

On Channel Reliability Measure Training for Multi-Camera Face Recognition

Binglong Xie, Visvanathan Ramesh, Ying Zhu

Terry Boulton

Real-Time Vision and Modeling Dept.
Siemens Corporate Research
Princeton, NJ 08540

Department of Computer Science
University of Colorado
Colorado Springs, CO 80933

Abstract

Single-camera face recognition has severe limitations when the subject is not cooperative, or there are pose changes and different illumination conditions. Face recognition using multiple synchronized cameras is proposed to overcome the limitations. We introduce a reliability measure trained from examples to evaluate the inherent quality of channel recognition. The recognition from the channel predicted to be the most reliable is selected as the final recognition results. In this paper, we enhance Adaboost to improve the component based face detector running in each channel as well as the channel reliability measure training. Effective features are designed to train the channel reliability measure using data from both face detection and recognition. The recognition rate is far better than that of either single channel, and consistently better than common classifier fusion rules.

1. Introduction

Traditionally face recognition was performed on 2D images, often frontal or near-frontal view faces. There are statistical methods including PCA (Principal Component Analysis, or Eigenfaces) [1] and LDA (Linear Discriminant Analysis, or Fisherfaces) [2], Neural Network approaches, EBGM (Elastic Bunch Graph Matching) [3] and so on. In general 2D face recognition methods suffer from pose and illumination changes, because they rely on seen images while the same face can generate novel image instances by varying the pose and lighting conditions. 3D face recognition methods include range-based recognition [4], stereo reconstruction [5], SFS (Shape From Shading) [6], 3D morphable model [7], *etc.* The 3D reconstruction used in these methods is often either intrusive, slow, inaccurate, or requiring manual initialization, and is not appropriate for real-time applications. Currently face recognition still has some severe limitations in typical applications like surveillance and access control, for example, when the subject is not cooperative and turns away from the camera, the accuracy of face recognition can be marred significantly [8].

We propose a face recognition system using two cameras [9]. In each channel, component-based face detector that is trained with a modified AdaBoost algorithm detects faces with pose and illumination changes and LDA-based face recognition is performed to recognize the normalized faces. The recognitions from the two channels are fused to get the final results, using a selection scheme based on a channel reliability measure trained inherent to the individual channel performance.

In this paper, we mainly discuss a modified AdaBoost algorithm and effective weak classifier design for reliability measure training. In selecting a weak classifier for the current strong classifier, the modified AdaBoost tries to lower the empirical training error instead of merely minimizing the error bound as in AdaBoost. Our experiments show that it provides an alternative way to AdaBoost with better performance. We also design effective features from face detection and recognition to train the channel reliability measure using the modified AdaBoost. The recognition rate of this approach is far better than that of either single channel, and consistently better than common classifier fusion rules.

The paper is organized as follows. Section 2 revisits the basic AdaBoost algorithm, as this is referred to throughout the paper. Section 3 briefly introduces the system we use to perform multi-camera face recognition. Section 4 describes the modified AdaBoost algorithm. Section 5 shows how the features and rules are designed to train the reliability measure. In Section 6 we show the experiments and results of the modified AdaBoost, reliability measure training, and system performance. We then conclude in Section 7.

2. AdaBoost

The AdaBoost learning algorithm was first introduced by Freund *et al*[10]. In AdaBoost, the rough base rules are called the *weak classifiers*, and they are combined into the *strong classifier* that is much more accurate than any of the individual weak classifiers.

For two-class AdaBoost, the sample set $\{(x_i, y_i)\}, i = 1, \dots, m$ contains pairs of data x_i and its label y_i , where $x_i \in X$, and $y_i \in Y = \{-1, +1\}$. X is the sample data space,

Given: Samples $\{(x_i, y_i)\}, i = 1, \dots, m$

Initialize: Weights $D_1(i) = \frac{1}{m}$

Iteration: For each $t = 1, \dots, T$:

- Normalize samples weights $D_t(i) \leftarrow \frac{D_t(i)}{\sum_{i=1}^m D_t(i)}$
- Retrain weak classifiers $\{h_j(x)\}$ with D_t
- Choose a h_t having least weighted error $\epsilon_t = \sum_{\{i; y_i \neq h_t(x_i)\}} D_t(i)$
- Compute $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
- Update sample weights $D_{t+1}(i) = D_t \exp(-\alpha_t y_i h_t(x_i))$

Output: The strong classifier:

$$h(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

Figure 1: Basic AdaBoost algorithm

and Y is the label space. The samples with label $+1$ are called positive samples, or object samples; and the samples with label -1 are called negative samples, or non-object samples. There exist some weak classifiers, $\{h_j(x) : X \rightarrow Y\}$. AdaBoost finds the strong classifier $y = h(x) : X \rightarrow Y$ using the process described in Figure 1.

In each step, AdaBoost selects a weak classifier that has the least weighted error ϵ_t , which is a summation of weights of samples that are incorrectly classified. The samples are then reweighted according to the weak classifier’s classification correctness. The weight updating scheme makes the next weak classifier focus more on the incorrectly classified samples. Step by step, the strong classifier is ‘boosted’ and performs better and better.

The iterative process terminates when certain conditions are met, one of them uses empirical training error, or in short, training error. The training error of a strong classifier $h(x)$ is defined as $e_e = \sum_{\{i; y_i \neq h(x_i)\}} \frac{1}{m}$. For example, when the training error of the current strong classifier is small enough (*e.g.*, 0) the training is over.

3. System Overview

The multi-camera face recognition system is depicted in Figure 2. The face is captured by two cameras, and each channel has independent face detector and recognizer. The channel reliability measure, f , is estimated for each channel, and the more reliable channel is selected as the final recognition, subject to a threshold on reliability.

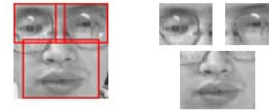


Figure 3: Face components, on the template face (Left) and individual (Right).

3.1. Component-Based AdaBoost Face Detection

We use a component-based face detector, which is robust with face pose and illumination variations, because the small components are less subject to those variations and the fusion of the components allows moderate geometric deformations of the configuration of the components [11]. Our facial components include left eye, right eye and the lower face component covering mouth and part of nose, see Figure 3. The component detectors are trained with the modified AdaBoost algorithm (See Section 4) using Haar wavelet features that can be evaluated very quickly with an integral image [12]. When the components are detected, we apply statistical fusion of individual component detectors with a probabilistic geometric face model [13]. This gives us robust face detection and accurate face localization, which is very important for face recognition.

3.2. LDA-Based Face Recognition

The detected face is recognized with an LDA recognizer using the nearest neighbor method in the LDA subspace. The matches are sorted by the distance, the top match with the smallest distance.

3.3. Fusion by Selection from Trained Reliability Measure

When multiple face recognizers yield individual recognitions, fusion can be performed to improve the performance. The common combining rules: minimal geometric mean, minimal arithmetic mean, minimal median, minimal minimum, minimal maximum and majority voting are summarized in [14] by assuming statistical independence and so on [15]. These common fusion rules are however ‘rigid’, meaning that they cannot take advantage of the available examples for better fusion.

We perform fusion using our learning techniques. The training examples are processed by channels and we collect the data x from face detection and recognition, and label it as positive $y = +1$ if the top match corresponds to the example’s true identity, or $y = -1$ otherwise. So we put the problem into the AdaBoost setting. Friedman [16] proved that in an additive logistic regression model, when the AdaBoost error bound is minimized by choosing appropriate

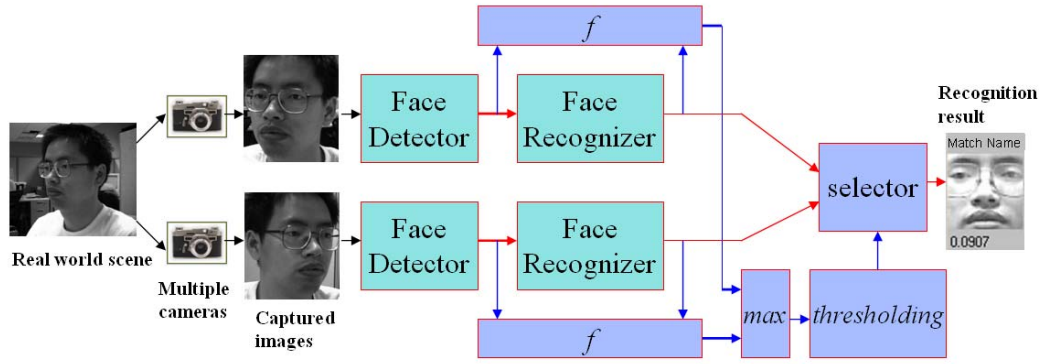


Figure 2: Reliability based selection of multiple channel face recognition.

$f(x)$ in boosting, the channel reliability $P(y = +1|x)$ is a monotone function of the strong classifier response $f(x)$:

$$P(y = +1|x) = \frac{e^{f(x)}}{e^{f(x)} + e^{-f(x)}} = \frac{e^{2f(x)}}{e^{2f(x)} + 1}, \quad (1)$$

where $f(x)$ is ‘unthresholded’ strong classifier response:

$$f(x) = \sum_{t=1}^T \alpha_t h_t(x) \quad (2)$$

Therefore, we can train the function $f(x)$ to represent the channel reliability equivalently using AdaBoost. If we can find a good representation of channel sample x and train an appropriate strong classifier response function $f(x)$, we can robustly estimate the channel reliability.

4. Modified AdaBoost

AdaBoost does not improve the empirical training error directly, instead, it improves the error bound of the strong classifier [10]. We are more interested in the empirical training error, *i.e.*, the performance of the strong classifier on the training examples, than the error bound, because the bound is too loose [17]. However, the prediction based on weighted error in basic AdaBoost is not always aligned with the empirical training error. We use a modified version of AdaBoost as shown in Figure 4.

The difference from the basic AdaBoost is that in each step t , it chooses n weak classifiers instead of one weak classifier as temporary candidates. It then checks if the first temporary candidate weak classifier in the chosen list improves the empirical training error. If it does, the weak classifier is accepted; otherwise it looks at the next temporary candidate. This tries to get a strong classifier that is verified to be empirically better than that of the previous boosting step. Note that it does not try to look for the weak classifier that improves the empirical training error most. It finds the

Given: Samples $\{(x_i, y_i)\}, i = 1, \dots, m$

Initialize: Weights $D_1(i) = \frac{1}{m}$

Iteration: For each $t = 1, \dots, T$:

- Normalize samples weights $D_t(i) \leftarrow \frac{D_t(i)}{\sum_{i=1}^m D_t(i)}$
- Retrain all weak classifiers $\{h_j(x)\}$ with D_t
- Choose top n weak classifiers with least weighted errors $\epsilon_j = \sum_{\{i; y_i \neq h_j(x_i)\}} D_t(i)$, denote as pairs $\{h_k^t, \epsilon_k^t\}_{k=1, \dots, n}$.
For each $k = 1, \dots, n$:
 - Let $h_t = h_k^t, \epsilon_t = \epsilon_k^t$
 - Compute temporary $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$
 - Get a temporary strong classifier $h^t(x) = \text{sign} \left(\sum_{l=1}^t \alpha_l h_l(x) \right)$
 - If $h^t(x)$ lowers training error e_e , add $h_t(x)$ to strong classifier.
 - Otherwise, move to next k .
- Update sample weights $D_{t+1}(i) = D_t \exp(-\alpha_t y_i h_t(x_i))$

Output: The strong classifier:

$$h(x) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right)$$

Figure 4: Modified AdaBoost algorithm

weak classifier with least weighted error *and* improving the empirical error. This is based on the observation that if only the empirical training error is targeted, the generalization power of AdaBoost is hurt, and the boosting process can be trapped in a local minimum faster, in turn limits itself from improving the empirical training error more. If none of the n temporary candidate weak classifiers improves the empir-

ical error, it simply falls back to the basic AdaBoost and uses the first candidate.

The choice of n is a tradeoff among generalization, empirical error oriented training, and training time. If n is 1, the modified AdaBoost falls back to basic AdaBoost. If n is large, it can potentially improve the empirical error more at the current step, but it may again hurt the performance in future training iterations, and training is slower.

5. Weak Classifier Design for Reliability Measure

5.1. Example Data Space

The example data space X should capture the variation of the channel data. Even though the image is the source of the whole channel processing, the information in it is not directly useful for reliability estimation. Instead, we collect the output of the face detector and output of the face recognizer and form a compound vector \mathbf{x} as our example variable. This makes sense, because the reliability of the channel depends on how good both face detection and recognition are. This is in contrast to common fusion rules as they only try to fuse the face recognitions.

Our example data consists of three basic data pieces:

Face detection geometric data \mathbf{x}_{fdgeo} The direct output of face detector includes the detected individual component locations, sizes, confidences and overall face detection confidence.

Face detection Haar wavelets \mathbf{x}_{fdrs} In the individually trained face component detectors, there are weak classifiers using Haar wavelets. The responses of these weak classifiers not only serve the component detectors, but also provide valuable information about the component and face, which can be used for channel reliability training.

Face recognition matching distances \mathbf{x}_{fr} The output of face recognizer includes a list of matching distances (or scores), one for each class.

The example data is denoted as $\mathbf{x} = [\mathbf{x}_{fdgeo}^T, \mathbf{x}_{fdrs}^T, \mathbf{x}_{fr}^T]^T$.

5.2. Feature Design

In any boosting, the key to get an effective strong classifier is the design of weak classifiers matching the nature of specific classification problem. A feature, also called a filter, is a function $g(\mathbf{x})$ mapping the input \mathbf{x} into a feature response space Z , in our case, $Z = R = (-\infty, +\infty)$. A weak classifier then imposes another mapping from Z to the label space Y . We use both binary and trisegmental thresholding to derive weak classifiers from a feature.

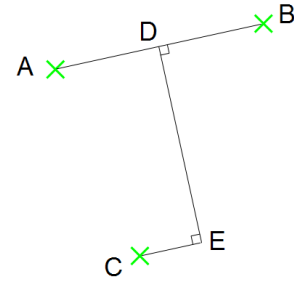


Figure 5: Face geometric configuration. A , B and C are the positions of left eye, right eye and lower face components of the detected face, respectively. D is the middle point of A and B . Skewness is defined as $\frac{CE}{AB}$. Tilting is defined as $\frac{DE}{AB}$.

We employ five categories of features for reliability measure training for a two-channel setting:

Face detection geometric ($fdgeo$) features These features are defined on \mathbf{x}_{fdgeo} , including literal overall face confidence, component confidences, face scale, face skewness and tilting (see Figure 5) and so on.

Face detection Haar wavelets ($fdrs$) features Currently the left eye, right eye and lower face detectors on a 64×64 pixel face use 98, 79, and 99 Haar wavelet features respectively. These features from \mathbf{x}_{fdrs} are directly used as reliability measure features.

Face recognition matching distance (fr) features There are 9 types of features defined on the 10 best matching distances output \mathbf{x}_{fr} by the face recognizer, including: literal distances; distance differences; normalized distances to top match; differences of mean distance to top match, either plain or normalized; standard deviations of distance, either plain, or normalized by mean, or normalized by difference of mean to top match; and slope changes of distance to top match.

Joint channel ($joint$) features The information from the other channel also could help to determine if this channel is reliable because there is correlation between two channels in fact. The joint features are the differences of the single-channel features of the two channels.

Consecutive time ($conti$) features Assuming the correlation along time of the face appearing in any channel, the features are also correlated over time. The stability of the features is used to design consecutive time features. These features are the differences of the single-channel features of the two consecutive frames.

6. Experiments and Results

6.1. Modified AdaBoost Training

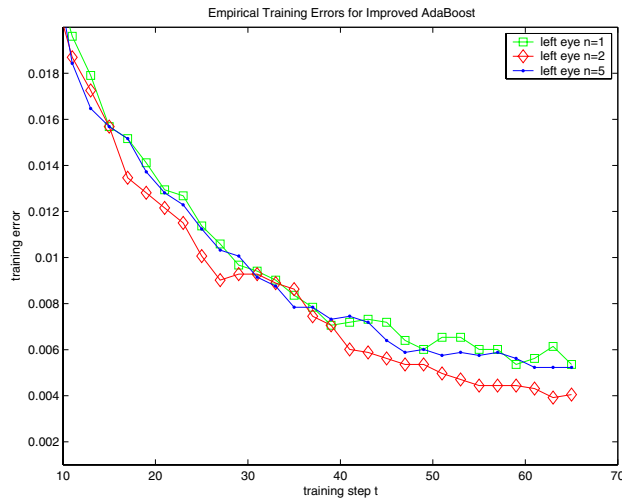


Figure 6: Modified AdaBoost training for the left eye.

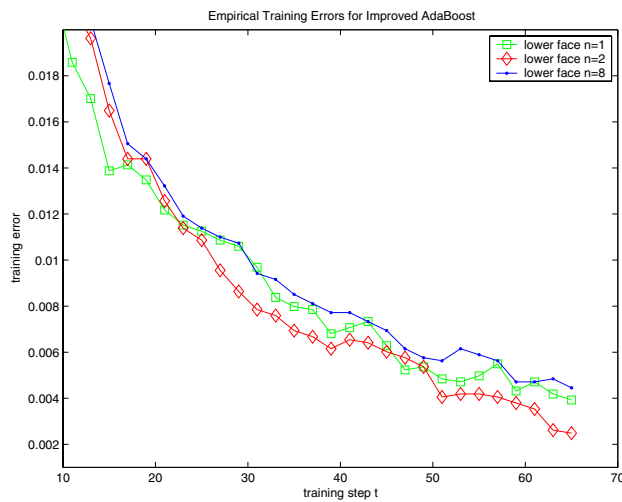


Figure 7: Modified AdaBoost training for the lower face.

Two examples of training errors for modified AdaBoost with 1862 positive and 6231 negative examples are shown in Figures 6 and 7. Notice $n = 1$ corresponds to basic AdaBoost. Our examples show that in general modified AdaBoost performs better for an appropriate n , $n = 2$ in the examples. If n is too large, for example, $n = 5$, the results are not consistently better than those of basic AdaBoost. In practice we try with different n and take the best strong classifier.

6.2. Reliability Measure Training

We collect 33 synchronous videos from 33 subjects, each has about 683 frames, with various poses. Around 481 frames/subject are used for reliability training. The training errors are shown in Figure 8. It is obvious that adding more features results in a better trained reliability measure.

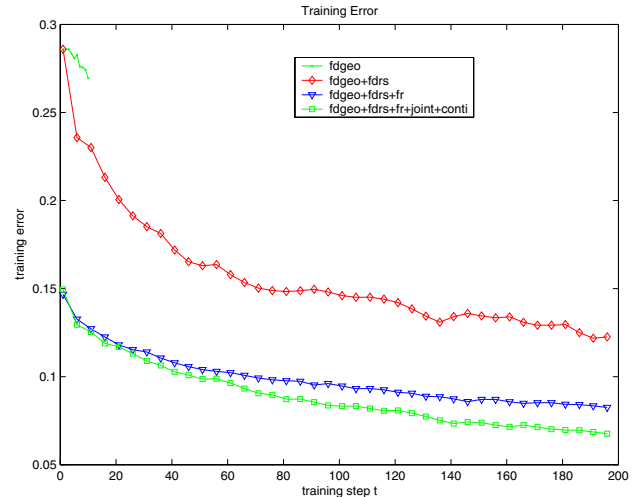


Figure 8: The channel reliability training errors with different feature sets. Training of *fdgeo* features-only strong classifier stops early because it has only 12 weak classifiers.

6.3. System Performance

After the reliability measure is trained, we use the testing frames (about 202 frames for each subject) to test the system performance. The detection rate is defined as the number of frames with the selected channel's reliability over a threshold divided by the total frame number. The recognition rate is defined as the number of detected and correctly recognized frames divided by the total frame number. The performance of the system using the modified AdaBoost and reliability measure-base selection is compared to the common fusion rules in Figures 9 and 10. In Figure 9, generally fusions perform better than any of the single channels, channel 0 or channel 1, except that the minimal maximum rule is worse than channel 0 but better than channel 1. The perfect selection is the selection by human at the maximal detection rate, which is the upper limit of recognition rate using selection. Especially, our fusion by trained reliability is the closest to the perfect selection as shown in partial performance curves magnified in Figure 10. We use leave-one-out strategy to sample the 202 testing frames for each subject, and estimate the confidence of the recognition rate. As shown in Figure 10, our fusion by selection outperforms the best common fusion rule, the minimal minimum,

with high confidence. The curves are well separated with $\pm 3\sigma$, which corresponds to confidences larger than 99.7%.

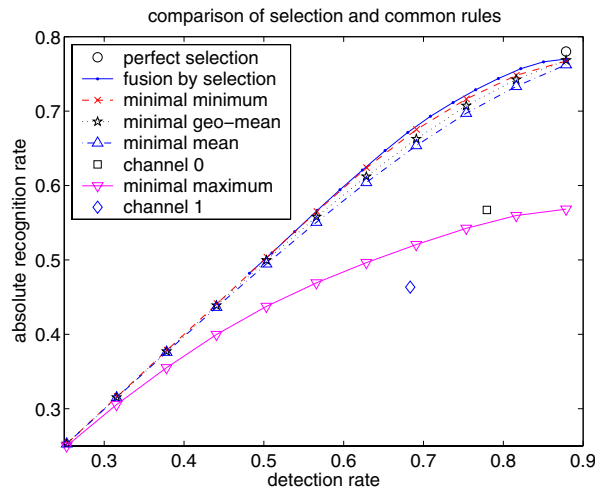


Figure 9: Performance of different fusions.

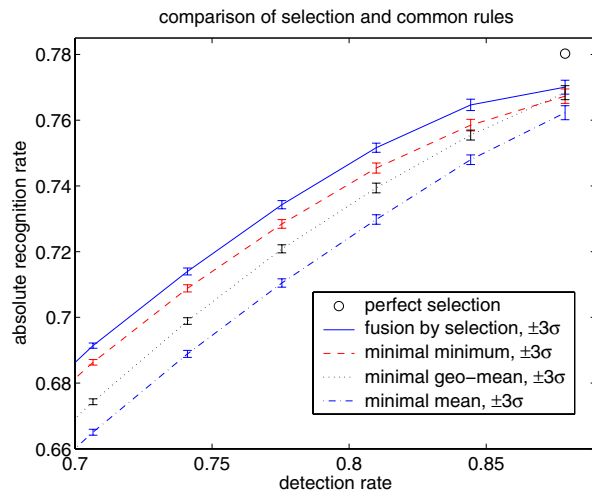


Figure 10: Performance of different fusions (blown-up).

7. Conclusion

We introduce a modified AdaBoost algorithm and effective weak classifier design from face detection and recognition for reliability measure training, in a multi-camera face recognition system with fusion by reliability-based selection. The modified AdaBoost improves boosting to a lower training error with same number of weak classifiers. The recognition rate of this approach is far better than that of either single channel, and consistently better than common classifier fusion rules.

References

- [1] M. Turk and A.P. Pentland, "Eigenfaces for recognition," *CogNeuro*, vol. 3, no. 1, pp. 71–96, 1991.
- [2] Peter N. Belhumeur, Joao Hespanha, and David J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," in *ECCV (1)*, 1996, pp. 45–58.
- [3] L. Wiskott, J.M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *PAMI*, vol. 19, no. 7, pp. 775–779, July 1997.
- [4] C.S. Chua, F. Han, and Y.K. Ho, "3d human face recognition using point signature," in *AFGR00*, 2000, pp. 233–238.
- [5] Shihong Lao, Masato Kawade, Yasushi Sumi, and Fumiaki Tomita, "Building 3d facial models and detecting face pose in 3d space," in *Proc. 2nd Int'l Conf. on 3-D Digital Imaging and Modeling, 3DIM99*, 1999, pp. 398–404.
- [6] W. Zhao, "3d model enhanced face recognition," in *ICIP00*, 2000, p. WA02.05.
- [7] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3d morphable model," in *AFGR02*, 2002, pp. 192–197.
- [8] P.J. Phillips, P. Grother, R.J. Micheals, D.M. Blackburn, E. Tabassi, and J.M. Bone, "Frvt 2002: Overview and summary," <http://www.frvt.org/FRVT2002/documents.htm>, March 2003.
- [9] Binglong Xie, Terry Boulton, Visvanathan Ramesh, and Ying Zhu, "Multi-camera face recognition by reliability-based selection," in *IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety*, Alexandria, VA, USA, October 2006.
- [10] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Computational Learning Theory: Second European Conference (EuroCOLT'95)*, P. Vitanyi, Ed., pp. 23–37. Springer, Berlin, 1995.
- [11] B. Heisele, T. Serre, M. Pontil, and T. Poggio, "Component-based face detection," in *CVPR01*, 2001, pp. I:657–662.
- [12] P. Viola and M. Jones, "Robust real-time face detection," in *ICCV01*, 2001, p. II: 747.
- [13] Binglong Xie, Dorin Comaniciu, Visvanathan Ramesh, Terry Boulton, and Markus Simon, "Component fusion for face detection in the presence of heteroscedastic noise," in *Annual Conf. of the German Society for Pattern Recognition (DAGM'03)*, Magdeburg, Germany, 2003.
- [14] Shaohua Zhou, Rama Chellappa, and Wenyi Zhao, *Unconstrained Face Recognition*, Springer, 2006.
- [15] J. Kittler, M. Hatef, R.P.W. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.
- [16] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," 1998.
- [17] R. Schapire, "The boosting approach to machine learning: An overview," 2001.