Joint Haar-like Features for Face Detection

Takeshi Mita Toshimitsu Kaneko Osamu Hori
Multimedia Laboratory, Corporate Research & Development Center, Toshiba Corporation
1 Komukai Toshiba-cho, Saiwai-ku, Kawasaki-shi, Kanagawa 212-8582, Japan
E-mail: takeshi.mita@toshiba.co.jp

Abstract

In this paper, we propose a new distinctive feature, called joint Haar-like feature, for detecting faces in images. This is based on co-occurrence of multiple Haar-like features. Feature co-occurrence, which captures the structural similarities within the face class, makes it possible to construct an effective classifier. The joint Haar-like feature can be calculated very fast and has robustness against addition of noise and change in illumination. A face detector is learned by stagewise selection of the joint Haar-like features using AdaBoost. A small number of distinctive features achieve both computational efficiency and accuracy. Experimental results with 5,676 face images and 30,000 nonface images show that our detector yields higher classification performance than Viola and Jones' detector, which uses a single feature for each weak classifier. Given the same number of features, our method reduces the error by 37%. Our detector is 2.6 times as fast as Viola and Jones' detector to achieve the same performance.

1. Introduction

Detecting faces in images is a fundamental task for realizing surveillance systems or intelligent vision-based human computer interaction [1]. To build flexible systems that work in a variety of lighting conditions and run on mobile phones or handheld PCs, robust and efficient face detection algorithms are required.

Appearance-based methods are mainly employed to achieve high detection accuracy. They solve a two-class problem by using a probabilistic framework or finding a discriminant function from a large set of training examples. For example, neural network-based methods, support vector machines and other kernel methods have been proposed [2, 3, 4, 5]. Most of these algorithms use raw pixel values as features. However, they are sensitive to addition of noise and change in illumination. Instead, Papageorgiou et al. [6] used Haar-like features, which are similar to Haar basis functions. The features encode differences in average intensities between two rectangular regions, and they can extract texture without depending on absolute intensities. Recently, Viola and Jones proposed an efficient scheme for calculating these features [7]. They also proposed a method for constructing a strong classifier by selecting a small number of distinctive features using AdaBoost. This framework provides both robustness and computational efficiency.

Many improvements or extensions of this method have been proposed. We will introduce two major approaches and then point out problems of these approaches. The first approach is an improvement of the boosting algorithm. There are modified versions of AdaBoost such as RealBoost [8], KLBoosting [9] and FloatBoost [10]. The second approach is using extensions of the feature sets such that various image patterns can be evaluated. As shown in Figure 1, in addition to the basic feature set (a), different arrangements or numbers of rectangles such as (b), (c) or (d) are used in [7, 11]. Lienhart et al. [12] introduced an efficient scheme for calculating 45° rotated features. Although both approaches are effective, they are insufficient to achieve more accurate face detection. Most of these methods construct a weak classifier by selecting one feature from the given feature set. However, the generalization performance is no longer improved in later rounds of the boosting process because the classification task using only one feature becomes more difficult. Viola and Jones reported that features which were selected in later rounds yielded error rates between 0.4 and 0.5 while features selected in early rounds had error rates between 0.1 and 0.3 [7]. Such *'too weak'* classifiers do not contribute to improving the generalization performance while they can reduce the training error. Any sophisticated boosting algorithm will face this problem. Taking the simplest approach, which increases the number of rectangles, is not feasible because it needs too much time for learning. It is also difficult to create new distinctive feature sets manually.

To solve this problem, it is necessary to find more distinctive features, which can capture the structural similarities within the face class. In this paper, we propose a new

Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)
1550-5499/05 $20.00 © 2005 **IEEE**

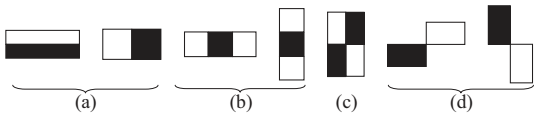

Figure 1. Examples of Haar-like feature sets.

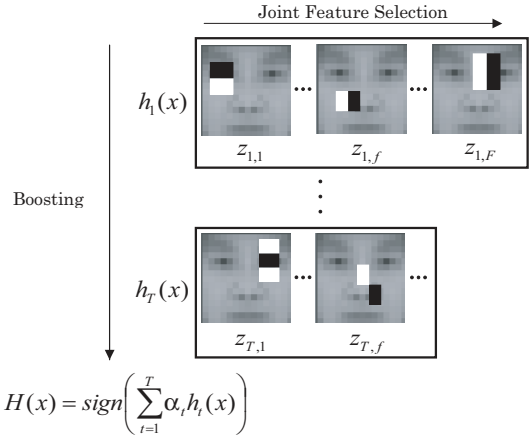

Figure 2. Overview of our method.

feature, called joint Haar-like feature, for detecting faces in images. This is based on co-occurrence of multiple Haar-like features. Feature co-occurrence, which captures the characteristics of human faces, makes it possible to construct a more powerful classifier. The joint Haar-like feature can be calculated very fast independently of image resolution and has robustness against addition of noise and change in illumination. A face detector is learned by stagewise selection of weak classifiers based on the joint Haar-like features using AdaBoost. Figure 2 shows an overview of our method. The final strong classifier $H(x)$ is a linear combination of weak classifiers $h_1(x)$ to $h_T(x)$. In contrast to the method by Viola and Jones, each weak classifier consists of multiple features. For example, $h_1(x)$ observes F features in total and evaluates joint statistics of these features. The structural similarities of faces, which cannot be evaluated using a single feature, are extracted from $z_{1,1}$ (eye regions are darker than neighboring regions), $z_{1,f}$ (nostrils are dark) and $z_{1,F}$ (the region between the eyes is brighter than the eyes). These combined features are selected in each round of the boosting process, such that the error on the training set is minimized. The combination of spatially separated features as shown in Figure 2 cannot be found by the extended feature set shown in Figure 1.

We first describe the joint Haar-like features. Then, we show an algorithm to construct a strong classifier using AdaBoost. Experimental results with 35,676 images show that our method yields higher classification performance than Viola and Jones' method.

2. Features

2.1. Haar-like Features

Haar-like features have scalar values that represent differences in average intensities between two rectangular regions. They capture the intensity gradient at different locations, spatial frequencies and directions by changing the position, size, shape and arrangement of rectangular regions exhaustively according to the base resolution of the detector. For example, when the resolution is 19×19 pixels, 80,160 features are generated from the feature sets (a) to (d) in Figure 1. For each feature, a one dimensional probability density is calculated from all training examples. Figure 3 shows a density calculated from $z_{1,1}$. The two curves indicate the densities from the face and nonface class. In [7], a weak learning algorithm is designed to select the single feature that best separates the face and nonface examples. A small number of effective features are selected by updating the sample distribution using AdaBoost. However, as mentioned above, the error rate of the weak classifiers selected in later rounds of the boosting process becomes large because the updated sample distribution consists of many difficult face and nonface examples, which are similar to each other. Even the best feature selected from 80,160 features cannot provide good classification performance. Figure 4 shows the performance of Viola and Jones' detector. The training error and the generalization error are plotted against the number of weak classifiers. The training error converges to zero when the number of features reaches about 500. However, the generalization error is no longer reduced after 1,000 features are selected. This means that no effective features remain and further improvement cannot be expected. Wu et al. [11] divided the range of the feature values into 64 partitions to express complex densities. However, the above problem still remains.

2.2. Feature Value Binarization

To improve the generalization performance, we use weak classifiers that observe multiple features. Feature co-occurrence makes it possible to classify difficult examples that are misclassified by weak classifiers using a single feature. We represent the statistics of feature co-occurrence using their joint probability. To calculate the joint probability, we quantize the feature value z to two levels. By doing so, each feature value is represented by a binary variable s, which is 1 or 0, specifying face or nonface respectively. The

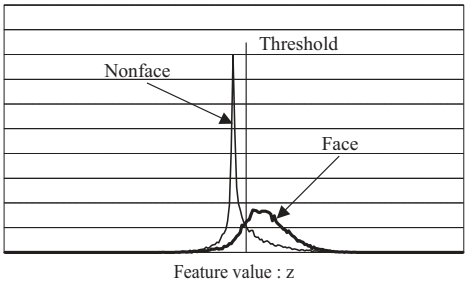

Figure 3. An example of a probability density.

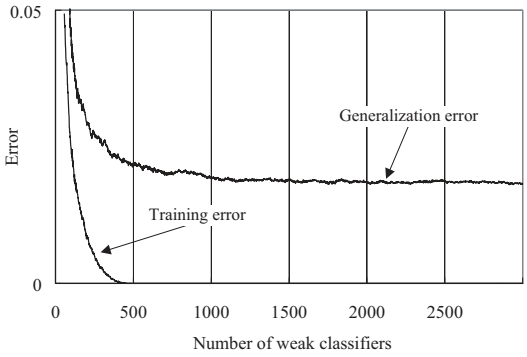

Figure 4. Performance of Viola and Jones' classifier.

variable s for an example x is calculated by

$$s(x) = \begin{cases} 1 & \text{if } p \cdot z(x) > p \cdot \theta \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

where θ is a threshold and p is a parity indicating the direction of the inequality sign. The values of θ and p are determined so that the error rate is minimized. This binarization rule is the same as Viola and Jones' weak classifier. In order to confirm only the effectiveness of exploiting feature co-occurrence, we do not use any operations different from Viola and Jones' method except for combining multiple features. The joint Haar-like features are not limited to the case of using binarized feature values. Multi-level quantization of the feature value fits more complex distributions than binarization. However, we do not focus on how many levels are appropriate in this paper.

2.3. The Joint Haar-like Features

The joint Haar-like features are represented by combining the binary variables computed from multiple features.

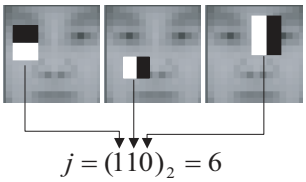

Figure 5. An example of the joint Haar-like feature.

Figure 5 shows an example of the joint Haar-like feature, which is based on the co-occurrence of three Haar-like features. When the variables are 1, 1 and 0, the value of the joint Haar-like feature is calculated by

$$j = (110)_2 = 6. \quad (2)$$

The feature value j as a binary number specifies an index for 2^F different combinations, where F is the number of combined features. The feature represents the feature co-occurrence between different positions, resolutions and orientations.

For each class, statistical dependencies between the features are obtained by observing j for each example. We use such dependencies for classification. The subwindow is classified to be face or nonface by evaluating from which class the feature value is likely to be observed. The combined features are selected to capture distinctive structural similarities of faces. In the subsequent section, we will show the algorithm for selecting the joint Haar-like features.

3. Learning Classifiers

This section describes an algorithm for learning a face detector. First, we give a learning procedure based on AdaBoost. Then, we describe the weak classifiers based on the joint Haar-like features and how distinctive feature combinations are found.

3.1. Selecting Joint Haar-like Features

The learning algorithm is shown in Figure 6. A set of N labeled training examples is given as $(x_1, y_1), \ldots, (x_N, y_N)$, where $y_i \in \{+1, -1\}$ is the class label associated with example x_i. $D_t(i)$ is a weight of example x_i. The weights are initialized by $D_1(i) = 1/N$. The final strong classifier $H(x)$ is a linear combination of T weak classifiers $h_t(x)$:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right). \quad (3)$$

1. Given example images $(x_i, y_i), \cdots, (x_N, y_N)$, $y_i \in \{+1, -1\}$ for face and nonface examples respectively.
2. Initialize weights $D_t(i) = \frac{1}{N}$.
3. For $t = 1, ..., T$:
   (A) For each feature, calculate a feature value.
   (B) Binarize each feature value and assign a binary variable according to Eq.(1).
   (C) Train a weak classifier based on a combination of features. The error is evaluated with respect to $D_t(i)$, $\epsilon_t = \sum_{i:y_i \neq h_t(x_i)} D_t(i)$.
   (D) Choose $h_t(x)$ with the lowest error ϵ_t.
   (E) Update the weights:
   $$D_{t+1}(i) = \frac{D_t(i) \exp\left(-\alpha_t y_i h_t(x_i)\right)}{\sum_i D_t(i) \exp\left(-\alpha_t y_i h_t(x_i)\right)},$$
   where $\alpha_t = \frac{1}{2} \log\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
4. The final strong classifier is:
   $$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Figure 6. Learning procedure based on AdaBoost.

In each round of the boosting process, the best joint Haar-like feature is selected according to the step (A) to (E) in Figure 6.

3.2 Weak Classifiers Based on the Joint Haar-like Features

A weak classifier based on Bayes decision rule is expressed as follows:

$$\forall x \in j, \quad h_t(x) = \begin{cases} +1 & \text{if } P(y=+1|j) > P(y=-1|j) \\ -1 & \text{otherwise} \end{cases}, \tag{4}$$

where j is a feature value of the joint Haar-like feature. $P(y=+1|j)$ and $P(y=-1|j)$ are joint probabilities observing feature co-occurrence represented by j. They are evaluated with respect to weights $D_t(i)$ of examples as follows:

$$P(y=+1|j) = \sum_{i:x_i \in j \wedge y_i = +1} D_t(i), \tag{5}$$

$$P(y=-1|j) = \sum_{i:x_i \in j \wedge y_i = -1} D_t(i). \tag{6}$$

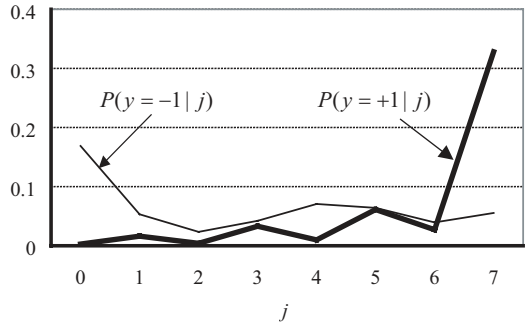

Figure 7. A weak classifier based on class-conditional joint probabilities.

An example of the joint probabilities $P(y=+1|j)$ and $P(y=-1|j)$ measured from three features is shown in Figure 7. The three features yield eight combinations of binary variables, which are from $(000)_2$ to $(111)_2$. If $j = (111)_2 = 7$, the input pattern is classified to be a face.

3.3. Searching for Distinctive Feature Co-occurrence

To construct a weak classifier, we need to find distinctive feature co-occurrences. The best feature combination can be found by exhaustive search from all possible feature combinations. However, it is not feasible for a limited training time. The computational complexity for selecting F from M features is $O(M^F)$. Several solutions for efficient feature selection have been proposed, but without the guarantee of optimal selection [13]. We use the well-known Sequential Forward Selection. Features are added one by one to improve the classification performance. The computational complexity becomes $O(M \cdot F)$.

How we determine F is also important. Choosing F too large leads to overfitting. Furthermore, the range of j becomes twice as large by adding one feature. To avoid statistical unreliability due to long histograms, we limit F by,

$$2^{F^{max}} \times 10 < N, \tag{7}$$

so that at least 10 examples are assigned to each bin when each j is uniformly observed.

The following two methods for determining F are considered:
(1) Select the best classifier from multiple classifiers trained using different F. Since a fixed F is used for each classifier, all weak classifiers observe the same number of features.
(2) Choose the best F in each boosting round using the hold-out method. Each weak classifier tries to find the best number of combined features, so that the validation error

$\epsilon_{T'}$ is minimized. A set of N' labeled validation examples (x'_i, y'_i) is used.

$$F_{opt} = \arg\min_{F} \epsilon_{T'}, \tag{8}$$

where

$$\epsilon_{T'} = \frac{1}{N'} \sum_{i=1}^{N'} I\left(H_{T'}(x'_i) \neq y'_i\right), \tag{9}$$

and $H_{T'}(x)$ is a strong classifier at $t = T'$:

$$H_{T'}(x) = \sum_{t=1}^{T'} \alpha_t h_t(x). \tag{10}$$

Schneiderman and Kanade [14] learnt the appearance of faces and nonfaces by evaluating dependencies between wavelet coefficients. The best performance has been reported for their detector. However, it does not run in real-time since a large set of feature combinations given in advance is used for evaluating joint statistics. Our algorithm automatically selects a small number of distincitve feature combinations.

4. Experimental Results

This section describes the following three experiments:
(1) Comparison in performance between our classifier and Viola and Jones' one. Viola and Jones' learning algorithm is implemented by ourselves.
(2) Reliability evaluation by cross-validation (exchanging face examples of the training data with those of the test data).
(3) Comparison in performance between classifiers using different F (the number of combined features).

4.1. Learning Classifiers

A set of 13,000 frontal face images is collected from various sources, including well-known face databases, Yale [15] and XM2VTS [16]. All faces are scaled and aligned to a base resolution of 19×19 pixels. Another set of 10,000 nonface images is collected by cropping images containing no faces. Using this data, we trained Viola and Jones' classifier. We added 3,000 nonface images misclassified by this classifier and collected 13,000 in total. Then, we trained all classifiers compared in the subsequent sections using 26,000 examples. Figure 8 shows typical examples of the training data. The boosting iterations are stopped when the total number of selected features reaches 1,000. The number of weak classifiers of Viola and Jones' classifier becomes 1,000 ($T = 1,000$). Our classifier consists of 200 weak classifiers ($T = 200$), when the number of combined features is fixed as $F = 5$. The computational cost of both classifiers are equal when the same number of features are used.

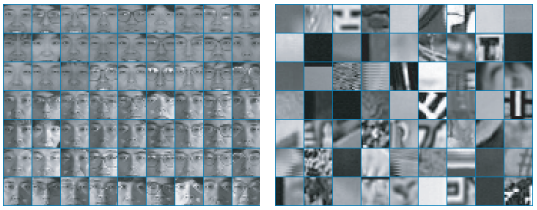

Figure 8. Some examples in the training data. left: face-pattern, right: nonface-pattern.

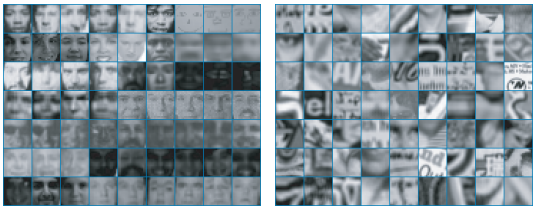

Figure 9. Some examples in the test data. left: face-pattern, right: nonface-pattern.

4.2. Test Data

The test data consists of 5,676 face images and 30,000 nonface images. The face images are collected from the MIT+CMU test set [3] and our own database, and they are cropped in the same manner as the training data. The nonface images are collected from the MIT+CMU test set by cropping sub-regions containing no faces. Figure 9 shows typical examples of the test data. We can compare only classification performance by using the test images whose sizes are the same as the training images. In practice, the classifier is scanned across the image at multiple scales and locations for detecting different size faces. However, there is no uniform criterion to determine that a face is correctly detected, i.e., the square window should contain the eyes and also the mouth. This evaluation method is appropriate for comparison in classification performance rather than the face detection systems.

4.3. Results (1): Performance Comparison

The first experiment compares Viola and Jones' classifier with our classifier. The former is restricted to using a single feature for each weak classifier. The latter combines three features, i.e. $F = 3$. Four classifiers shown in Table 1 are compared in this experiment. They are different from the feature set and the number of combined features F. C1 and C3 use feature set (a) in Figure 1. On the other hand, C2 and C4 use extended feature set (a), (b) and (c). C1

Table 1. Compared four classifiers.

	Feature set	F	T	# of features
C1	Fig.1 (a)	1	1,000	1,000
C2	Fig.1 (a)(b)(c)	1	1,000	1,000
C3	Fig.1 (a)	3	333	999
C4	Fig.1 (a)(b)(c)	3	333	999

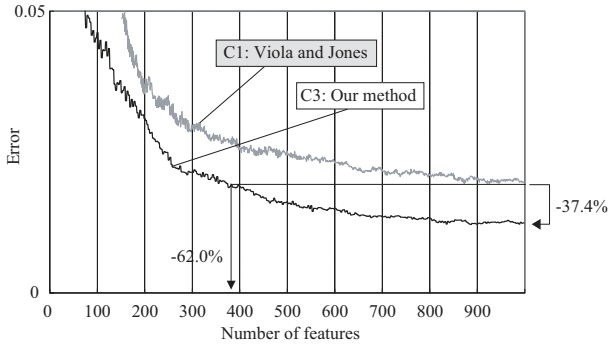

Figure 10. Comparison between C1 and C3.

and C2 are restricted to use a single feature for each weak classifier. C3 and C4 combine three features ($F = 3$). On the training set, the weak classifiers of C1 have error rates between 0.23 and 0.45, whereas the weak classifiers of C3 yield error rates between 0.18 and 0.35. Our method rapidly reduces the training error.

The error curves measured on the test data are shown in Figure 10 and Figure 11. The following conclusions can be drawn from these curves: 1) Given the same number of features, our classifier achieves a lower error rate than Viola and Jones' classifier. The lowest error rate of C1 is 0.0193, whereas that of C3 is 0.0121. Our method reduces the error rate by 37.4%. The lowest error rate of C2 is 0.0122, whereas that of C4 is 0.0095. The error reduction is 22.1%. 2) Our classifier needs smaller numbers of features than Viola and Jones' classifier in order to achieve the same error rate. For example, C3 needs 380 features while C1 needs 1,000 features. That is, our classifier is 2.6 times as fast as Viola and Jones' classifier. C4 needs 470 features while C2 needs 1,000, which is 2.1 times faster. 3) The feature set (a), (b) and (c) improves the performance. Extending the feature set is a reasonable way to construct a powerful classifier. Our method further improves the performance.

4.4. Results (2): Cross-Validation

The second experiment evaluates the reliablity of our method by cross-validation. The face examples of the training data are exchanged with those of the test data. Two

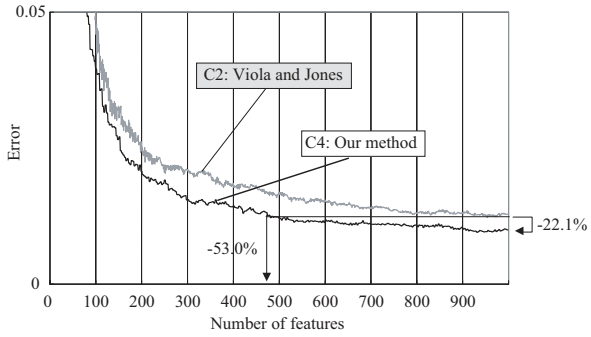

Figure 11. Comparison between C2 and C4.

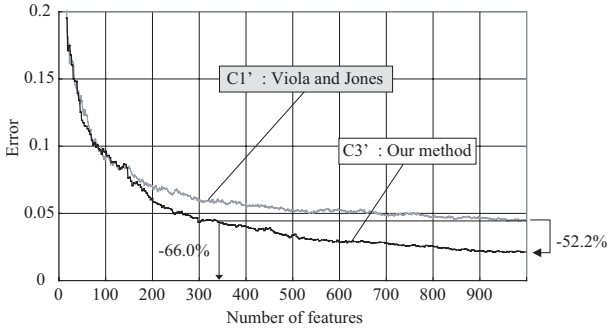

Figure 12. Comparison between C1' and C3'.

classifiers C1' and C3' ($F = 3$), which correspond to C1 and C3 in the first experiment respectively, are learned using the new training data. The error curves measured on the new test data are shown in Figure 12. The test data, that is the original training data, contains difficult face examples with large change in illumination, such as images from Yale and XM2VTS databases. This yields higher error rates for both C1' and C3' than for C1 and C3. However, our classifier C3' achieves lower error rates than C1'. The lowest error rate of C1' is 0.0443, whereas that of C3' is 0.0212. Our method reduces the error rate by 52.2%. C3' needs 340 features while C1' needs 1,000 features to achieve the same error rate.

4.5. Results (3): Choosing the Number of Combined Features

The third experiment compares six classifiers shown in Table 2. Four classifiers indicated as F1, F3, F6 and F9 are learned using fixed F for all weak classifiers. Viola and Jones' classifier is F1. The numbers of combined features for each weak classifier of H1 and H2 are determined by the hold-out method explained in section 3.3. They are different from the validation data used for training. The validation

Table 2. Compared six classifiers.

	F	T	# of features
F1	1	1,000	1,000
F3	3	333	999
F6	6	166	996
F9	9	111	999
H1	4.0	249	998
H2	3.4	289	996

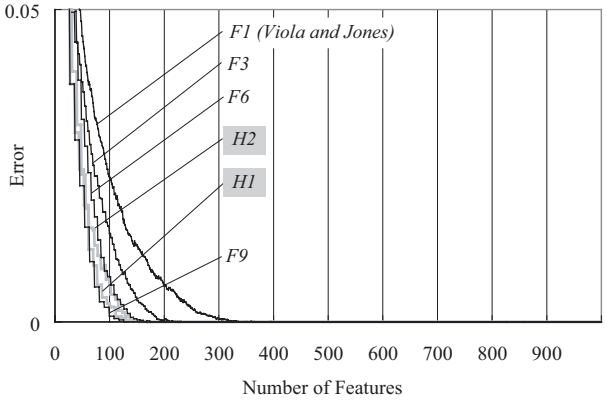

Figure 13. Comparison of the training error.

data for H1 is a set of images collected independently of the training data. H2 is trained using the test data as the validation data, which is the ideal condition. As a result of the training, each weak classifier of H1 and H2 uses 4.0 and 3.4 features respectively on average. A maximum of 9 features are combined.

The error curves measured on the training data are shown in Figure 13. We can see that a larger F rapidly reduces the training error. Figure 14 shows the error curves measured on the test data. F1 consistently has the highest error rate. The ideal classifier H2 yields the best performance as we expected. The error rate of F9 is higher than those of F3 and F6. This means that F9 slightly overfits the training data. Choosing an appropriate F is important for improving performance. From the error curve of H1, we can see that selecting F by the hold-out method is effective.

4.6. Discussion

The experimental results show that our method yields higher performance than Viola and Jones' method even when both methods use the same number of features. That is, adding multiple features in each boosting round is more effective than adding a single feature if distinctive feature co-occurrence exists. In this section, we discuss differences

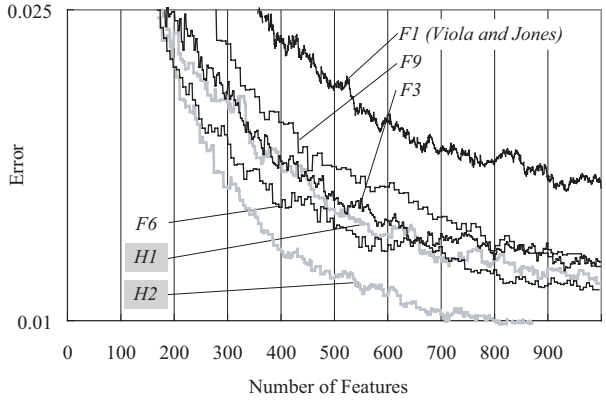

Figure 14. Comparison of the generalization error.

between these two methods.

The AdaBoost algorithm selects one feature so that the number of misclassified examples by the previous feature is minimized. Therefore, the features selected one by one have dependencies on each other. The dependencies are represented as follows [10]:

$$P(z_1, ..., z_T|y, D_T) = P(z_1|y, D_1) \cdot P(z_2|y, D_2) \cdots P(z_T|y, D_T), \tag{11}$$

where z_t is a selected feature and D_t is a sample distribution. Dependencies between features z'_t selected three by three are represented by

$$P(z'_1, ..., z'_T|y, D_{T/3}) = P(z'_1, z'_2, z'_3|y, D_1) \cdots P(z'_{T-2}, z'_{T-1}, z'_T|y, D_{T/3}). \tag{12}$$

The number of weak classifiers is $T/3$ when the total number of features is T. Our method utilizes not only the dependencies obtained from different sample distributions but also feature co-occurrence extracted from the same sample distribution.

The feature co-occurrence improves the classification power of each weak classifier. This decreases the number of too weak classifiers which do not contribute to improving the generalization performance. We can confirm this by the following discussion. The AdaBoost algorithm minimizes the exponential loss $L(G) = E\left(exp\left(-yG(x)\right)\right)$ of a classification function:

$$G(x) = \sum_{t=1}^{T} \alpha_t h_t(x). \tag{13}$$

The loss $L(G)$ is minimized, when [17]

$$G(x) = \frac{1}{2} \log \frac{P(y = +1|x)}{P(y = -1|x)}. \tag{14}$$

Table 3. Performance comparison of classifiers using three features.

	Viola-Jones	Our method
Training error	0.192	0.177
Generalization error	0.508	0.231

Viola and Jones' classifier can be seen as a Bayes classifier which approximates $P(y|x)$ by

$$P(y|x) \approx P\left(y|z_1\right) \cdot P\left(y|z_2\right) \cdots P\left(y|z_T\right). \quad (15)$$

Our classifier approximates $P(y|x)$ by

$$P(y|x) \approx P\left(y|z'_1, z'_2, z'_3\right) \cdots P\left(y|z'_{T-2}, z'_{T-1}, z'_T\right). \quad (16)$$

If distinctive feature co-occurence exists, the latter approximation based on the joint probability of multiple features increases the difference between $P(y = +1|x)$ and $P(y = -1|x)$. This leads an improvement of the classification performance.

Table 3 shows error rates measured on the training data and the test data using only three features selected in the earliest boosting rounds. Three features jointly selected by our method yields lower error rates than Viola and Jones' method.

5. Conclusions

In this paper, we describe a new visual feature, called joint Haar-like feature, for face detection and show how it can be selected. The feature captures the characteristics of human faces and improves the performance of each weak classifier. A face detector is learned by stagewise selection of the joint Haar-like features using AdaBoost. Experimental results with many images show that our method yields higher classification performance than Viola and Jones' method. The results also indicate that *'too weak'* classifiers used in the conventional method do not contribute to improving the generalization performance.

The proposed method gives a new framework for feature selection and it is not restricted to the use of only Haar-like features. Gabor features are strong candidates for further improvement of the classification performance. Applying other boosting algorithms to this framework is also effective. We have confirmed that RealBoost [8] yielded higher performance than AdaBoost.

Acknowledgments

This work has been done as part of the contract research "User oriented multimedia technology" conducted by National Institute of Information and Communications Technology (NICT).

References

[1] M. H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Trans. on PAMI*, 24(1):34–35, 2002.

[2] K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE Trans. on PAMI*, 20(1):39–51, 1998.

[3] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Trans. on PAMI*, 20(1):23–38, 1998.

[4] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. *Proc. of CVPR*, pages 130–136, 1997.

[5] B. Heisele, T. Poggio, and M. Pontil. Face detection in still gray images. *A.I. Memo*, (1687), 2000.

[6] C. P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. *Proc. of ICCV*, pages 555–562, 1998.

[7] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proc. of CVPR*, pages 511–518, 2001.

[8] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine Learning*, 37(3):297–336, 1999.

[9] C. Liu and H. Y. Shum. Kullback-leibler boosting. *Proc. of CVPR*, pages 587–594, 2003.

[10] S. Z. Li and Z. Q. Zhang. Floatboost learning and statistical face detection. *IEEE Trans. on PAMI*, 26(9):1112–1123, 2004.

[11] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on Real AdaBoost. *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition*, pages 79–84, 2004.

[12] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. *Proc. of ICIP*, 1:900–903, 2002.

[13] S. D. Strearns. On selecting features for pattern classifiers. *Proc. of ICPR*, pages 71–75, 1976.

[14] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. *Proc. of CVPR*, pages 746–751, 2000.

[15] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman. From few to many: generative models for recognition under variable pose and illumination. *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition*, pages 277–284, 2000.

[16] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: the extended M2VTS database. *Proc. of 2nd Conf. on Audio and Video-based Biometric Personal Verification (AVBPA99), URL: http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb*, 1999.

[17] J. Friedman, T. Hastie, and R. J. Tibshirani. Additive logistic regression: a statistical view of boosting. *Technical report, Department of Statistics, Sequoia Hall, Stanford Univerity*, July 1998.