Face Detection using SURF Cascade

Jianguo Li, Tao Wang, Yimin Zhang
Intel Labs China
{jianguo.li, tao.wang, yimin.zhang}@intel.com

Abstract

We present a novel boosting cascade based face detection framework using SURF features. The framework is derived from the well-known Viola-Jones (VJ) framework but distinguished by two key contributions. First, the proposed framework deals with only several hundreds of multi-dimensional local SURF patches instead of hundreds of thousands of single dimensional haar features in the VJ framework. Second, it takes AUC as a single criterion for the convergence test of each cascade stage rather than the two conflicting criteria (false-positive-rate and detection-rate) in the VJ framework. These modifications yield much faster training convergence and much fewer stages in the final cascade. We made experiments on training face detector from large scale database. Results shows that the proposed method is able to train face detectors within one hour through scanning billions of negative samples on current personal computers. Furthermore, the built detector is comparable to the state-of-the-art algorithm not only on the accuracy but also on the processing speed.

1. Introduction

Face detection is one hot research topic in computer vision due to wide applications. Great advances have been made in the passed decade, especially since the milestone work by Viola and Jones [32]. To realize good generalization performance on the data explosive era, more training data are required and used in the learning procedure. This is not a problem nowadays, since it is much cheaper to collect many labeled samples from Internet with small incremental human labeling effort like Mechanical Turk [27]. Consequently, large scale training data make the learning a critical bottleneck. This is especially true for training face detectors [33, 18, 36]. As is known, the training is usually required to reach very low false positive rate per scan-window (FPPW) such as 1e-6 [32, 33], which means that hundreds of millions or even billions of negative samples should be processed during the training procedure. Therefore, training face detector is a very time-consuming task. In early works, it may take months to train a high-quality detector due to limited computing resources. Even with the great increase of computing power today, existing learning frameworks are still inefficient to handle such a large-scale training problem. It is still required several days or even weeks to obtain a high-quality detector [37, 24, 30, 36]. As there are usually required some fine-tuning of parameters based on training experiments [35, 2], the long training time yields very painful experiences for researches in face detection field.

Besides the large-scale dataset, there are two major factors that impact the efficiency of training. First factor is the size of feature search space. There are three typical ways to learn base classifier on feature spaces.

- The VJ framework [32] learns base classifier on each single haar feature separately. In a typical 20×20 detection template, there are more than hundreds of thousands of haar features. It is a critical problem of feature selection on such a large space.
- The HoG-SVM framework [4] directly learns SVM classifier from high dimensional concatenated HoG features. The high-dimensional representation not only yields time-consuming training procedure, but also leads to slow detection speed.
- The cascade-HoG framework [37] learns base classifier over local HoG blocks. The feature space has been reduced from hundreds of thousands to several thousands. However, the training is still required days-level. Parts are due to slow feature extraction speed, and parts are due to the training convergence speed.

Second factor is the training convergence speed. Most existing cascade detection frameworks are trained based on two conflicting criteria (false-positive-rate and detection-rate) for the detection-error tradeoff. Although some researches introduced intermediate tuning of cascade threshold with some optimization methods [35, 2, 26], they attempted to remove manual threshold tuning step, but did not touch the training convergence problem. As a result, the final cascade usually has very long stages.

In this paper, we proposed a new cascade learning framework for face detection, named SURF cascade. First, we

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00

represent target detection window by SURF features [1] in local patches within the window. In each detection window, the number of local SURF patches is limited to several hundreds. We adopt logistic regression as base learner on each local patch. In this way, the feature pool size is reduced greatly which makes feature selection much easier. Moreover, the feature extraction of SURF is much faster than existing local features like HoG[4]. Second, we adopt AUC (Area under ROC curve) as a single criterion for convergence test during cascade training instead of the two conflicting criteria (detection-rate and false-positive-rate) on ROC curve. The training of SURF cascade converges much faster and generates cascade with much fewer stages. Our experiments showed that the proposed framework can build highly accurate detector from billions of samples within one hour on current personal computers. In summary, this paper presents three major contributions:

(1) We introduced SURF features for fast face detection.

(2) We proposed the single AUC based criterion for cascade optimization, which makes the cascade converge faster and yield much fewer cascade stages.

(3) We showed a system that can train cascade face detector from billions of samples within one hour on current personal computers.

In the rest of the paper, we will first revisit related works on face detection in Section 2, and present the details of the proposed framework in Section 3. Experiments are shown in Section 4. Conclusions are drawn in Section 5.

2. Related Works

The boosting cascade framework by Viola-Jones is a milestone work in object detection [32, 33]. It inspired many following works. It is necessary to revisit the VJ framework before diving into more details. Basically, there are three key ideas that make it able to build real-time detectors: the integral images for efficient haar feature computing, the boosting learning of weak classifier, and the attentional cascade structure for fast negative rejection. There are several limitations regarding to the VJ framework.

First, the feature pool size of haar-like features is too large, which is usually in hundreds of thousands level for a typical 20×20 detection template. The learning of boosting decision tree requires testing all the features for all the training data in each iteration. This is extremely large computing requirement. Some heuristic feature selection or filtering strategies have been proposed to speedup the training by several hundreds of folds [19, 3, 23]. However, these kind of techniques are not only still unsatisfied due to such a large feature pool size, but also may lead to possible performance loss due to un-optimal search.

Second, the feature representation capacity of haar feature is very limited. It can not well handle viewpoint, pose and illumination variations. On the one hand, some researches extended haar-features to propose rotated haar-like features [16], joint haar-features [22], sparse features [11], polygon features [24], etc. These features can improve the detection accuracy but make feature space even larger, and thus the feature-selection problem becomes even critical. On the other hand, some complex features are proposed and used in object detection field, which includes histogram of oriented gradient (HoG) features [4, 37], region-covariance features [28], etc. They are more complex than haar-features, and thus are much slower in computing. This paper introduces SURF features [1] to describe local patches within detection window. It is close related to HoG representation in the cascade-HoG framework [37]. However, SURF is much faster in computing than that of HoG.

Third, the attentional cascade is trained based on two conflicting criteria on ROC curve (false-positive rate f_i and detection-rate d_i) for the detection-error tradeoff. The overall false-positive-rate (FPR) of the T-stage cascade is thus $F = \prod_{i=1}^{T} f_i$, while the overall detection-rate is $D = \prod_{i=1}^{T} d_i$. Due to the detection error trade-off on the ROC curve, the VJ framework suggests setting the maximum false alarm rate such as f_i=0.5 and the minimum detection-rate such as d_i=0.995 during the training procedure. If we want to reach overall false positive-rate 1e-6, it requires at least 20 stages ($0.5^{20} \approx$1e-6) by the given global setting. This is obviously inefficient since on some stages, it may reach the goal without convergence. It is better that we have adaptive value for f_i among different stages such that we could easily reach overall training goal. Some works designed some automatic scheme to tune the intermediate threshold of each stage [35, 2, 26, 3]. These may alleviate the painful manual tuning effort, but have nothing to do with the convergence speed. Inspired by the work [17], we introduced area under ROC curve (AUC) [6] as a single criterion for cascade convergence testing. This AUC based training procedure realizes adaptive false-positive-rate for each stage, and thus the cascade converges much faster, and the final cascade has much fewer number of stages.

The proposed approach has some relationship with part-based detection methods[10, 8]. Early part-based method trained parts detectors separately and integrated them together [20] for detection. Later works trained mixtures of deformable part models even under the cascade framework [8, 9]. The proposed method can be viewed as a simple case of the part-based methods when viewing base classifier on each local patch as part-based model. However, the proposed approach just combines local parts in a discriminative way for each stage without any pictorial structures among parts. Although it may not have the same power to handle deformable objects, it is still valuable in object detection field due to its simplicity and efficiency.

3. SURF Cascade

The proposed algorithm contains three ingredients: SURF feature to represent local patches; logistic regression based weak classifier; and the AUC-based boosting cascade learning algorithm. We describe them below separately.

3.1. SURF Features

SURF (Speeded Up Robust Features) is a detector and descriptor of local scale- and rotation- invariant image features [1]. It has been successed applied in applications like object recognition, image matching, structure-from-motion, etc. This paper only used SURF descriptor for face detection, and did not need the keypoint detection part.

The original SURF descriptor is defined in the gradient space. The gradient may have different granular. However, currently we simply define d_x be the horizontal gradient image obtained with the filter kernel $[-1, 0, 1]$ and d_y be the vertical gradient image by the filter kernel $[-1, 0, 1]^T$. We adopted the extended form of SURF descriptor, i.e., the sums of d_x and $|d_x|$ for $d_y < 0$ and $d_y \geq 0$, and the sums of d_y and $|d_y|$ for $d_x < 0$ and $d_x \geq 0$ for given local patches.

We densely sampled SURF descriptor over the target detection window. The descriptor is thus not rotation invariant. In practice, given a detection window, we consider possible local patches within the detection window similar to the scheme used in cascade HoG [37]. For instance, given a 40×40 detection template, we choose patch size range from 12×12 to 40×40, and allow the patch width and height ratio like 1:1, 1:2, 2:1, 2:3, 3:2. We slide the same size patch over the detection window with fixed step 4 pixels [1]. In total, 396 patches are generated from the 40×40 detection template. Each local patch is represented by 2×2 cells of extended SURF features. Concatenating features in 2×2 cell together, this yields a feature vector of 8×2×2=32 dimensions. Finally, this descriptor is turned to unit vector with L_2 normalization, which makes the descriptor scale-invariant to the size of detection window.

For fast feature extraction, we exploit the integral image techniques [32]. Since there are 8-channels of the extended SURF features in each cell, if building one integral image for each channel separately, it needs 4×8 = 32 image access operations to obtain SURF feature from one cell. Here, we adopt the structure-of-array trick which packs all 8 channels together with a structure array, and thus only need 4 access operations from the structure array to obtain SURF features in one cell. This trick greatly improves the feature extraction speed.

[1] As the SURF feature for each local patch is 32 dimensional, less than 4-pixel step does not provide sufficient differences for the SURF features of two neighbor and same-size local patches.

3.2. Weak Classifier

With SURF feature representation, we built weak classifier over each local patch, and selected optimal classifier in each boosting iteration from all possible local patches.

In the VJ framework, the base classifier is decision tree. In some following works, people introduced other linear classifiers such as Fisher linear discriminant [14], linear asymmetric classifier [25, 3] and linear SVM [4]. In this paper, we choose logistic regression as base learner. This is due to two reasons:

- Logistic regression can directly give probability output, which further makes the cascade easily produce probability at any level. This is very helpful for AUC-based cascade training.
- Logistic regression is simple linear model, which can be trained and evaluated very efficiently.

Given extended SURF features $\mathbf{x}$ over a local patch, the logistic regression [21] defines a probability model

$$P(y = \pm 1|\mathbf{x}, \mathbf{w}) = \frac{1}{1 + \exp(-y\mathbf{w}^T\mathbf{x})}, \quad (1)$$

where y is the label of the patch belongs to (positive or negative), $\mathbf{w}$ are weight parameters of the model.

Given training samples $\{\mathbf{x}_i, y_i\}_{i=1}^N$, the parameters can be found via minimizing the following objective

$$l(\mathbf{w}) = \sum_{i=1}^{N} \log(1 + \exp(-y\mathbf{w}^T\mathbf{x}) + \lambda\mathbf{w}^T\mathbf{w},$$

while λ is a parameter of the regularization term.

This optimization can be efficiently resolved by technology like iterative scaling [21].

3.3. AUC based Cascade Training

We trained the boosting cascade on local-patch features from large scale data with improvement over the original VJ cascade [32] and the cascade-HoG framework [37]. Suppose there are N training samples, and K possible local patches represented by d-dimensional (=32) extended SURF feature $\mathbf{x}$, each stage is a Boosting learning procedure with logistic regression as base learner. A strong classifier is a linear combination of T base learner h_t

$$H^T(x) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x}, \mathbf{w}), \quad (2)$$

where α_t is a weight of the base learner h_t in the range of [0,1] and $\sum_{t=1}^{T} \alpha_t = 1$. As each base learner gives probability output, the linear combination also produces probability output.

The strong classifier is learned in the following way. In the k-th boosting round, we built K logit models

$\{h_k(\mathbf{x})\}_{k=1}^{K}$ for each local patch in parallel from the boosting sampled sub- dataset. We tested each model $h_k(\mathbf{x})$ with combination of previous t-1 rounds in a brute force way. That is to say, we tested $H^{t-1}(\mathbf{x}) + \alpha_k h_k(\mathbf{x})$ on all the N training samples. Each tested model will produce an AUC score $J(H^{t-1}(\mathbf{x}) + \alpha_k h_k(\mathbf{x}))$. We choose the one as current model which can produce the highest AUC score, i.e.,

$$H^t(\mathbf{x}) = \arg\max_{k=1:K} J(H^{t-1}(\mathbf{x}) + \alpha_k h_k(\mathbf{x})). \quad (3)$$

The boosting procedure is converged when the AUC score is converged or the designed number of iteration is reached.

The weak classifier is trained on balanced positive and negative samples. To avoid overfitting of each weak classifier, we restrict the number of samples used in training as in [35]. In practice, we found that it is generally good sampling $20\times d$ number of positive/negative samples. Table 1 summarizes the detail algorithm of one boosting stage.

Table 1. AUC-Boosting of local SURF patches in one stage

1. Given training set: $\{(\mathbf{x}_i^k, y_i)\}_{i=1}^{N}$, $k = 1:K$, where N is the number of samples and K is the total number of local patches. $\mathbf{x}^k \in R^d$ is d-dimensional SURF feature representation of the k-th local patch.
2. Initialize weight for positive and negative respectively.
   (1) $w_{1,i}^{+} = \frac{1}{N_+}$ for those y_i=1;
   (2) $w_{1,i}^{-} = \frac{1}{N_-}$ for those y_i= -1.
3. For $t = 1:T$ boosting round
   3.1 Bootstrap sampled $20\times d$ (d=32 in 2×2 SURF patch) positive samples and $20\times d$ negative samples from training set according to the weight;
   3.2 Parallel for each patch $\{\mathbf{x}_i^k, y_i\}$, train a logistic regression model $h_k(\mathbf{x}, \mathbf{w})$;
   3.3 For each h_k, combine it with existing model H^{t-1}; and evaluate on the whole training set to obtain AUC score $J(H^{t-1} + h_k)$;
   3.4 Choose best model $H^t(\mathbf{x})$ according to Eq.3;
   3.5 Test $H^t(\mathbf{x})$ on the whole training set to get $H^t(\mathbf{x_i})$ and AUC score J_t;
   3.6 Define error $e_t = 1 - J_t$, and $\alpha_t = \frac{1}{2}\ln\frac{1-e_t}{e_t}$;
   3.7 Update weight $w_{t,i} = w_{t,i}\exp[-y_i\alpha_t H^t(\mathbf{x_i})]$ for each sample;
   3.8 Normalize the weight such that $\sum_i w_{t,i}^{+} = 1$ and $\sum_i w_{t,i}^{-} = 1$;
   3.9 If AUC value J_t is converged, break the loop.
4. Output final strong model $H^T(\mathbf{x})$ for this stage.

The procedure in one stage is forward selection of weak classifier over possible local patches. It is simple to extend the procedure with backward removing capability, which is not only able to shrink the number of weak classifier in each stage, but also able to improve the generalization capacity of the strong classifier. For details on including backward removing or even floating searching into boosting framework, please refer to [15].

Within one stage, we did not need to give threshold for each weak classifier. We just need to determine the decision threshold θ of the strong classifier $H^t(\mathbf{x})$. In the VJ framework [32, 37], the threshold is manual tuned on a validation set based on the two conflicting criteria, false-positive rate and detection rate. In our case, the threshold is much easier to be determined since ROC curve is consistently available at any stage[6]. With ROC curve, the false-positive-rate is easily determined when given minimal detection-rate d_{min}. We decreased d_i from 1.0 on the ROC curve, until reached the transit point that d_i equals to d_{min}. The false-positive-rate f_i is automatic determined at this point on the curve.

After one stage is converged, we continued to train another stage with false-positive samples coming from scanning non-face images with existing partial cascade until the overall false-positive rate reaches the final goal. Table 2 summarizes the cascade training algorithm based on ROC analysis.

With this framework, false-positive-rate f_i is adaptive among different stages, and thus the cascade may converge much faster. Let's give an example, when setting global FPR f_i = 0.5, we know that the VJ framework requires at least 20 stages to reach overall false positive rate 1e-6. However, in our case, f_i is adaptive among stages, and it is much smaller than 0.5 in most stages. For instance, we get an 8-stage cascade with f_i at each stage forming a vector like $(0.305, 0.226, 0.147, 0.117, 0.045, 0.095, 0.219, 0.268)$. Hence, the AUC based cascade optimization can converge with stage number as fewer as 8. This will not only increase the detection speed, but also make model-size very smaller (As an example, the 8-stage SURF cascade model may be as smaller as 50KB).

4. Experiments

In this section, we will show the implementation details, evaluation results on public data sets and the detection speed of the proposed SURF cascade.

4.1. Implementation Details

We implemented all the training and detection modules in C/C++ on x86 platform. For the feature extraction part, we adopted the integral image technique to speedup the computation as we described in Section 3.1. For the cascade training part, we parallelized the time-consuming weak classifier training step (step 3.2 in Table 1) with OpenMP, which is a very simple task-level parallelization. Furthermore, we did some optimization on the vector operations using SIMD. This includes the L_2 vector normalization of the SURF features and the vector dot-product in logistic

Table 2. Training the cascade based on ROC

- Input: over all false-positive-rate F_t; minimum detection-rate per stage d_{min}; set of positive samples Pos; set of negative samples Neg.
- Initialize $F_i = 0$, $D_i = 0$, i=0.
- while $F_i < F_t$
  1. $i = i + 1$;
  2. Train one stage boosting model $H^i(\mathbf{x})$ using Pos and Neg samples with algorithm in Table 1;
  3. Evaluate the model $H^i(\mathbf{x})$ on the whole training set to obtain ROC curve;
  4. Determine the threshold θ_i by searching on the ROC curve to find the point (d_i, f_i) such that $d_i = d_{min}$;
  5. $F_{i+1} = F_i \times f_i$ and $D_{i+1} = D_i \times d_i$;
  6. Empty the set Neg;
  7. If $F_{i+1} > F_t$, adopt current cascaded detector to scan non-target images with sliding detection window and put false-positive samples into the set Neg until the size $|Neg| > |Pos|$.
- Output the boosting cascade detector $\{H^i(\mathbf{x}) > \theta_i\}$ and overall training accuracy F and D.

model (i.e., $\mathbf{w^T x}$). The detection module shares the same SIMD optimization with the training part.

The training and detection experiments were done on a personal workstation with 3.2GHz Core-i7 CPU (4 cores 8 threads) and 12GB RAM.

This paper demonstrated the effectiveness of the proposed approach on frontal face detection. We collected training samples from Internet. The positive samples included frontal faces from the GENKI dataset[29], the facetracer dataset [13] and some of in-house photo-album collections (about 1000 face images). We cropped 13000 frontal faces from these datasets. We further derived 13000 faces with horizontal mirrors, and 13000 faces with downsampling 10% followed by upsampling 10%. All cropped and derived frontal faces are resized to 40×40. Finally, we had 39000 positive training samples. The negative images are collected from Caltech 101 dataset[7], SIMPLICITY image retrieval dataset[34] and ImageNet[5]. Totally, we collected about 18000 images without faces, with image resolution range from 10k pixels to 1M pixels.

We placed 396 local patches on the 40×40 detection template as described in section 3.1. We set the maximum number of weak learners in each stage to 128. To obtain fast detector, we only allowed at most 4 weak learners in the first 3 stages. We set the minimum detection-rate to 0.995 for each stage and the overall false-positive-rate per window (FPPW) to 1e-6. Given these parameters, the training procedure is fully automatic. It took about 47 minutes (2841 seconds) to converge at the 8th stages. During the training procedure, we scanned both the original and the horizontal mirrors of each negative image with a size-scalable sliding detection window to get false-positive samples. In the end, more than 13.6 billions of negative samples were scanned.

In comparison, we tried to train face detector using the OpenCV haar training modules on the same training set. However, we can't finish the training within 2 days on the same computer with parallel processing tuned on. Besides, we also tried the same strategy in the VJ framework (i.e., the two conflicting criteria) to control the SURF cascade training instead of the AUC criterion. We found that the VJ strategy for training SURF cascade required more than 5 hours to converge at the 19th stages. This experiment shows the benefit of AUC criterion for cascade training.

Figure 1(a) and 1(b) illustrate details of the final cascade, including the number of weak learner in each stage and the accumulated rejection rate over the cascade stages. It shows the first three stages rejects 98% negative examples with only 7 weak classifiers. The cascade detector contains 334 weak classifiers, and only need to evaluate 1.5 per negative window. On the contrary, the default haar-based face detector in OpenCV contains more than 24 stages and 2912 weak classifiers, and is required to evaluate more than 28 haar-features per negative window [16].

We further illustrate the top-3 most discriminative local patches that the system selected in Figure 3. We observed that the best local patch locates in the regions of two eyes. This is similar to that of Haar-based detector [32].

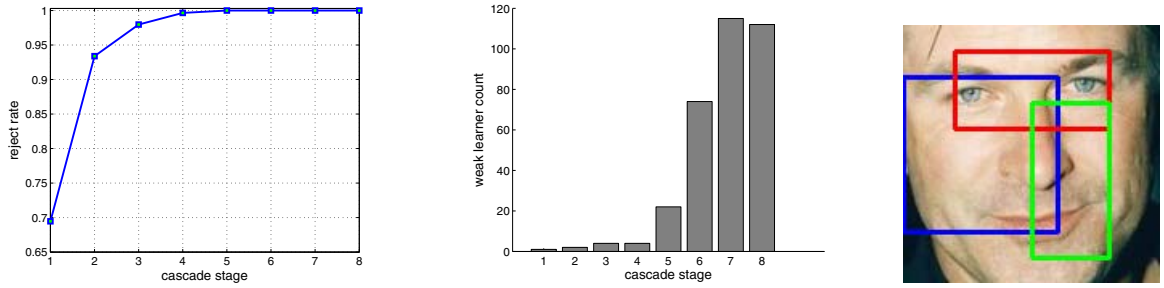
The final detector contains a post-processing step, which merges cascade results using the disjoint-set algorithm and filters unreliable results using the non-maximum suppression algorithm.

4.2. Face Detection Evaluation

We evaluated the accuracy of SURF detector on two public data-sets: first is the CMU+MIT data set, the other is the UMass FDDB data set [12].

The standard CMU+MIT data set consists of 130 gray images with 507 frontal faces. As SURF cascade can directly output probability score (in the range 0∼1) at any stage, it is natural to define score for each detection window w as $s(w) = p(w) + k(w)$, where $k(w)$ is the number of passed stages and $p(w)$ is probability output of the exiting stage. With this score, we strictly followed the benchmark protocol suggested in [12] [2], and generated the ROC curve as shown in Figure 2. Comparable results are depicted for some recent works in face detection such as the VJ detector [33], polygon-feature detector [24], soft cascade detector [2] and recycling-cascade detector [3]. Figure 2 shows that SURF cascade achieves comparable performance to the

[2] We implemented a benchmark tool for CMU+MIT data-set based on the three criteria in section 6 of [12] with some modifications of the original FDDB evaluation code, since CMU+MIT does not provide ground truth as elliptical regions.



(a) (b) (c)

Figure 1. (a)The number of weak classifiers at each stage of the face detector, (b) the accumulated rejection rate over the cascade stages of the face detector; (c)Top-3 local patches selected by training procedure in the red-green-blue order.

state-of-the-art method soft-cascade [2], while outperforms all the other methods.

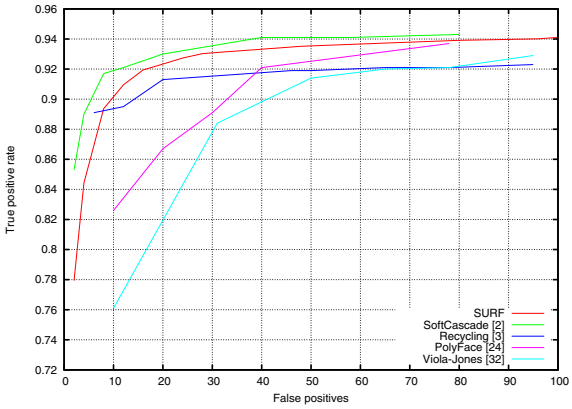


Figure 2. ROC curves for different methods on CMU+MIT dataset

The CMU+MIT dataset is a little out-of-date as it only contains gray, relative low-resolution images, and the size of the data set is too small to reflect nowadays data explosion status. Hence, the UMass face detection data-set and benchmark (FDDB) is introduced [12]. It contains 2845 images with a total of 5771 faces under a wide range of conditions. Besides, it provides a systematic protocol to evaluate performance of face detection system. Figure 3 shows the discrete score and continuous score ROC curve generated by SURF cascade detector in comparison to available results on the benchmark [31, 16, 20]. [3]. It is obvious that SURF cascade outperforms others significantly. Furthermore, we believe that when multi-view SURF-based face detector is trained, the result would be even better.

Figure 4 illustrates some examples of face detection results on CMU+MIT and UMass FDDB datasets.

[3] Some detectors did not shared their result files on the website, however it is obvious that we have much better performance than others except the one by Olaworks inc., which should utilize multi-view information.

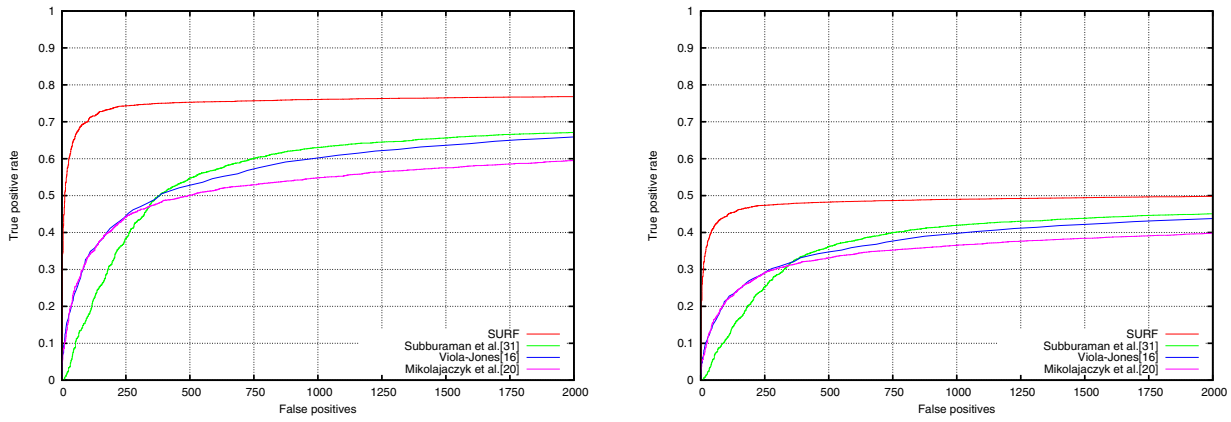
4.3. Detection Speed

Finally, we further made comparison of detection speed between the proposed detector and the OpenCV face detector. As is known, the OpenCV face detector was well tuned in speed. We tested the two detectors on three video clips (about 2∼3 minutes each) under pure detection mode with the same parameters (scale factor 1.2 for scaling detection window and minimum detection window size 40×40). We first compared the detection speed on a low-end single core Intel Atom 1.6GHz CPU. The first 3 rows of Table 3 list the compared results. Note that video B and C have different number of faces in each frames. On average, video B has 3 frontal faces in one frame while video C has only 1 frontal face in one frame. We further compared the speed on the high-end Core-i7 CPU with parallel processing enabled on both detectors. The results are listed in the last 3 rows of Table 3. Note that we only counted the computing time in detection part without taking any other computing (such as video decoding) into consideration on both detectors.

Two points can be found from the table. First, our detector is faster than OpenCV default detector. This is due to the facts that ours detector has fewer number of stages (8 vs 24), fewer number of weaker classifiers (334 vs 2916) and fewer average number of evaluated weak classifiers per detection window (1.5 vs 28) than that of haar-based detector. These advantages can compensate one weakness point that the computing on each weak classifier is larger than that of OpenCV detector. Second, the parallel scaling of the proposed detector is better than that of OpenCV detector. This is due to two facts. First, the fewer stages of SURF-cascade may yield better load balance among threads than that of longer stages of Haar-cascade. Second, SURF-cascade benefits much more from the SIMD optimization (i.e., $\mathbf{w}^{\mathbf{T}}\mathbf{x}$ in logit model) than that of OpenCV.

5. Conclusions

This paper presents a novel boosting cascade framework based on SURF features for face detection. The major con-



(a) (b)

Figure 3. (a)Discrete score ROC curves and (b) Continuous score ROC curves for different methods on UMass FDDB dataset.

Table 3. Detection speed comparison between OpenCV default detector and SURF-cascade. The first three rows on 1.6GHz Atom CPU, and the rest on the 3.2GHz Core i7.

Videos	resolution	Ours (fps)	OpenCV (fps)
A.avi	352×288	30.1	25.0
B.mpg	640×480	5.8	4.6
C.avi	640×480	7.3	5.4
A.avi	352×288	312	190
B.mpg	640×480	71.3	49.7
C.avi	640×480	90.5	58.2

tributions are three points. First, we introduce SURF features for fast face detection. Second, we proposed AUC as the single criterion for cascade training, which leads to fast training convergence. Third, we showed a real example that can train fast and accurate cascade face detector from billions of samples within one hour on personal computers. This is a big improvement over the Viola-Jones framework.

We also compared the detector accuracy and speed with existing algorithms. It shows SURF cascade can achieve results comparable to state-of-the-art detectors. Furthermore, the speed of the detector even outperforms the tailored optimized OpenCV detector.

Future work may consider extending the approach on other more complex object detection task such as human detection and multi-view object detection.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359, 2008. 2, 3

[2] L. Bourdev and J. Brandt. Robust object detection via soft cascade. In *CVPR*, 2005. 1, 2, 5, 6

[3] S. C. Brubaker, J.Wu, J. Sun, M. D.Mullin, and J. Rehg. On the design of cascades of boosted ensembles for face detection. *IJCV*, 2008. 2, 3, 5

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2, 3

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[6] T. Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine Learning*, 2004. 2, 4

[7] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *CVPR workshop on Generative-Model Based Vision*, 2004. 5

[8] P. Felzenszwalb, D.McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 2

[9] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010. 2

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 2

[11] C. Huang, H. Ai, Y. Li, and S. Lao. Learning sparse features in granular space formulti-view face detection. In *AFGR*, 2006. 2

[12] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010. 5, 6

[13] N. Kumar, P. N. Belhumeur, and S. K. Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2008. 5

[14] I. Laptev. Improvements of object detection using boosted histograms. In *bMVC*, 2005. 3

[15] S. Li, Z. Zhang, and et al. Floatboost learning for classification. In *NIPS*, 2002. 4

[16] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *ICIP*, 2002. 2, 5, 6

[17] P. Long and R. Servedio. Boosting the area under the roc curve. In *NIPS*, 2007. 2

[18] S. Maji and A. C. Berg. Max-margin additive classifiers for detection. In *ICCV*, 2009. 1

(a) (b) (c) (d)

Figure 4. Example detection results on CMU+MIT (a) and UMass FDDB (b,c,d) datasets.

[19] B. McCane and K. Novins. On training cascade face detectors. *Image and Vision Computing*, 2003. 2

[20] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *ECCV*, 2004. 2, 6

[21] T. Minka. A comparison of numerical optimizers for logistic regression. Technical report, Microsoft Research, 2004. 3

[22] T. Mita, T. Kaneko, and O. Hori. Joint haar-like features for face detection. In *ICCV*, 2005. 2

[23] M.-T. Pham and T.-J. Cham. Fast training and selection of haar features during statistics in boosting-based face detection. In *ICCV*, 2007. 2

[24] M.-T. Pham, Y. Gao, V. D. Hoang, and T.-J. Cham. Fast polygonal integration and its application in extending haar-like features to improve object detection. In *CVPR*, 2010. 1, 2, 5

[25] C. Shen, P. Wang, and H. Li. Lacboost and fisherboost: optimally building cascade classifiers. In *ECCV*, 2010. 3

[26] J. Sochman and J. Matas. Waldboost - learning for time constrained sequential detection. In *CVPR*, 2005. 1, 2

[27] A. Sorokin and D. A. Forsyth. Utility data annotation with amazon mechanical turk. In *First Internet Vision Workshop*, 2008. 1

[28] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006. 2

[29] http://mplab.ucsd.edu. The MPLab GENKI Database. 5

[30] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 1

[31] S. M. Venkatesh Bala Subburaman. Fast bounding box estimation based face detection. In *ECCV workshop on face detection*, 2010. 6

[32] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 1, 2, 3, 4, 5

[33] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 1, 2, 5

[34] J. Z. Wang, J. Li, and G. Wiederhold. Simplicity: Semantics-sensitive integrated matching for picture libraries. *IEEE TPAMI*, 23(9):947–963, 2001. 5

[35] R. Xiao, H. Zhu, H. Sun, and X. Tang. Dynamic cascades for face detection. In *ICCV*, 2007. 1, 2, 4

[36] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical Report MSR-TR-2010-66, Microsoft Research, 2010. 1

[37] Q. Zhu, S. Avidan, M.-C. Yeh, and K.-T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *CVPR*, 2006. 1, 2, 3, 4