific goal value, maximize the number of nontarget training instances whose margin $yf(x) + \theta_T$ are larger than the specific θ while maximizing θ, too. This suggests a rule for the optimization of the base hypothesis weight set of the single layer detectors in a boosted cascade detector.

3.3. DOOMRED

In this section, we propose a simple and fast algorithm DOOMRED that optimizes the base hypothesis weight set $W = \{w_1, \ldots, w_T\}$ of $f(x)$ in such a way to maximize the number of nontarget training instances whose margins are above a specific θ value. Algorithm 1 shows the pseudocode of DOOMRED. In [8], an algorithm named DOOM is introduced. DOOM optimizes the base hypothesis weight set $W = \{w_1, \ldots, w_T\}$ to minimize the cost function of the margins of the training instances. This optimization process results in the minimization of the classification error. However, DOOM cannot be directly applied to the detection problem. The reason is that basically DOOM is a two-class classification algorithm, and moreover it deals with the entire training instances of both classes in its optimization process. Since there are absolutely large amount of nontarget instances than that of target instances in rare-event detection problem, DOOM might show worse performance, especially low detection rate, when it is used for a rare-event detector. To solve this problem, we design the DOOMRED algorithm in which the target and the nontarget instances are dealt with separately in the optimization process.

DOOMRED is designed by adopting a simple steepest-gradient descent method. It is to guarantee the simplicity of the algorithm to minimize the computational burden for the optimization. Although DOOMRED only modifies the weights of the base hypotheses, there are great amount of training instances to deal with, which might result in a great amount of optimization time.

Before the optimization procedure, we need to define a marginal cost function to be minimized, that should be a monotonically decreasing function defined in the range from -1 to a specific θ value. The mean shifted sigmoid function in (13) and Figure 1 is an example. It represents the importance of each training instance during the optimization procedure:

$$C_\theta(m) = \frac{1}{1 + \exp\left(a \times (m - (\theta - 1)/2)\right)}, \quad \text{where } a > 0. \tag{13}$$

In DOOMRED, first, among the target and nontarget instances contained in the target training set S_P and the nontarget training set S_N, the instances whose margins are under the predefined θ_P and θ_N are classified into the sets E_P and E_N, respectively. The training instances contained in E_P and E_N only affect the modification of the base hypothesis weight set W. Then each base hypothesis weight $w_i \in W$ is modified to increase the margin of instances both in E_P and E_N. w_i is increased if it decreases the summation of the marginal cost function $C_\theta(m)$ $(\text{cost}(W))$ of the instances in E_P and E_N, and vice versa. The amount of modification of w_i is determined by the characteristic of the marginal cost function $C_\theta(m)$. For an example, when (13) is used as a $C_\theta(m)$, instances whose margins are around $(\theta - 1)/2$ largely affect the amount of modification of w_i. These two simple processes are iterated until cost (W) or variance of W converges. Note that DOOMRED may decrease the margins of the training instances which are not contained in the sets E_P and E_N. However, we also note from (2) that, after a certain value of θ is determined, the accuracy of $f(x)$ is not affected by the instances whose margins are above θ. It is because once θ is determined, the only issue we should consider is the portion of training instances whose margins exceed θ. After each DOOMRED execution, the threshold value is adjusted to make the training detection rate to be the specific goal value.

Table 1: Parameter settings for the single layer optimization method in Algorithm 2 used in our experiments.

Parameter	Value
$C_\theta(m)$	Figure 1(b)
N_W	300
Variance of θ_P	0.0 to 0.5 (step 0.1)
Variance of θ_N	0.4 to 0.8 (step 0.1)
Prec	0.01

3.4. Single layer optimization method

Although we have derived a simple and clear optimization rule for a boosted single layer detector from Theorem 1, one problem still remains that (2) does not provide the exact values of the key parameters to minimize the test error for the nontarget set. Since our objective is to find a globally optimal solution, DOOMRED is executed on the various randomly selected initial values of the base hypothesis weight set W_R, θ_P, and θ_N. Among these various trials, W_S and θ_S which have the least false positive rate on the validation nontarget set S_{NV} are selected as the final output when the detection rate is fixed on the training target set S_P. The final output of the optimization is a boosted detector $f(x)$ that is expressed with the original H, $W = W_S$, and $\theta_T = \theta_S$. The pseudocode of the single layer detector optimization is given in Algorithm 2.

4. CASCADE DETECTOR OPTIMIZATION

For the optimization of a boosted cascade detector, the false-classified instances occurred in the real application are enforced to the first layer of the cascade detector. Then the optimization method for a single layer detector in Algorithm 2 is applied to each layer. In order not to degrade the efficiency of the cascade detector, the target and nontarget instances which are not rejected by any prelayer are used for the optimization of the postlayers.

5. EXPERIMENTAL RESULTS

5.1. Experimental environments

We tested the proposed algorithm to the frontal face detection problem. A face database used for our experiments contains 7143 24×24 sized face instances. Nonface instances are cropped from a 2179 natural scene images collected from the world wide web. Then both target and nontarget instances are divided into three groups. The first group is used as an initial training set, and the false-classified

```
DOOMRED (H, W, S_P, S_N, θ_P, θ_N, prec)
    exe = true
    while (exe)
        E_P = [(x, y) | (x, y) ∈ S_P, yf(x) < θ_P], E_N = [(x, y) | (x, y) ∈ S_N, yf(x) < θ_N]
        g = −∇_W cost (W)
        if (W + g has any negative valued element)
            scale g that no element of W + g has negative value
        if (weight Sum (g) ≥ prec)
            W_B = normalize (W + g)
            if (cost (W_B) < cost (W))        W = W_B
            else exe = false
        else exe = false
    return W

Notations
    H := {h_i | i = 1, ..., T}, number of T base hypotheses set
    W := {w_i | i = 1, ..., T, 0 < w_i < 1, Σ_{i=1}^{T} w_i = 1}, base hypothesis weight set
    S_P := {(x_i, y_i) | x_i = target training instance, y_i = 1}
    S_N := {(x_i, y_i) | x_i = non-target training instance, y_i = −1}
    cost (W) = Σ_{(x_i,y_i)∈E_P} C_{θ_P}(y_i f(x_i)) + Σ_{(x_i,y_i)∈E_N} C_{θ_N}(y_i f(x_i))
    weight Sum (W) = Σ_{i=1}^{T} w_i, w_i ∈ W
    normalize (W) = scale W that Σ_{i=1}^{T} w_i = 1
```

Algorithm 1: The pseudocode of DOOMRED.

instances among the second group by the completely trained detector are used as the enforced training data. The last group is used to measure the test errors.

Table 1 shows the parameter settings for the optimization method used in the single layer detector in Algorithm 2. First, the digitized sigmoid function shown in Figure 1(b) is defined as $C_\theta(m)$. Since the exponential function requires large computational burden, the region from -1 to specific θ is divided into 100 segments and the gradient of each segment is precomputed. Second, N_W and the variation ranges of θ_N and θ_P are set constant for all layers. These fixed values are determined by our tuning process. It might seem to be unreasonable to fix N_W to be independent of T, which is the number of the base hypotheses of $f(x)$. However, for the boosted cascade detectors in most of real applications, no more than 200 base hypotheses are used to construct each single layer detector in the cascade detector. The number 300 for the random base hypothesis weight set is enough to make the performance of DOOMRED stable when the number of base hypothesis is under 200.

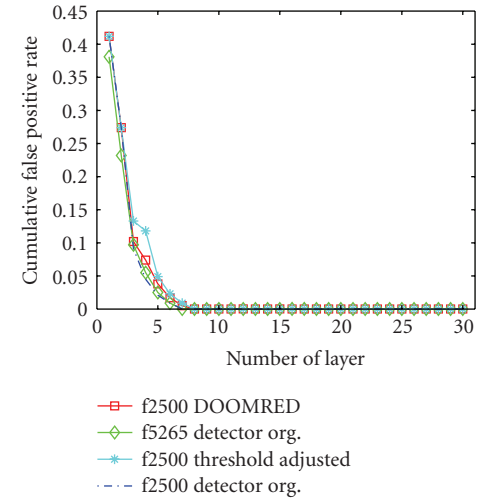
Experiments are performed on various boosted frontal face detectors trained using the Adaboost algorithm with different conditions of the size of initial training set, base-hypotheses contained, and the enforced target training set. Note that since DOOMRED only modifies the weights of the base hypotheses, the performance of DOOMRED depends on the quality of the initial feature space constructed in the learning procedure.

The performance of DOOMRED is evaluated based on two criteria; (1) ROC curve, (2) optimization (or training) time. The first factor is critically related to the accuracy and the detection time in real applications. The second factor is related to the training or optimization cost of a detector. This factor is particulary important since a boosting algorithm generally requires a large amount of training time. The performance of DOOMRED is compared to those of other heuristic solutions such as the adjusting threshold θ_T of single layer detector and retraining.

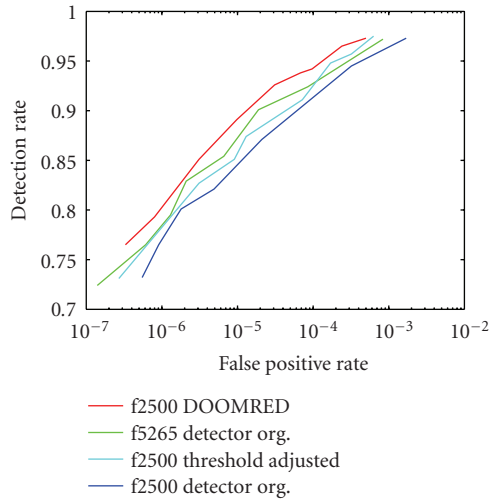
5.2. Experiments on single layer detectors

Figures 2, 3, and 4 show the ROC curves of various single layer detectors. In Figure 2, each original (or initial) detector was trained to have 200 base hypotheses. The face instances as many as 1000, 2000, 3000, and two times of each for the nonface instances were given as an initial training set. Then, false-rejected face instances which were not contained in the initial training set were enforced as many as 2000 false-classified nonface instances were enforced for the threshold adjusting and DOOMRED solutions, while 500 false-classified nonface instances were enforced for the retraining since the Adaboost algorithm occasionally failed to finish the learning procedure when 2000 nonface instances were enforced. In Figure 2, we can see that the amount of improvement in the accuracy by DOOMRED increased as the number of the initial training instance increased. Because DOOMRED deals with the weight set of base hypotheses only, the performance of DOOMRED seems to be affected by the quality of the base hypotheses selected during the initial training process.

In Figure 3, each initial detector was trained to have the number of base hypotheses of 3, 100, and 200. The sizes of the initial face and nonface training sets were 2000 and 4000, respectively. The sizes of the enforced face and nonface sets were same as in Figure 2. Except for the case when the number of the base hypotheses was



(a)



(b)

Figure 6: (a) The false positive rates of the boosted cascade frontal face detectors before and after the optimization. (b) ROC curves of the boosted cascade frontal face detectors before and after the optimization estimated on CMU+MIT frontal face test images composed of 130 images containing 507 faces.

3, DOOMRED demonstrated stable and also remarkable performances compared with those of the other approaches. In the case when the number of the base hypotheses was 3, DOOMRED failed to increase the accuracy. This was because there were large amount of the face instances which were determined as -1 (nonface) by all the 3 base hypotheses. There was no chance for these faces to be determined as faces by only adjusting the weights of base hypotheses using DOOMRED.

In Figure 4, one initial detector was trained to have 200 base hypotheses with 2000 face and 4000 nonface instances. Then 10, 20, and 40 false-rejected face instances which were not contained in the initial training set were enforced. The size of the enforced nonface set were same as in Figure 2.

Table 2: The average number of the base hypotheses evaluated per each nonface instance and the optimization (or training) time estimated on Pentium 4 2.8 GHz PC.

Detector	Average num. of base hyp.	Optimization Time(min)
F2500 org. detector	13.9	8291
F2500 doomred	17.6	704
F2500 thres adj.	20.3	4
F5265 org. detector	14.2	17431

It can be observed that DOOMRED increased the accuracy of the detectors when more enforced training instances were given. DOOMRED still showed stable and remarkable performance compared with those of the heuristic solutions. An interesting observation in our experiments was that the retraining sometimes failed to select 200 base hypotheses whose errors on the training set were under 0.5. A conclusion we can make on the single layer detector experiments is that DOOMRED exhibits a more stable and better performance than the other two naive approaches. The only exception was when the detector was initially trained with excessively small number of base hypotheses.

Figure 5 shows the optimization (training) time for the tests in Figures 2, 3, and 4. DOOMRED required only less than 11.3% of the computation time for the retraining method, while showing similar or better test false positive rates as shown in Figures 2, 3, and 4. Although the threshold adjusting method was fast by taking only a few minutes for any case, the performance was not satisfactory.

5.3. Experiments on cascade detectors

For this experiment, two boosted cascade frontal face detectors were initially constructed using AdaBoost algorithm. One was trained with an insufficient number of training data including 2500 face and 5000 nonface instances, and it was composed of 30 layers (2500-face detector). The second cascade detector was trained with an abundant number of training data including 5265 face and 10530 nonface instances, and was composed of 30 layers (5265-face detector). Each cascade detector was constructed sequentially with one 3-base hypotheses, one 5-base hypotheses, three 20-base hypotheses, and two 50-base hypotheses layers. The number of base hypotheses of all the postlayers was set to 200. Then 431 false-rejected face instances which were not contained in the initial training set were enforced to the first layer of the 2500-face detector. Due to occasional failures in the learning procedure of the retraining solution as mentioned in Section 5.2, tests for the performance of the retraining method were substituted by that of the 5265-face detector. Note that the 5265-face detector may be considered as a detector retrained with 2500 initial faces and 2765 enforced faces.

In Figure 6(a), the 2500-face detector shows the steepest decrement in the test false positive rate while showing the worst accuracy as can be seen in Figure 6(b). Meanwhile, the 5265-face detector shows similarly a low false positive rate

```
OptimizeSingleLayer (H, S_PI, S_PE, S_NI, S_NE, D_G, prec)

    S_P := sum S_PI and S_PE, S_N := sum S_NI and S_NE
    divide S_N into S_NT and S_NV
    for (number of N_W)
        W_R = randomly generated base hypothesis weight set of H
        for (various θ_P and θ_N)
            W_O = DOOMRED (H, W_R, S_P, S_NT, θ_P, θ_N, prec)
            adjust θ_T to get D_G on S_P with H, W_O
            if (least false positive rate is made on S_NV with H, W_O, θ_T)
                W_S = W_O, θ_S = θ_T
    return W_S, θ_S

Notations
    H: base hypothesis set
    S_PI, S_NI: initial target, non-target training set
    S_PE, S_NE: enforced target, non-target training set
    D_G: goal detection rate of single layer detector
```

Algorithm 2: The pseudocode for the optimization of the boosted single layer detector on the enforced training set.

at each single layer detector while showing good accuracy. As the 431 false-rejected faces were enforced to the 2500-face detector, the threshold adjusting method demonstrated good accuracy in the ROC curve even compared to that of the 5265-face detector as shown in Figure 6(b). We think that this is because the informative training instances (enforced training instances) effectively compensated the distribution of the initial training set. However, one problem of this heuristic method is that the detector becomes slower in real applications as the number of the enforced instance increases. In Table 2, the average number of the base hypotheses calculated per nonface instance is shown, which is critically related to the detection time in real applications. When the threshold θ_T of each layer was simply adjusted to acquire 99.5% of the training detection rate, the detection time increased by 46.0% and 42.9% compared to those of the 2500-face detector and 5265-face detectors. However, if DOOMRED was applied, these computational cost increments decreased to 26.6% and 23.9% while showing the best accuracy among those of three other cases in Figure 6(b). Note also that the optimization time of DOOMRED on the 2500-face detector was 704 minutes as shown in Table 2. This is barely 8.5% and 4.0% of the training time required in the 2500-face detector and 5265-face detectors, respectively. Therefore, we can conclude that the proposed DOOMRED is a reasonable solution for the optimization of the boosted cascade detector on the enforced training set considering its excellent performance to enhance the detection speed and accuracy in reasonable optimization time.

6. CONCLUSION

In this paper, we proposed DOOMRED, an algorithm to modify the base hypothesis weight set initially constructed by a boosting algorithm. It can be applied to the boosted single layer or cascade detector when the false-classified training set is enforced. Experimental results demonstrated that DOOMRED excellently enhanced the performance of the boosted single layer or cascade detectors compared to those of other heuristic approaches while requiring reasonable optimization time. DOOMRED, however, showed weak performance when the number of the base hypotheses is small. To overcome this limitation, we are planning to develop an efficient algorithm that can substitute the inappropriate base hypotheses with the optimal ones.

ACKNOWLEDGMENT

This work was supported in part by the ITRC program by Ministry of Information and Communication and in part by Defense Acquisition Program Administration and Agency for Defense Development, South Korea, through the Image Information Research Center under the Contract UD070007AD.

REFERENCES

[1] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '01)*, vol. 1, pp. 511–518, Kauai, Hawaii, USA, December 2001.

[2] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor, "Linear programming boosting via column generation," *Machine Learning*, vol. 46, no. 1–3, pp. 225–254, 2002.

[3] S. Z. Li and Z. Zhang, "FloatBoost learning and statistical face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1112–1123, 2004.

[4] P. Viola and M. Jones, "Fast and robust classification using asymmetric adaboost and a detector cascade," in *Advances in Neural Information Processing Systems 14*, pp. 1311–1318, MIT Press, Cambridge, Mass, USA, 2002.

[5] T. J. Hastie and R. J. Tibshirani, *Generalized Additive Models*, Chapman & Hall/CRC, London, UK, 1990.
[6] R. E. Schapire, Y. Freund, P. L. Bartlett, and W. S. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.
[7] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.
[8] L. Mason, P. L. Bartlett, and J. Baxter, "Improved generalization through explicit optimization of margins," *Machine Learning*, vol. 38, no. 3, pp. 243–255, 2000.