Visual Object Recognition with Supervised Learning

Bernd Heisele, *MIT Center for Biological and Computational Learning*

A component-based approach to visual object recognition rooted in supervised learning allows for a vision system that is more robust against changes in an object's pose or illumination.

Learning figures prominently in the study of visual systems from the viewpoints of visual neuroscience and computer vision. Whereas visual neuroscience concentrates on mechanisms that let the cortex adapt its circuitry and learn a new task, computer vision aims at devising effectively trainable systems. Vision systems that learn and adapt are one of the most important trends in computer vision research. They might offer the only solution to developing robust, reusable vision systems.

Supervised learning

Over the last four years, our team at MIT developed a supervised-learning approach to address vision perception in machines. The research is rooted in the mathematical foundations of learning and paralleled by neuroscience studies. For instance, radial basis function networks, which originated from the mathematics of learning theory, suggested a view-based model for biological object recognition.[1] Psychophysical data[2] and physiological experiments in cortex[3] found evidence for view-tuned neurons that the model had predicted.

In our work, we distinguish between two main object recognition tasks: *categorization* and *identification*. We use the term categorization for between-class object classification (such as classification between faces and other objects) and identification for within-class object classification (such as recognizing someone's face among other faces).

Our approach considers recognition a supervised-learning problem. We label a set of training images and use the labels for training the classifier. In categorization, the label specifies the class of the object in the image; in identification, it specifies the individual object. We train the learning module with a set of input-output examples, which are image pairs and their associated labels. The learning task's difficulty depends on the training set's size and composition and on how much the training examples cover the variability required for generalization. For instance, the learning module couldn't identify a face from any viewpoint if trained with only a single view of that face. Conversely, the same module, if trained with a large set of examples covering the relevant variability, might perform the task.

Although we trained our component-based systems on various object classes, this article focuses on human faces. Face recognition has a wide variety of real-world applications, ranging from human-machine interfaces to surveillance systems.

Background: Statistical-learning theory

In supervised learning, a machine chooses a function that best describes the relation between the inputs and the outputs. SLT[4] asks how well the chosen function generalizes on previously unseen inputs.

Regularization theory framework

Following work done elsewhere,[5] we approach SLT using regularization theory.[6] We are interested in learning schemes that lead to solutions of the form

$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i K(\mathbf{x}, \mathbf{x}_i), \qquad (1)$$

where $\mathbf{x}_i$, $i = 1, \ldots, l$ are the input examples, K is a certain symmetric positive-definite function named kernel, and α_i is a set of parameters to be determined from the examples. Solution f is an example of a *regularized* solution and is found as the minimizer of functionals of the type:

1094-7167/03/$17.00 © 2003 IEEE
Published by the IEEE Computer Society

$$\Phi[f] = \frac{1}{l}\sum_{i=1}^{l} V(y_i, f(\mathbf{x}_i)) + \lambda \|f\|_K^2. \qquad (2)$$

V is a *loss function*, which measures the predicted output $f(\mathbf{x}_i)$'s goodness with respect to the given output y_i. $\|f\|_K^2$ is a smoothness, or regularizing, term, which is the norm in the reproducing kernel Hilbert space that is defined by kernel *K*. λ is a positive parameter controlling the relative weight between the data and the regularizing term. The choice of the loss function determines different learning techniques, each leading to a different learning algorithm for computing the coefficients α_i in Equation 1.

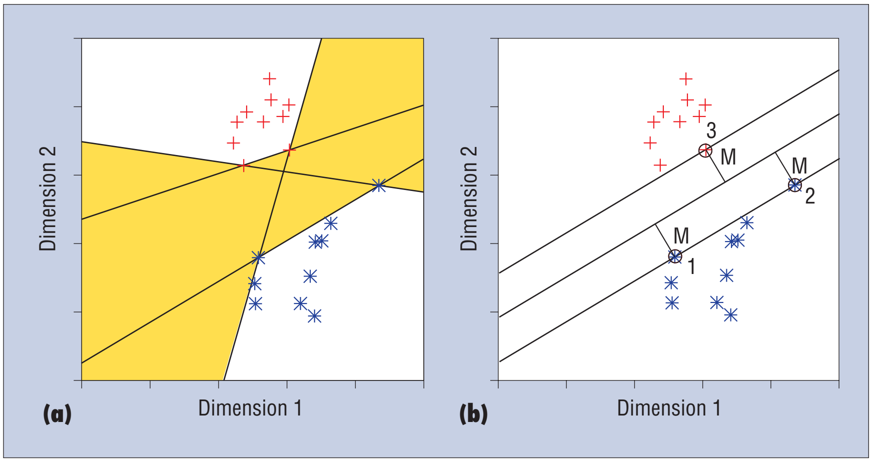

Figure 1. (a) The yellow area shows all possible hyperplanes that separate the two classes (represented by + and *). (b) The optimal hyperplane maximizes the distance to the closest points. These points (1, 2, and 3) are support vectors. The distance M between the hyperplane and the SVs is called the *margin*.

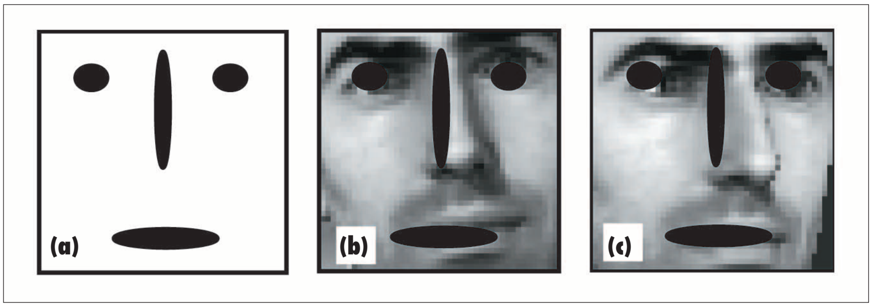

Figure 2. Matching with a single template: (a) the schematic global template of a frontal face; slight face rotations (b) in the image plane and (c) in depth lead to considerable discrepancies between template and face.

Support vector machines

You obtain SVM classification[7] by using the following loss function *V*:

$$V(y, f(\mathbf{x})) = (1 - y\,f(\mathbf{x}))_+, \qquad (3)$$

where $(t)_+ = t$ if $t > 0$ and is zero otherwise.

You can find the coefficients α_i in Equation 1 by solving a quadratic programming problem with linear constraints. With SVMs, remarkably, the loss function leads to *sparse* solutions. Typically, only a small fraction of the coefficients α_i in Equation 1 are nonzero. The data points $\mathbf{x}_i$ associated with the nonzero α_i are called *support vectors*. If you discarded all data points that aren't support vectors from the training set, you'd find the same solution.

An SVM has an interesting geometrical property: The separating surface has maximum distance to the closest points in the training data (see Figures 1a and 1b).

An almost unbiased upper bound *L* on the expected error of an SVM trained on *l* data points drawn according to a probability $p(\mathbf{x}, y)$ is given by[4]

$$L = \frac{1}{l} E\left[\min(m_l, \frac{r_l^2}{M_l^2})\right], \qquad (4)$$

where $E[\cdot]$ denotes the expectation over the probability $p(\mathbf{x},y)$, m_l the number of support vectors, r_l the radius of the smallest sphere containing the support vectors, and M_l the margin of the SVM trained on *l* data points.

Object categorization

Detecting objects in images is a major task in visual-scene analysis. A common way to do this is to shift a search window over an input image and categorize the object in the window with a classifier. The main problem with categorization is the large range of possible variations within an object class. The classifier must generalize not only across different viewing and illumination conditions, but also across a class's different exemplars. To simplify categorization, most vision systems use sets of binary classifiers (one for each object category). In our approach, we only consider the binary categorization task where the classifier must separate one class of objects from all other objects.

Component-based approach

Often, people approach object categorization by representing all the search window's contents by one feature vector that is fed to a single classifier. This global approach worked well for detecting objects under fixed viewing conditions.[8,9] However, problems occur when the objects' viewpoint and pose vary, especially when the training set doesn't cover all viewing variations in the test set. Figure 2 illustrates this for a face detection system that is trained on frontal, upright faces and tested on rotated faces. A single face template can represent the result of training a linear classifier on frontal faces. Even for small rotations, the template clearly deviates from the rotated faces. To overcome this problem, we developed a component-based approach[10] that breaks the object into a set of components that are interconnected by a flexible geometrical model. Although their relative positions change, each component varies less under pose changes than the pattern belonging to the whole object. Figure 3 illustrates the component-based idea.

From this, we derive two main issues: how to include information about the geometrical relation between components in the classification process and how to choose a set of relevant components.

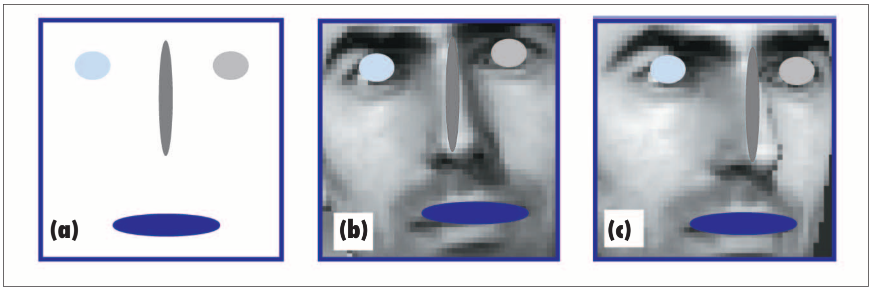

Figure 3. Matching with a set of component templates: (a) the schematic component templates for a frontal face; shifting the component templates can compensate for slight face rotations (b) in the image plane and (c) in depth.

Geometrical classifier

We developed a two-level classification system for face detection that implies geometrical relations between components (see Figure 4). On the first level, component classifiers independently detect face components. On the second level, the combination classifier checks if the components' geometrical configuration corresponds with the learned geometrical model of a face.

Learning components

A component-based approach must also know how to choose the set of discriminatory object parts. For faces, an obvious choice would be the eyes, nose, and mouth. For other classes, it might be harder to define a set of intuitively meaningful components. Instead of manually choosing the components, it would make more sense to choose automatically based on their discriminative power and robustness against pose and illumination changes. We developed a method that automatically determines rectangular components from a set of face images. The algorithm started with a small, rectangular component located around a preselected point on the face (center of the left eye, for example). The algorithm extracted the component from all face images to build a training set of positive examples. We also generated a training set of nonface patterns that had the same rectangular shape as the face component. After training an SVM on the component data, we determined the SVM's performance based on a rough estimate $\tilde{L}$ of the upper bound L in Equation 4 given by

$$\tilde{L} = \frac{1}{l}\frac{R^2}{M^2}. \qquad (5)$$

l is the number of training patterns, R the diameter of the smallest sphere containing the data points in the training set (not just the support vectors as in the earlier equation), and M the classifier's margin. After determining $\tilde{L}$, we enlarged the component by expanding the rectangle into one of the four directions. Again, we generated training data, trained an SVM, and determined $\tilde{L}$. We did this for expansions in all four directions and kept the expansion that decreased $\tilde{L}$ the most. We continued this process until the expansions in all four directions led to an increase of $\tilde{L}$. Figure 5 shows the results of component growing for 14 components.

We compared the component-based system to a global classifier trained on the whole face pattern. The training and test data included faces rotated between about –45° and 45° in depth. Figure 6 shows the classification performance of a linear SVM global classifier and the component-based system. Figure 7 shows some detection results the component-based system generated.

Object identification

Object identification distinguishes between exemplars of the same class. This is difficult

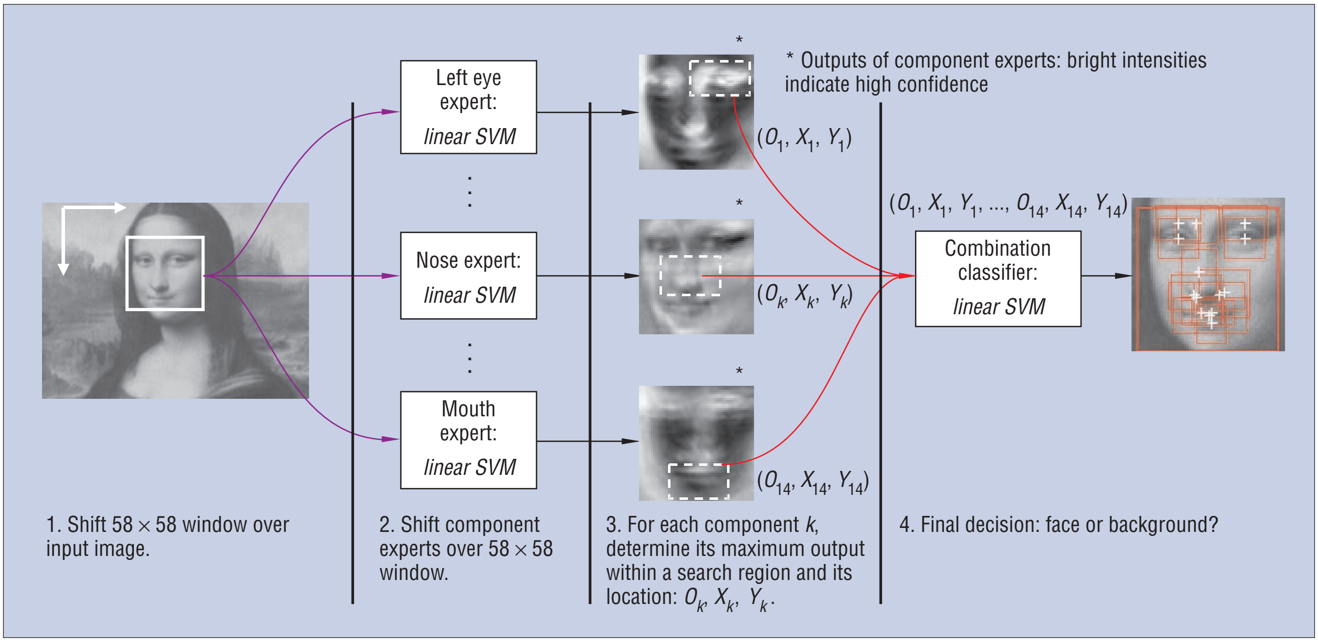
* Outputs of component experts: bright intensities indicate high confidence
Left eye expert:
linear SVM
Nose expert:
linear SVM
Mouth expert:
linear SVM
*
*
*
(O_1, X_1, Y_1)
(O_k, X_k, Y_k)
(O_{14}, X_{14}, Y_{14})
$(O_1, X_1, Y_1, \ldots, O_{14}, X_{14}, Y_{14})$
Combination classifier:
linear SVM
1. Shift 58 × 58 window over input image.
2. Shift component experts over 58 × 58 window.
3. For each component k, determine its maximum output within a search region and its location: O_k, X_k, Y_k.
4. Final decision: face or background?

Figure 4. System overview of the component-based detection system.

because objects belonging to the same class might differ only in details.

Component-based face identification

Face identification is a classic computer vision problem. Various approaches include eigenfaces, linear discriminant analysis, elastic graph matching, and SVMs. Following up on the idea of component-based face detection, we built a face identification system[11] that is more robust against pose changes than common systems using global approaches. We extracted facial components from the face image using the component-based detector described earlier. We normalized the components in size, combined them into a single feature vector, and fed them to the identification classifiers. As in the global approach, we ended up with one feature vector as input to the identification classifier. However, each feature was attached to a facial location (for example, the left corner of the mouth) rather than to a fixed *x-y* location in the image.

We performed experiments on a database of five subjects and compared the component-based system with a global system. Figure 8 shows the receiver operating characteristic curves for the two systems. Each point on the curve corresponds to a different rejection threshold of the classifier. At the end point of an ROC curve, the rejection rate is zero.

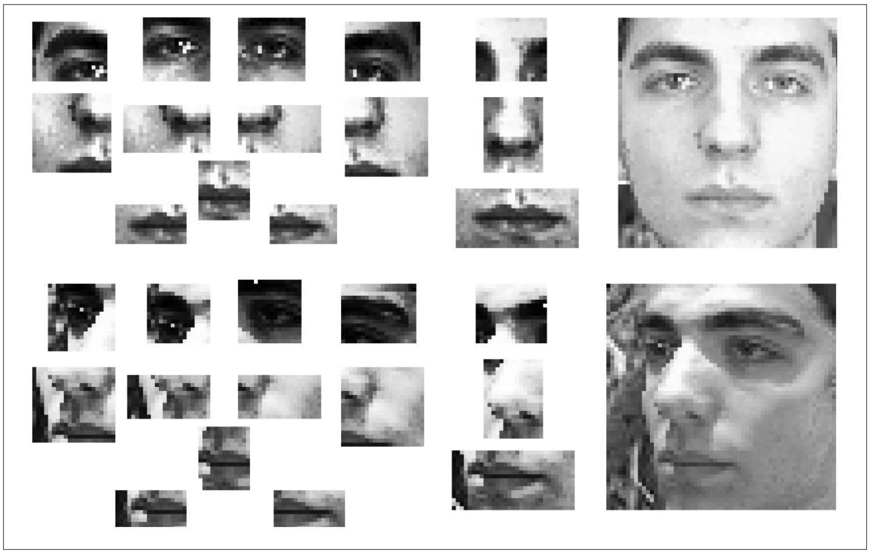

Figure 5. The 14 learned components for a frontal and a half-profile view of a face.

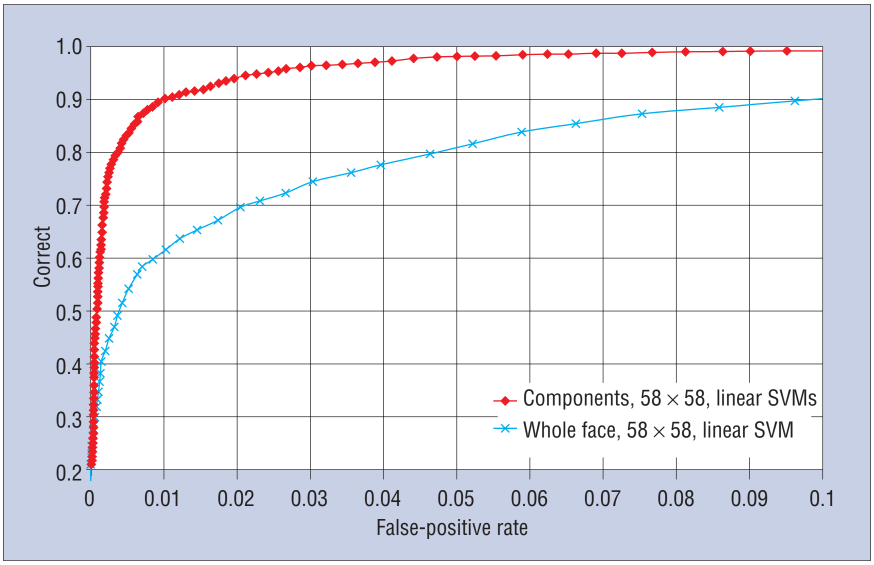

Figure 6. Receiver operating characteristic curves for a linear whole face classifier and a component classifier consisting of 14 linear component classifiers and a linear combination classifier. The graph gives the false-positives (FPs) relative to the number of nonface images. The test set consists of 1,834 faces and 24,464 nonfaces.

In another article,[10] we suggested using computer graphics techniques in face detection. We generated thousands of synthetic face images from 3D head models to train a face detection system. Rendering 3D models instead of manually extracting face images from real pictures saved much time. Furthermore, we could modify the rendering parameters (viewpoint and illumination, for example) arbitrarily. In more recent work, we used 3D morphable models[12] to fit a 3D face model to only two face images of a person. Based on the 3D models, we trained a pose- and illumination-invariant face identification system.[13]

Our approach might also help address the old problem of *scene interpretation*. Training classifiers to detect specific object classes in the image leads to the idea that you could use a sufficiently complete dictionary of such classifiers for scene interpretation. Because the locations of individual objects in a scene are correlated (for example, telephones are usually located on desks, paintings on the wall), the problem arises of how to exploit these spatial relations to perform scene interpretation. This resembles the previously discussed problem of including information about the spatial relation between components into object categorization. ■

References

1. T. Poggio and S. Edelman, "Network that Learns to Recognize 3D Objects," *Nature*, vol. 343, no. 6255, Jan. 1990, pp. 263–266.

2. H.H. Bülthoff and S. Edelman, "Psychophysical Support for a 2D View Interpolation Theory of Object Recognition," *Proc. Nat'l Academy of Sciences*, vol. 89, no. 1, Jan. 1992, pp. 60–64.

3. N. Logothetis, J. Pauls, and T. Poggio, "Shape Representation in the Inferior Temporal Cortex of Monkeys," *Current Biology*, vol. 5, no. 5, May 1995, pp. 552–563.

4. V.N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

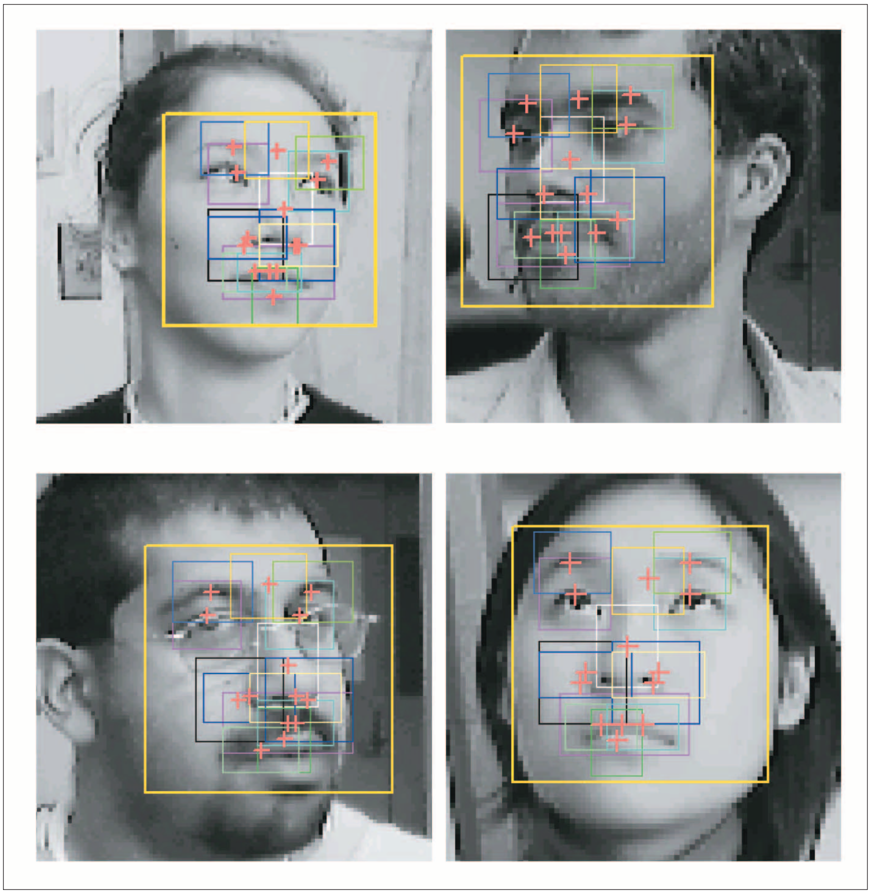

Figure 7. Faces detected by the 14-component system.

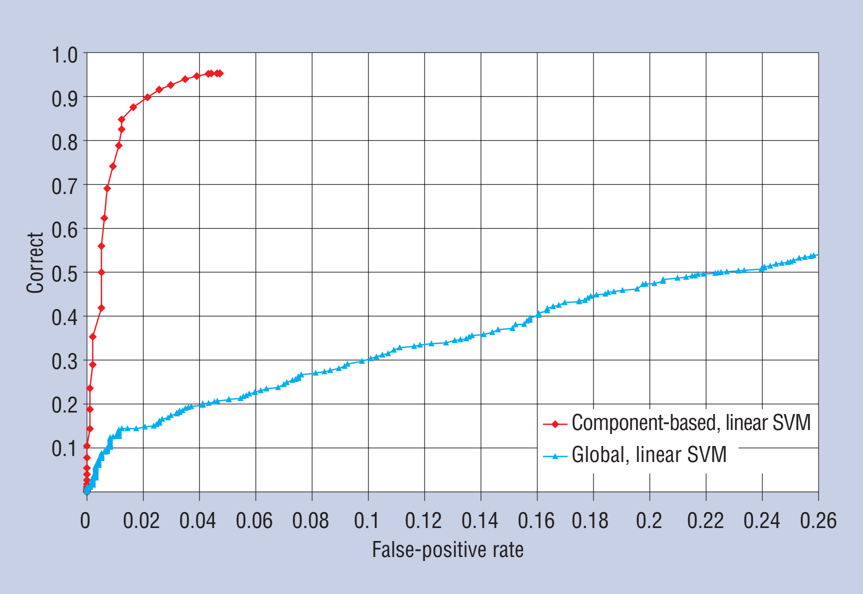

Figure 8. Receiver operating characteristic curves for the component-based and global face identification systems. Both systems used linear SVMs as classifiers, one for each person in the database. We trained the systems on five people (8,593 images, frontal and rotated) and tested on the same five people (974 different images, frontal and rotated).

5. T. Evgeniou, M. Pontil, and T. Poggio, "Regularization Networks and Support Vector Machines," *Advances in Computational Math.*, vol. 13, 2000, pp. 1–50.

6. A.N. Tikhonov and V.Y. Arsenin, *Solutions of Ill-posed Problems*, W.H. Winston, 1977.

7. C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning*, vol. 20, no. 3, Sept. 1995, pp. 1–25.

8. C. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *Int'l J. Computer Vision*, vol. 38, no. 1, June 2000, pp. 15–33.

9. H. A. Rowley, S. Baluja, and T. Kanade, *Rotation Invariant Neural Network-Based Face Detection*, tech. report CMU-CS-97-201, Computer Science Dept., Carnegie Mellon Univ., Pittsburgh, 1997.

10. B. Heisele et al., "Component-based Face Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (CVPR 2001), IEEE CS Press, pp. 657–662, 2001.

11. B. Heisele, P. Ho, and T. Poggio, "Face Recognition with Support Vector Machines: Global versus Component-Based Approach," *Proc. 8th Int'l Conf. Computer Vision* (ICCV 2001), IEEE CS Press, 2001, pp. 688–694.

12. V. Blanz and T. Vetter, "A Morphable Model for the Synthesis of 3D Faces," *Proc. Siggraph 99*, ACM Press, 1999, pp. 187–194.

13. J. Huang, V. Blanz, and B. Heisele, "Face Recognition Using Component-Based SVM Classification and Morphable Models," *Proc. Pattern Recognition with Support Vector Machines* (SVM 2002), Springer-Verlag, pp. 334–341.

For more information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.

The Author

Bernd Heisele is senior scientist at the Honda Research Laboratory in Boston and a visiting researcher at the MIT Center for Biological and Computational Learning. His research interests include learning-based object detection and recognition and motion analysis in image sequences. He received a PhD from the University of Stuttgart. Contact him at the Honda Research Institute US, 145 Tremont St., Boston, MA 02111; bheisele@honda-ri.com.