Robust Face Detection with Multi-Class Boosting

Yen-Yu Lin Tyng-Luh Liu
Institute of Information Science, Academia Sinica, Nankang, Taipei 115, Taiwan
{*yylin, liutyng*}*@iis.sinica.edu.tw*

Abstract

With the aim to design a general learning framework for detecting faces of various poses or under different lighting conditions, we are motivated to formulate the task as a classification problem over data of multiple classes. Specifically, our approach focuses on a new multi-class boosting algorithm, called MBHboost, and its integration with a cascade structure for effectively performing face detection. There are three main advantages of using MBHboost: 1) each MBH weak learner is derived by sharing a good projection direction such that each class of data has its own decision boundary; 2) the proposed boosting algorithm is established based on an optimal criterion for multi-class classification; and 3) since MBHboost is flexible with respect to the number of classes, it turns out that it is possible to use only one single boosted cascade for the multi-class detection. All these properties give rise to a robust system to detect faces efficiently and accurately.

1. Introduction

Accuracy and *efficiency* are two of the most important issues in evaluating a face detection system. For accuracy, a number of learning techniques have been used to accomplish satisfactory results, including, e.g., SVMs [8], neural network [9], multi-layer perceptron [15], Fisher linear discriminant [18]. For efficiency, Viola and Jones [17] introduce a framework to elegantly combine Adaboost with a cascade scheme to detect faces in real time. Subsequently, several variants have been proposed to extend or improve the detection through a boosted cascade, [4], [6], [14].

The foregoing works consider mainly one *type/class* of faces, e.g., frontal faces. Such a restriction may limit their practical use because faces in images can occur with various poses like in-plane or out-of-plane rotations, or under various situations such as lighting conditions, expressions, and occlusions. So, the visual appearances and features of faces could vary significantly with respect to different scenarios/circumstances. It is therefore more reasonable to formulate a face detection task as a *multi-class* learning problem. With that, we aim to address the task in a general manner without compromising accuracy and efficiency.

1.1. Previous Work

According to the system structure and the mechanism, we divide the following recent works that detect faces based on multi-class learning into three categories.

View-based systems. In general, for such approaches, multiple detectors are specifically designed and independently trained so that each can deal with one particular class of faces. A testing pattern will be examined in parallel by all the detectors, and the outcome is the union of their respective detection results. Schneiderman and Kanade [12] train view-based detectors with features formed by a set of wavelet coefficients. Their system has been shown to achieve high accuracy rates for profile face detection. Nevertheless, the computation time of a view-based system is directly proportional to the number of detectors used, i.e., the number of face types it aims to handle.

Estimate-before-detect systems. By investigating the *dissimilarities* among faces from different viewpoints, the pose of an input pattern can be first estimated. Then, according to the estimation, the pattern is dispatched to the corresponding detector for further verification. Rowley et al. [10] use two separate neural networks to carry out this strategy for detecting faces with in-plane rotations. Jones and Viola [2] apply the C4.5 decision tree followed by cascaded detectors to handle faces with in-plane or out-of-plane rotations. Clearly, using the scheme, the computation time is reduced to one estimation and one detection instead of multiple parallel detections. The main disadvantage is that it is generally difficult to establish a reliable (pose) estimator, especially when the number of face classes is large, that guarantees both low computational costs and high accuracy rates.

Classifier-sharing systems. The *similarities* among different types of faces can also be explored to enhance the detection efficiency. Li et al. [3] propose a *detector-pyramid* system to detect profile faces. Depending on its level in the pyramid, each classifier can be shared by a collection of face classes. In a related work [5], we have proposed an *evidence cascading* scheme to detect faces with occlusions. The idea is motivated by the fact that outside the occluded regions, faces with partial occlusions have the same features as the regular frontal faces, and the information can be used to design a classifier sharing mechanism to detect

faces with different areas of occlusions. The drawback of the two systems is that the underlying relations among face classes need to be identified before training. Consequently, the resulting rules for classifier sharing may not be general enough to be extended to detect other additional types of faces, e.g., faces under different lighting conditions.

1.2. Our Approach

Instead of treating a general face detection task as solving many individual binary classification problems, we propose a multi-class boosting to directly address the underlying difficulties. Specifically, the novelty of our approach consists in three key techniques. First, we introduce the multi-class Bhattacharyya (MBH) weak learners that is *vector-valued* and applicable to simultaneously classify each class of data. Unlike the direct sharing of a weak learner among all classes, we propose to share a good projection direction to take account of the diverse distributions from different classes of data. Second, we establish the MBHboost algorithm that optimally minimizes the weighted error upper bound of all classes at each boosting iteration. The effectiveness of the resulting multi-class boosting algorithm is guaranteed by an optimal criterion that we will later prove analytically. Third, since the MBHboost is flexible with respect to the number of classes of the data, we can derive a detection structure that uses only a single boosted cascade for multi-class classifications.

Figure 1: Nine different types/classes of profile faces.

2. Face Detection with MBHboost

Recall that a binary boosting algorithm is derived basically from iteratively separating the weighted positive and weighted negative training data. To generalize this idea to deal with multi-class data, we aim to establish an iterative process, the MBHboost, that simultaneously separates the weighted face and non-face data of each type. Thus the focal point of our discussion in this section is to explain how MBHboost accomplishes the goal and retains its efficiency in performing multi-class classifications. And that in turn has to do with the use of MBH weak learners. For convenience, we shall illustrate the formulation with an example of profile face detection, where the face data are categorized into nine types/classes, shown in Figure 1, according to their rotated angles. Other scenarios of face detection will be further explored in Section 4.

2.1. MBH Weak Learners

We focus on weak learners that involve 1-D projection directions and derive their outputs by analyzing the projected training data. Recent works, e.g., [5], [6], have demonstrated these weak learners can be efficiently combined to accurately detect frontal faces. Noteworthily, the real-valued BH weak learners in [5] can be further extended into vector-valued MBH weak learners, and we will give an analytic proof that the resulting MBHboost satisfies an optimal criterion for multi-class classification.

Consider now a typical binary classification problem for detecting one class of faces. Let training data $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)\} = D^+(\text{face}) \cup D^-(\text{non-face})$, and Φ be the set of all possible projection directions. We also need to maintain a weight vector w_t over D at each boosting iteration t. Through any projection $\phi \in \Phi$ and w_t, the projected data $\phi(D)$ form two weighted histograms $p^+(\phi)$ and $p^-(\phi)$ over a bounded segment of the real line with m equal-size bins $\{b_k\}_{k=1}^m$. More precisely, in each bin b_k, the two weighted histograms are computed as follows:

$$p_k^+(\phi) = \sum\nolimits_{\mathbf{i}_k^+} w_t(i) \quad \text{and} \quad p_k^-(\phi) = \sum\nolimits_{\mathbf{i}_k^-} w_t(i), \quad (1)$$

where $\mathbf{i}_k^{+(-)} = \{i \,|\, \mathbf{x}_i \in D^{+(-)}, \, \phi(\mathbf{x}_i) \in b_k\}$, the indexes of *positive* (*negative*) training data projected into bin b_k by ϕ. In [5], the projection with the minimal Bhattacharyya coefficient (BHC) between its two weighted histograms is selected at each iteration. The choice has the property that the error upper bound of boosting, i.e., $\prod Z_t$ in [11], will be iteratively minimized if the output of the weak learner associated with the selected projection ϕ is defined by

$$h(\mathbf{x}) = \ln \sqrt{p_k^+(\phi)/p_k^-(\phi)}\,, \text{ if } \phi(\mathbf{x}) \in b_k. \quad (2)$$

Indeed, each projection in Φ uniquely corresponds to a rectangle feature, and therefore can be efficiently evaluated by referencing the integral image in constant time [17]. In Figure 2a, the projection ϕ_1 is induced by a rectangle feature, and gives rise to two weighted histograms, $p^+(\phi_1)$ and $p^-(\phi_1)$, shown in Figure 2b. The respective weak learner can be constructed according to (2).

For the multi-class case, the example of detecting profile faces serves as a good starting point to examine the core of our proposed method. Let $\Gamma = \{\mathcal{A}, \mathcal{B}, ..., \mathcal{I}\}$ denote the set of nine profile face types, shown in Figure 1, and $|\Gamma| = 9$. For each $\mathcal{X} \in \Gamma$, the type-$\mathcal{X}$ training set is labeled as $D^{\mathcal{X}} = \{(\mathbf{x}_1^{\mathcal{X}}, y_1^{\mathcal{X}}), \ldots, (\mathbf{x}_{|D^{\mathcal{X}}|}^{\mathcal{X}}, y_{|D^{\mathcal{X}}|}^{\mathcal{X}})\} = D^{\mathcal{X}+}$ (type-$\mathcal{X}$ face, i.e., $y^{\mathcal{X}} = 1$) $\cup\, D^{\mathcal{X}-}$ (non-face, i.e., $y^{\mathcal{X}} = -1$). Clearly, $D^{\Gamma} = \bigcup_{\mathcal{X}\in\Gamma} D^{\mathcal{X}}$. Analogous to the binary classification, a weight vector $w_t^{\mathcal{X}}$ is updated over $D^{\mathcal{X}}$ at each MBHboost iteration. However, each projection ϕ over D^{Γ} now produces 18 weighted histograms (each type of training set forms two histograms). In Figures 2d, 2f, we respectively plot six weighted histograms for illustration, yielded

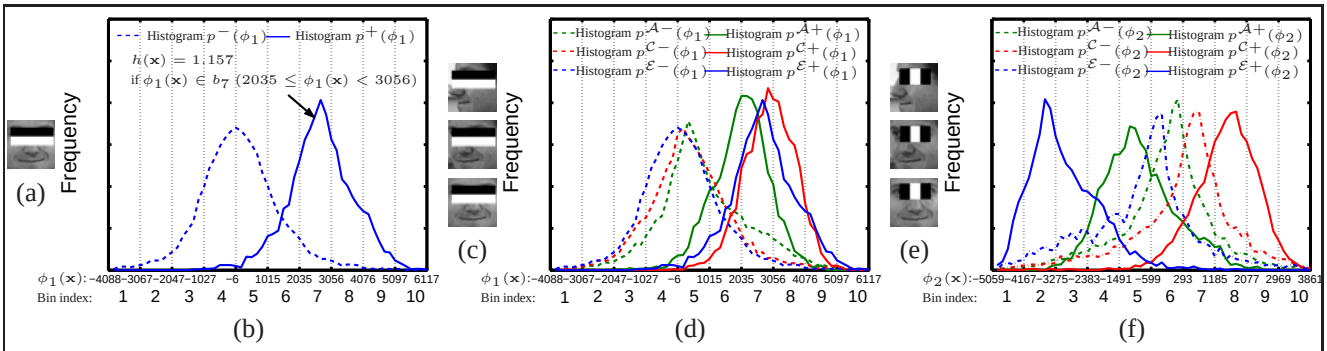


Figure 2: (a) A projection ϕ_1 on a frontal face sample. (b) Two weighted histograms, $p^+(\phi_1)$ and $p^-(\phi_1)$, are derived from the projected training set $D = D^+ \cup D^-$ by ϕ_1. (c) ϕ_1 on face samples from profile face types $\mathcal{A}$, $\mathcal{C}$, and $\mathcal{E}$, respectively. (d) Six weighted histograms derived from $D^{\mathcal{A}}$, $D^{\mathcal{C}}$, and $D^{\mathcal{E}}$. (e) & (f) Histograms due to another projection ϕ_2.

by projecting three classes of data with projections ϕ_1 and ϕ_2 shown in Figures 2c, 2e. Understandably, what remains to be addressed now is to come up with a definition to appropriately link each projection ϕ over training data D^Γ to its corresponding multi-class weak learner.

In [3], [5], each weak learner used in the boosting procedure is shared to detect a *pre-defined* subset of classes. That is, expert knowledge is required to sort the subsets before training the system. Torralba et al. [16] instead propose to share a weak learner by the most suitable subset that can be derived by *searching* among all possible $2^{|\Gamma|} - 1$ candidates. Even if an approximated search is used, the number of candidates is still around $\mathcal{O}(|\Gamma|^2)$. More critically is that among these works, each weak learner is *directly* shared, i.e., the decision boundary and the output value are the same for all sharing classes, and it could restrict the use of some discriminant weak learners. For instance, consider the projection ϕ_2 and the three pairs of histograms shown in Figures 2e, 2f. Although ϕ_2 is discriminant to separate the positive and the negative data of each class, its output value of each bin is hard to be shared since the three pairs of histograms are far differently distributed.

In view of the issues described above, we relax the idea of sharing a weak learner to sharing only a discriminant projection direction. Specifically, for each projection ϕ, the vector-valued MBH weak learner f is defined as follows:

$$f(\mathbf{x}) = [h^{\mathcal{A}}(\mathbf{x}), ..., h^{\mathcal{I}}(\mathbf{x})] \quad \text{(profile faces)}, \quad (3)$$

$$= [h^{\mathcal{X}}(\mathbf{x}) \,|\, \mathcal{X} \in \Gamma] \quad \text{(general case)}, \quad (4)$$

$$h^{\mathcal{X}}(\mathbf{x}) = \ln \sqrt{p_k^{\mathcal{X}+}(\phi)/p_k^{\mathcal{X}-}(\phi)} \ \text{ if } \phi(\mathbf{x}) \in b_k, \quad (5)$$

where $\mathcal{X} \in \Gamma$, and the meaning of $p_k^{\mathcal{X}+(-)}(\phi)$ is the same as $p_k^{+(-)}(\phi)$ in (1), except the training data are limited only to $D^{\mathcal{X}}$. From (4), an MBH weak learner consists of $|\Gamma|$ components that share a common projection ϕ: each component learns its own decision boundary and computes the output in each bin using the corresponding pair of weighted histograms. In what follows, we summarize some useful properties of MBH weak learners.

- An MBH weak learner is applicable to *all* classes of data. No expert knowledge or search techniques are needed to identify a subset of classes for sharing.
- Unlike direct sharing in [3], [5], [16], each MBH component *independently* learns outputs for each class of training data, and consequently achieve better classification efficiency and flexibility.
- On the other hand, because all MBH components share a same projection, they reference the same bin index k in (5) for an arbitrary pattern $\mathbf{x}$. Furthermore, since the output value of each component for any bin k has been learned in the training phase, the main computation cost of evaluating an MBH weak learner in testing is simply to find the value of k. In other words, the extra computation cost to extend weak learners to multi-class in our scheme is relatively low.

2.2. An Optimal Criterion

Having described the details of MBH weak learners, we are now ready to formalize the multi-class boosting algorithm. Our discussion will focus on an optimal criterion that is the cornerstone of the efficiency of the proposed MBHboost. We first begin with a definition to measure the difficulty of classifying each class of training data.

Definition 1 *At boosting iteration t, the difficulty to classify the type-$\mathcal{X}$ data of a multi-class training set is given by $\Delta_t^{\mathcal{X}} = \sum_{i=1}^{|D^{\mathcal{X}}|} \exp\left(-y_i^{\mathcal{X}} H_{1:t-1}^{\mathcal{X}}(\mathbf{x}_i^{\mathcal{X}})\right)$, where $H_{1:t-1}^{\mathcal{X}} = \sum_{\tau=1}^{t-1} h_\tau^{\mathcal{X}}$ is the intermediate classifier for $D^{\mathcal{X}}$, derived by combining the type-$\mathcal{X}$ components of the $t-1$ MBH weak learners selected in previous iterations.*

Algorithm 1: *MBHboost*

Input : Training data D^{Γ}; Projection set Φ; Number of iterations T.

Output : A vector-value MBH classifier F.

$w_1^{\mathcal{X}}(i) = 1/|D^{\mathcal{X}}|, \forall i = 1, 2, \ldots, |D^{\mathcal{X}}|$ *and* $\forall \mathcal{X} \in \Gamma$;

for $t \leftarrow 1, 2, \ldots, T$ **do**

1. Determine the optimal ϕ_t by solving (6);
2. Derive MBH weak learner f_t based on (4), (5);
3. $w_{t+1}^{\mathcal{X}}(i) \leftarrow w_t^{\mathcal{X}}(i) \exp\left(-y_i^{\mathcal{X}} h_t^{\mathcal{X}}(\mathbf{x}_i^{\mathcal{X}})\right) / Z_t^{\mathcal{X}}$, $\forall i = 1, 2, \ldots, |D^{\mathcal{X}}|$ and $\forall \mathcal{X} \in \Gamma$; (Note $Z_t^{\mathcal{X}}$ is a normalization factor to make $w_{t+1}^{\mathcal{X}}$ a distribution.)

Output $F = \sum_{t=1}^{T} f_t = [H^{\mathcal{A}}, H^{\mathcal{B}}, H^{\mathcal{C}}, \ldots]$
$= [H^{\mathcal{X}} = \sum_{t=1}^{T} h_t^{\mathcal{X}} \mid \mathcal{X} \in \Gamma]$;

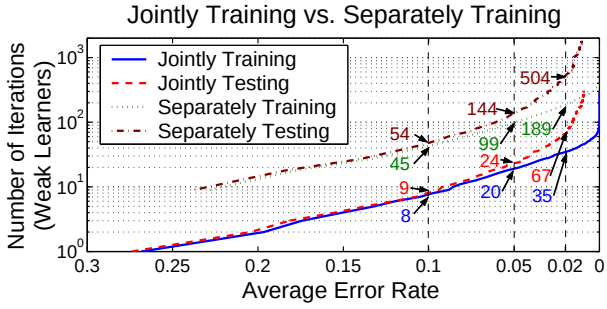


Figure 3: For a given average error rate, say 0.1, it takes 8 weak learners for an MBHboost detector. To achieve the same rate in testing, it requires 9 weak learners. In the case of using separately-trained detectors, it takes 45 and 54 weak learners, respectively.

The $\Delta_t^{\mathcal{X}}$ so defined is indeed proportional to the exponential loss of using $H_{1:t-1}^{\mathcal{X}}$ to classify the type-$\mathcal{X}$ data. Thus it serves a reasonable indicator to measure the classification complexity for type-$\mathcal{X}$ training data up to iteration t. With Definition 1, we can formulate the following criterion to iteratively derive an optimal projection ϕ_t for constructing the corresponding MBH weak learner f_t.

Theorem 1 *MBHboost is guaranteed to iteratively minimize the weighted error bound for multi-class classifications if ϕ_t stated in step 1 of Algorithm 1 is given by:*

$$\phi_t = \underset{\phi \in \Phi}{argmin} \sum\nolimits_{\mathcal{X} \in \Gamma} \Delta_t^{\mathcal{X}} \times \mathbf{BHC}_t^{\mathcal{X}}(\phi), \tag{6}$$

$$\textit{where } \mathbf{BHC}_t^{\mathcal{X}}(\phi) = \sum\nolimits_{k=1}^{m} \sqrt{p_k^{\mathcal{X}+}(\phi) p_k^{\mathcal{X}-}(\phi)}. \tag{7}$$

Proof: By recursively applying the relation between $w_t^{\mathcal{X}}$ and $w_{t+1}^{\mathcal{X}}$ in step 3 of Algorithm 1, we have

$$\begin{aligned} \Delta_t^{\mathcal{X}} =& |D^{\mathcal{X}}| \sum_{i=1}^{|D^{\mathcal{X}}|} w_1^{\mathcal{X}}(i) \exp(-y_i^{\mathcal{X}} \sum_{\tau=1}^{t-1} h_\tau^{\mathcal{X}}(\mathbf{x}_i^{\mathcal{X}})) \\ =& |D^{\mathcal{X}}| Z_1^{\mathcal{X}} \sum_{i=1}^{|D^{\mathcal{X}}|} w_2^{\mathcal{X}}(i) \exp(-y_i^{\mathcal{X}} \sum_{\tau=2}^{t-1} h_\tau^{\mathcal{X}}(\mathbf{x}_i^{\mathcal{X}})) = \cdots \\ =& |D^{\mathcal{X}}| Z_1^{\mathcal{X}} \cdots Z_{t-1}^{\mathcal{X}} \sum_{i=1}^{|D^{\mathcal{X}}|} w_t^{\mathcal{X}}(i) = |D^{\mathcal{X}}| \prod_{\tau=1}^{t-1} Z_\tau^{\mathcal{X}}. \end{aligned} \tag{8}$$

On the other hand, the Bhattacharyya coefficient can be shown to satisfy $\mathbf{BHC}_t^{\mathcal{X}}(\phi) = Z_t^{\mathcal{X}}/2$ (refer to equation (3) in [5] for details). Thus the criterion in (6) becomes

$$\sum_{\mathcal{X} \in \Gamma} \Delta_t^{\mathcal{X}} \times \mathbf{BHC}_t^{\mathcal{X}}(\phi) = \frac{1}{2} \sum_{\mathcal{X} \in \Gamma} |D^{\mathcal{X}}| \prod_{\tau=1}^{t} Z_\tau^{\mathcal{X}}. \tag{9}$$

From the theorem, the criterion (6) at each iteration t is to choose an optimal 1-D projection direction ϕ_t that minimizes the weighted error upper bound of multi-class boosting (9), where the weight of each class depends on the number of its training samples. When all classes have the same number of data, the criterion is reduced to minimizing an *average* error bound of all classes, i.e., $\sum_{\mathcal{X} \in \Gamma} \prod_{\tau=1}^{t} Z_\tau^{\mathcal{X}}$, and when the training data are of single class of positive and negative ones, it becomes the error upper bound that Adaboost is designed to iteratively minimize.

Another advantage of MBHboost is that it usually takes fewer weak learners to carry out a multi-class classification than those required by a typical view-based approach. Note that each component of an MBH classifier F implicitly corresponds to a detector for its own class of data. Therefore, via the sharing of ϕ_t at each t, all the component-wise detectors, $H^{\mathcal{X}} = \sum_{t=1}^{T} h_t^{\mathcal{X}}$, can be trained *jointly*. Conversely, each view-based detector needs to be trained *separately* for a particular class of training data. In Figure 3, the view-based detectors for the example frontal face detection are independently derived using Algorithm 1 with each single class of training data. Though, class-wise, a detector trained separately has a faster convergency speed, the total number of weak learners required in all $|\Gamma| = 9$ detectors for achieving a given error rate is much larger. In the plot, we record the training and testing error rates at each iteration for the two ways of training. By considering some fixed average error rates, say 10%, 5%, and 2%, it is clear that training jointly is more efficient and uses fewer weak learners.

3. The Detection Architecture

To detect faces from various scenarios in real time, we consider a detection structure based on the boosted cascade [17]. More specifically, with MBHboost we need to train

Algorithm 2: *Multi-Class Cascade: Training*

Input : Training data D^{Γ}; Images without faces Q;
μ, ν: target detection, false-positive rate.

Output : A cascade of MBH classifiers $\{F_1, ..., F_s\}$.

$k \leftarrow 1$; $\Gamma_k \leftarrow \Gamma$;
while $\Gamma_k \neq \emptyset$ **do**
  With D^{Γ_k} and Φ, use Algorithm 1 to derive $F_k = [\, H_k^{\mathcal{X}} \mid \mathcal{X} \in \Gamma_k \,]$ s.t. each $H_k^{\mathcal{X}}$ meets (μ, ν);
  foreach $\mathcal{X} \in \Gamma_k$ **do**
    $D^{\mathcal{X}+} \leftarrow \{\mathbf{x} \mid \mathbf{x} \in D^{\mathcal{X}+} \wedge H_k^{\mathcal{X}}(\mathbf{x}) \geq \theta_k^{\mathcal{X}}\}$;
    $D^{\mathcal{X}-} \leftarrow$ `False-Positives` from $D^{\mathcal{X}-}$ or from Q such that $|D^{\mathcal{X}-}| = |D^{\mathcal{X}+}|$;
    if *not enough False-Positives* **then**
      $s^{\mathcal{X}} \leftarrow k$; $\Gamma_k \leftarrow \Gamma_k - \{\mathcal{X}\}$;
  $\Gamma_{k+1} \leftarrow \Gamma_k$; $k \leftarrow k + 1$;

Algorithm 3: *Multi-Class Cascade: Testing*

Input : A test pattern $\mathbf{x}$; Face classes Γ;
A cascade of detectors $\{F_1, ..., F_s\}$;
Number of stages, $s^{\mathcal{X}}$, $\forall \mathcal{X} \in \Gamma$.

Output : A vector of boolean outputs, $output(\Gamma)$.

$k \leftarrow 1$; $\Lambda \leftarrow \Gamma$;
while $\Lambda \neq \emptyset$ **do**
  Jointly evaluate $H_k^{\mathcal{X}}(\mathbf{x})$, $\forall \mathcal{X} \in \Lambda$;
  foreach $\mathcal{X} \in \Lambda$ **do**
    if $H_k^{\mathcal{X}}(\mathbf{x}) < \theta_k^{\mathcal{X}}$ **then**
      $output(\mathcal{X}) \leftarrow$ `False`; $\Lambda \leftarrow \Lambda - \{\mathcal{X}\}$;
    else if $k = s^{\mathcal{X}}$ **then**
      $output(\mathcal{X}) \leftarrow$ `True`; $\Lambda \leftarrow \Lambda - \{\mathcal{X}\}$;
  $k \leftarrow k + 1$;

only *one* cascade for the multi-class detection. This is in contrast to the systems described in [2], [3], [5] that several cascades are deployed for detecting different classes of faces/objects, and thus additional mechanisms are needed to choose the most appropriate cascade for each input pattern.

Through a boosted cascade, the task of face detection becomes a series of classification problems. In our case, there are three key factors to be carefully planned during training to ensure good detection rates, including 1) the number of MBH weak learners used in each stage k; 2) the type-specific threshold $\theta_k^{\mathcal{X}}$ for $H_k^{\mathcal{X}}$ of the classifier F_k; and 3) the total number of stages, s, for implementing the cascade. In addition, besides the training data D^{Γ} we also prepare a set Q consisting of images that contain no faces, where its function will be self-evident as we describe the approach. Suffice it to say now that Q is used to generate new non-face data by bootstrap over the course of training.

Training. Let μ denote the target detection rate and ν be the maximal false positive rate for learning a classifier F_k at stage k of the cascade structure in Algorithm 2. Thus $F_k = [H_k^{\mathcal{X}} | \mathcal{X} \in \Gamma_k]$ can be derived by jointly training and by tuning the thresholds $\theta_k^{\mathcal{X}}$ to ensure all $H_k^{\mathcal{X}}$ have detection rates above μ and false positive rates below ν. (Empirically, μ is set between $99.5\% \sim 99.9\%$, and ν is about 40%.) This would give us a way to determine the number of MBH weak learners needed to construct a desirable F_k. In practice the value of $\theta_i^{\mathcal{X}}$ is often negative, and a pattern $\mathbf{x}$ is considered a type-$\mathcal{X}$ face at stage k iff $H_k^{\mathcal{X}}(\mathbf{x}) \geq \theta_k^{\mathcal{X}}$.

Note that the number of components in F_k is not fixed and is non-increasing as k becomes larger. This is due to different degree of difficulty in classifying each class of data. Specifically, the completion of the type-$\mathcal{X}$ training over the multi-class cascade is signaled by the condition when not enough non-face data can be generated from Q to ensure $|D^{\mathcal{X}+}| = |D^{\mathcal{X}-}|$ for the subsequent stage. These additional non-face data are obtained by applying bootstrap to Q to generate false positives that pass all the k type-$\mathcal{X}$ components, i.e., $H_1^{\mathcal{X}}, H_2^{\mathcal{X}}, \ldots, H_k^{\mathcal{X}}$. Each time a type-$\mathcal{X}$ training is completed at some stage k, we have $s^{\mathcal{X}} = k$, and delete $\mathcal{X}$ from Γ_k. The training procedure continues with the updated data, and eventually finishes when Γ_k becomes empty. The total number of stages in the cascade can then be determined by $s = \max\{s^{\mathcal{X}} \mid \mathcal{X} \in \Gamma\}$. We conclude with a remark that our proposed MBHboost is flexible enough for handling the stage-wise variable number of classes of training data, and that makes it very convenient to deal with multi-class classifications with one single cascade.

Testing. A test pattern $\mathbf{x}$ is considered a type-$\mathcal{X}$ face if it passes all the type-$\mathcal{X}$ components of detectors from the first to the $s^{\mathcal{X}}$th stages. Like a view-based system, our method provides a vector of boolean responses, each element indicating whether $\mathbf{x}$ is a face sample of one particular class. (See Algorithm 3.) Sometimes an output vector could have more than one positive element. Nevertheless, since a face may be detected in multiple nearby sub-windows, such matters can be easily resolved by voting or by measuring the detection confidences (margins).

4. Other Applications

In this section, we look at other cases of face detection.

With In-Plane Rotations. We divide all possible (in-plane) rotated faces into 12 classes according to the angles

Figure 4: Four examples of face classes Γ.

of rotations. As plotted in Figure 4a, they are distinguished by multiples of $\pm 30^\circ$ in-plane rotations. Similar to the case of profile face detection, the number of classes needed for smoothly detecting rotated faces depends on the robustness of the base detector, where we use classifiers derived by MBHboost. Of course, for that matter, the tradeoff between detection rates and efficiency also plays an important role.

Under Various Lighting Conditions. Techniques such as *illumination gradient correction* and *histogram equalization* have been incorporated into detectors, e.g., [8], [9], [15], to reduce the effects of lighting on detection rates. To some degree, correcting illumination gradients simply excludes the impacts on the pattern appearances due to extreme incident angles of light sources, and applying histogram equalizations mainly eliminates the variations caused by different lighting magnitudes. However, the two operations are impractical for a real-time system because their computational costs will dominate a detection process. With integral images, Viola and Jones [17] achieve lighting normalization via calculating the mean and standard deviation of the input pattern. In this way, the normalization cost is significantly reduced, but the effect of extreme incident angles of light sources on the pattern appearance remains unsolved. We instead address the problem of lighting conditions by categorizing the incident angles of a light source into, say, five classes (see Figure 4b), and then construct a multi-class cascade to efficiently account for the issue.

With Partial Occlusions. The appearances and features of occluded faces can be significantly different from faces without occlusions. In [5], the detector is designed to handle eight kinds of occluded faces. And the training data of occluded faces are simply derived from faces without occlusions by excluding features from the predefined occluded regions. The approach is, though effective, difficult to be generalized for detecting faces of other scenarios. With our proposed method, the same eight kinds of occluded faces in [5] can be robustly detected, where, in Figure 4c, the dark regions denote occlusions, and face data within the same class have occlusions at the same location.

With Various Facial Expressions. Unlike the foregoing, this is indeed an easier case since facial expressions only slightly increase the detection difficulty (see Figure 4d). The main purpose we include the example here is to demonstrate the generality of our approach. We shall not discuss this aspect of detections further.

5. Experimental Results

We experiment our system with all the described scenarios of multi-class face detection. In addition, we compare our results with those yielded by a widely used strategy, a view-based system, of which several cascades of detectors are derived by separately training with each corresponding class of data, and a test pattern will be examined by all cascades. Since MBHboost is also applicable for training with one single class of positive and negative data, we have used Algorithm 2 to build the $|\Gamma|$ view-based cascades. Thus, the comparisons between the two approaches will be on a fair ground because the two respective systems are established using the same training and testing data, and the same projection direction set Φ.

The face data are collected from a number of databases, including MIT-CBCL, AR [7], PIE [13], Yale [1], and are created with different poses, orientations, expressions, lighting conditions, and with or without occlusions. We then rotate, crop, and re-scale the face images into the resolution of 24×24 pixels. The initial training set of each class consists of $10,000$ face samples that are properly selected from these candidates and $10,000$ nontrivial non-face samples. For the supplementary set Q, about $20,000$ large-size images that do not contain any faces are gathered for generating non-face training data, through the stage-wise training of Algorithm 2, by the bootstrap technique.

All our testings are run on a P4 3.06 GHz PC, and some of the results are presented in Figure 6. We use the dataset in [12] for profile face detection, and the other one in [10] for rotated face detection. To detect faces under various lighting conditions or with partial occlusions, we respectively collect additional $1,000$ faces (either under various lighting conditions or with occlusions) and $1,000,000$ nontrivial non-face samples as data for validation. The quantitative comparisons reported here emphasize the aspects of accuracy and efficiency. Judging from the ROC curves depicted in Figure 5, the proposed method produces comparable performances to those yielded by the view-based system in accuracy. However, our approach is significantly more efficient, as summarized in Table 1. (Of note, the view-based system is trained with MBHboost.)

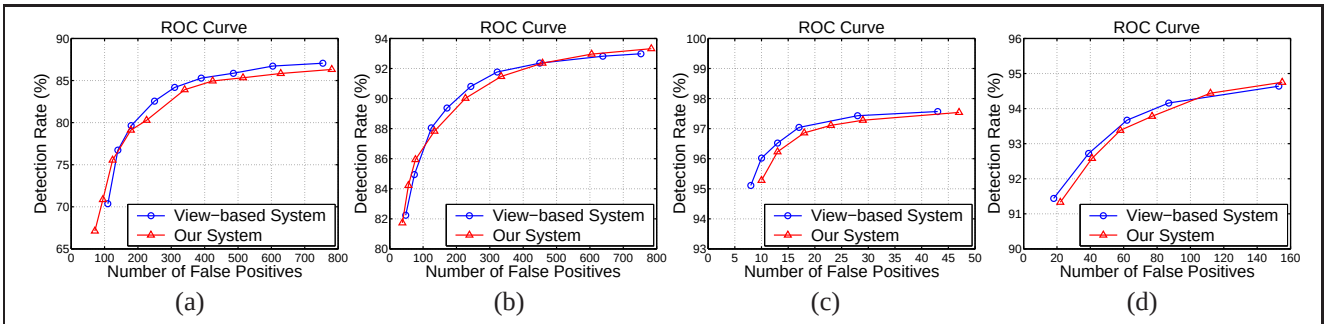


Figure 5: ROC curves. (a) Profile face detection on the dataset in [12]. (b) Rotated face detection on the dataset in [10]. (c) Detecting faces under various lighting conditions. (d) Detecting faces with partial occlusions.

Table 1: Quantitative results in terms of speedup and fps.

Application	Profile	Rotation	Lighting	Occlusion
class #, $\|\Gamma\|$	9	12	5	9
#-times	3.74	4.96	2.96	3.85
(320x240) fps	13.8	8.6	26.1	15.2

#-times speedup by our method over the view-based.

6. Discussion

Through the sharing of projection directions and the use of only one cascade, our face detection algorithm has been shown to have the advantages of generality, efficiency, and accuracy. Compared with other related works, the proposed method outperforms those described in [2], [10]. Though our accuracy for detecting profile faces falls behind that reported in [12], our system achieves real-time performance and is applicable to detect faces of many scenarios.

Acknowledgements. This work is supported in part by NSC grants 93-2213-E-001-010 and 93-2213-E-001-018.

References

[1] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. PAMI*, vol. 23, no. 6, pp. 643–660, 2001.

[2] M. Jones and P. Viola, "Fast Multi-view Face Detection," Tech. Rep. TR2003-96, Mitsubishi Electric Research Laboratories, 2003.

[3] S. Li, L. Zhu, Z. Zhang, A. Blake, H. Zhang, and H. Shum, "Statistical Learning of Multi-View Face Detection," *7th ECCV*, vol. 4, pp. 67–81, 2002.

[4] R. Lienhart and J. Maydt, "An Extended Set of Haar-Like Features for Rapid Object Detection," *ICIP*, vol. 1, pp. 900–903, 2002.

[5] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, "Fast Object Detection with Occlusions," *8th ECCV*, vol. 1, pp. 402–413, 2004.

[6] C. Liu and H. Shum, "Kullback-Leibler Boosting," *CVPR*, vol. 1, pp. 587–594, 2003.

[7] A.M. Martinez and R. Benavente, "The AR Face Database," Tech. Rep., CVC Technical Report #24, 1998.

[8] E. Osuna, R. Freund, and F. Girosi, "Training Support Vector Machines: An Application to Face Detection," *CVPR*, pp. 130–136, 1997.

[9] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. PAMI*, vol. 20, no. 1, pp. 23–38, 1998.

[10] H. Rowley, S. Baluja, and T. Kanade, "Rotation Invariant Neural Network-Based Face Detection," *CVPR*, pp. 38–44, 1998.

[11] R.E. Schapire and Y. Singer, "Improved Boosting Using Confidence-rated Predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[12] H. Schneiderman and T. Kanade, "A Statistical Method for 3D Object Detection Applied to Faces and Cars," *CVPR*, vol. 1, pp. 746–751, 2000.

[13] T. Sim, S. Baker, and M. Bsat, "The CMU PIE Database of Human Faces," Tech. Rep. CMU-RI-TR-01-02, The Robotics Institute, CMU, 2001.

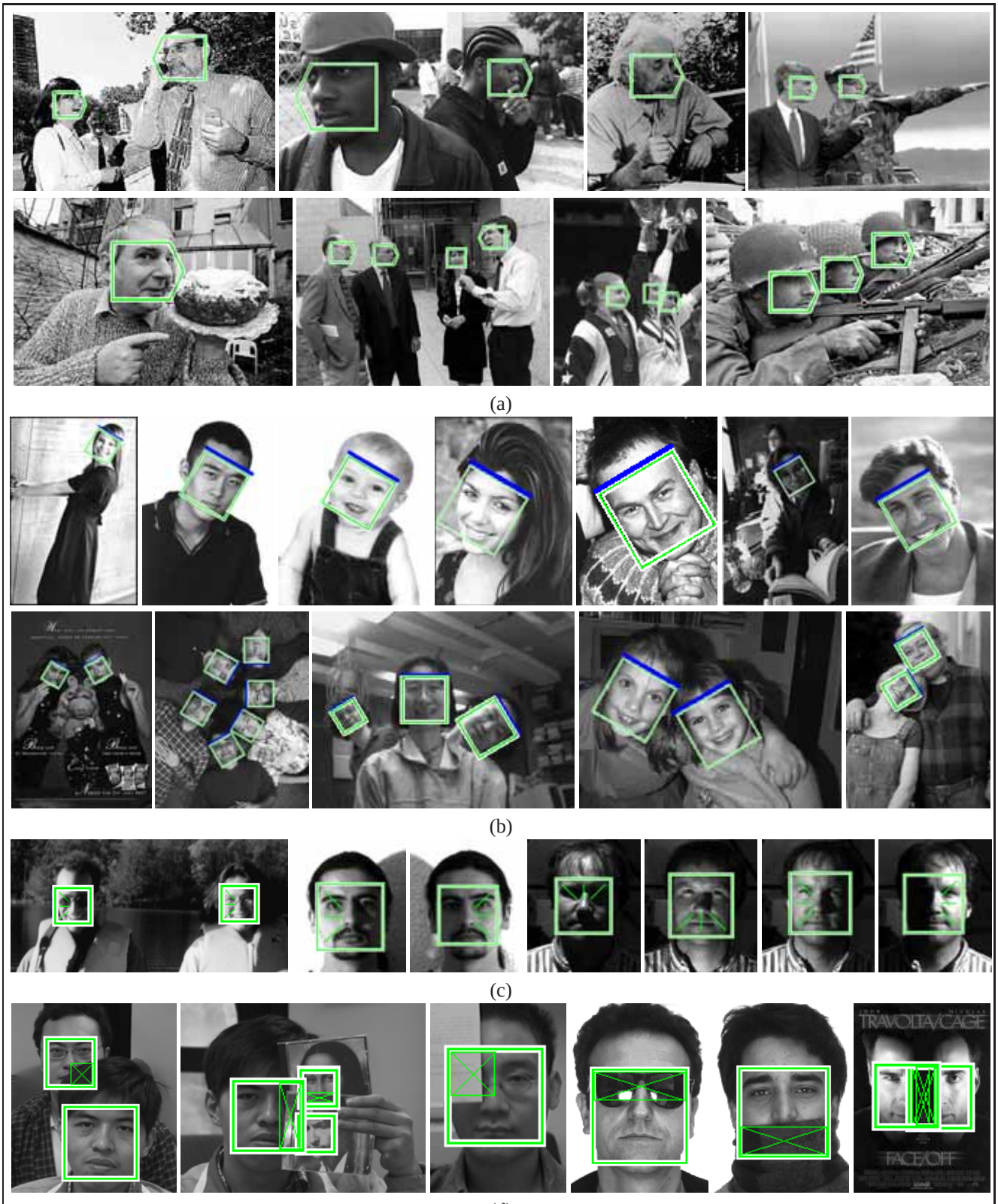
[14] J. Sun, J.M. Rehg, and A. Bobick, "Automatic Cascade Training with Perturbation Bias," *CVPR*, vol. 2, pp. 276–283, 2004.

[15] K.K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. PAMI*, vol. 20, no. 1, pp. 39–51, 1998.

[16] A. Torralba, K.P. Murphy, and W.T. Freeman, "Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection," *CVPR*, vol. 2, pp. 762– 769, 2004.

[17] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *CVPR*, vol. 1, pp. 511–518, 2001.

[18] M.H. Yang, N. Ahuja, and D. Kriegman, "Face Detection Using Mixtures of Linear Subspaces," *FGR*, pp. 70–76, 2000.



(a)

(b)

(c)

(d)

Figure 6: Detection results derived by applying our face detector to some testing images. (a) Profile face detection. (b) Rotated face detection. (c) Detecting faces with various lighting conditions. (d) Detecting faces with partial occlusions.