International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 23, No. 3 (2009) 379–400

World Scientific
www.worldscientific.com

TOWARDS PRACTICAL FACIAL FEATURE DETECTION

MICAH ECKHARDT

Machine Perception Laboratory
Institute for Neural Computation
University of California, San Diego
La Jolla, CA 92093, USA
micahrye@mplab.ucsd.edu

IAN FASEL

Department of Computer Science
University of Arizona
Tucson, AZ 85721, USA
ianfasel@cs.arizona.edu

JAVIER MOVELLAN

Machine Perception Laboratory
Institute for Neural Computation
University of California, San Diego
La Jolla, CA 92093, USA
movellan@mplab.ucsd.edu

Localizing facial features is a critical component in many computer vision applications such as expression recognition, face recognition, face tracking, animation, and red-eye correction. Practical applications require detectors that operate reliably under a wide range of conditions, including variations in illumination, pose, ethnicity, gender and age. One challenge for the development of such detectors is the inherent trade-off between robustness and precision. Robust detectors tend to provide poor localization and detectors sensitive to small changes in local structure, which are needed for precise localization, generate a large number of false alarms. Here we present an approach to this trade-off based on context dependent inference. First, robust detectors are used to detect contexts in which target features occur, then precise detectors are trained to localize the features given the detected context. This paper describes the approach and presents a thorough empirical examination of the parameters needed to achieve practical levels of performance, including the size of the training database, size of the detector's receptive fields and methods for information integration. The approach operates in real time and achieves, to our knowledge, the most accurate localization performance to date.

Keywords: Machine vision; feature detection; image registration.

1. Introduction

Localizing facial features is a critical component in many computer vision applications, including expression recognition, avatar animation, face recognition, head pose estimation, and artifact removal (e.g. red eye effect) in digital camera.[5, 8, 9, 15, 25, 27, 30] Despite its importance, facial feature localization is still an unsolved problem for applications that need to operate under a wide range of conditions that include realistic variations in illumination, ethnicity, gender, age, pose, and imaging hardware.[5]

While many approaches to feature detection have been proposed and tested on several benchmark datasets, the particular details for how any of these methods could be pushed to performance levels reliable enough for practical use is rarely studied systematically. The challenges are both theoretical and empirical. One theoretical challenge is an inherent trade-off between robustness and precision. Robust detectors that work reliably in a wide variety of conditions tend to provide poor localization performance, while detectors capable of distinguishing small deviations from target locations tend to generate a large number of false alarms. Here we present an approach to this trade-off based on context dependent inference (CDI): first, robust detectors are trained to detect the context in which target features occur, and then precise detectors are trained to localize the target features given the context.

Another challenge is the historical aversion of the computer vision community for empirical parametric studies. Such studies were critical in the development of fields such as automatic speech recognition,[16] but their importance has in general not been recognized yet in the computer vision community. The consequence has been slower technological progress, scientific progress, and practical application of computer vision in many domains.

Here we document the process of developing state of the art feature detectors using machine learning methods under the CDI approach described above. Empirical studies are presented on a large and challenging dataset of faces obtained from the Web, including a wide variety of illumination and rendering conditions, both indoors and outdoors. We investigatge and describe the effect on performance of a wide range of intervening factors. The parameters and algorithms that optimized performance, as well as those that did not, are clearly described to facilitate replicability and progress in the field.

2. Overview of the Approach

There is a fundamental trade-off inherent to the problem of feature localization. While robust feature detectors tend to localize poorly, detectors sensitive to small variations, which are needed for precise localization, tend to produce a large number of false alarms. One common approach to solve this trade-off is based on the operation of a set of independent feature detectors.[5, 6, 10, 14] The output of these detectors (e.g. a detector for each eye, a detector for the tip of the nose, a detector for points

on the mouth, etc.) are combined by considering spatial configurations that match the distribution of inter-feature distances typical of the human face.[6, 17, 18, 28] Unfortunately, the computational complexity of this approach scales exponentially with the number of false alarms of each feature detector, and the number of basic feature detector types, making such approaches generally impractical for real-time use.

The approach we explore here is based on context dependent inference, (CDI) an idea that was first formalized by Yuille and Bulthoff[31] to help explain biological vision. They proposed that it is too difficult to develop context independent perceptual systems capable of operating robustly and precisely under all possible conditions. Instead, they proposed that perceptual inference may be better handled using context-dependent experts, each specialized for making inferences given a specific context. The essence of this idea can be formalized as follows: let y be an observed image, t the location of a target feature (e.g. the left eye) and c the image region rendering a context relevant to this target (e.g. the set of pixel locations in the image that render human faces). Our goal is to infer the location of the target feature on the image plane. The information needed to solve this problem is contained in the posterior probability of the target t given the image y. Using the law of total probability, we have that

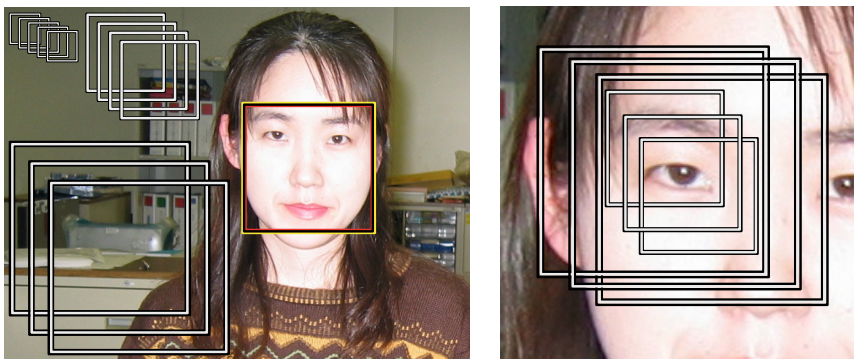
$$p(t \mid y) = \sum_{c} p(c \mid y) p(t \mid c, y) \tag{1}$$

where $p(c \mid y)$ is the posterior probability of a context given the observed image. For example, c may partition the image into pixels rendering a face and pixels rendering a generic everything else. The term $p(t \mid c, y)$ is a context specific target detector. It provides information about the location of the target, provided it operates in a specific context c. Thus (1) tells us that if we want to localize a target feature we can do so by combining the output of a system that detects the relevant context in which the targets occur and another system that localizes the target in given contexts.

A CDI approach to feature detection was first presented in Ref. 9. In this paper, while we maintain the same theoretical framework, we additionally provide a highly detailed analysis of the different factors affecting accuracy, and analyze performance on larger and more realistic databases than previously considered. We show that the resulting real-time system achieves performance levels unsurpassed by any other feature detector method published in the literature to date.

2.1. *Real-time inference architecture*

Figure 1 describes the general workflow of the inference architecture. It consists of two stages: the first stage detects the context under which the target features occur, in this case human faces. This first stage operates under very general background and illumination conditions, narrowing down the plausible locations of the target features in the image plane. It is based on the cascade of boosted classifiers architecture of Viola and Jones,[24] but with two key changes: (1) We use continuous



(a) (b)

Fig. 1. Context sensitive search (a) first, the contexts of interest, in this case faces, are selected by scanning a sliding detection window across the entire image at multiple scales, (b) next the target features are precisely localized given the context. In both cases we use a multiscale sliding-window detector approach.

nonparametric transfer functions rather than binary threshold functions over the box-filter features. This allows us to obtain continuous likelihood ratio estimates for each possible image patch, rather than just binary decisions, (2) we eliminate the cascaded architecture in favor of a probabilistic sequential decision making architecture. Under this approach, a decision is made on a feature-by-feature basis as to whether to process another feature or to stop processing and decide *face* or *nonface* using the features processed up to that point. In addition, contrary to the Viola–Jones architecture, the information from past features is never discarded, thus resulting in significant improvements in the speed-accuracy tradeoff. A full description of this approach can be found in Ref. 9.

The second stage, which is the focus of this paper, specializes in achieving high feature localization accuracy, provided it operates on the regions selected by the previous stage. This second stage uses the same inference architecture as the previous stage, but now applied only within the detected face region. The combined system operates at approximately ten 320×240 video frames per second using a single thread of a standard desktop computer (Apple PowerMac G5, 2.5 Ghz).

2.2. *Learning architecture*

The proposed approach requires conditional likelihood-ratio estimates; that is, given an arbitrary image patch y we need an estimate for the ratio between the probability of such a patch being generated by the target class versus the background class. Here we learn these likelihood ratios using a *boosting* algorithm known as *GentleBoost*.[12] Boosting[11] refers to a family of machine learning algorithms that builds accurate (strong) classifiers by combining a collection of weak classifiers. Each of these weak classifiers is chosen in a sequential manner for its capacity to reduce the mistakes made by the current collection of weak classifiers. While each weak classifier may

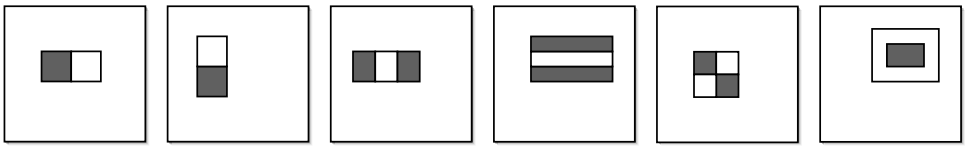


Fig. 2. Each filter is computed by taking the difference of the sums of the pixels in the white boxes and grey boxes. Filter types include those in Ref. 24, plus a center-surround type filter.

perform only slightly above chance, the combined system (i.e. the strong classifier) may achieve very high levels of accuracy.

In Ref. 12, it was shown that boosting methods can be reinterpreted from the point of view of sequential maximum likelihood estimation, an interpretation that makes it possible to use these methods within the framework proposed here. Learning in GentleBoost is accomplished by sequentially choosing weak classifiers and combining them to minimize a chi-square error function. In our application, each weak classifier consists of a simple linear filter, selected from a large fixed library of filters, followed by a nonlinear transfer function.

The pool of filters we use are the same as those used in Ref. 24, with the addition of a center-surround filter class (see Fig. 2). The main reason for using these relatively simple features is that they can be computed very efficiently in general purpose computers without the need of specialized hardware (see Refs. 23 and 24 for detailed explanation). In Ref. 24, the nonlinear transfer function was a simple threshold function whose output was in the set $\{-1, +1\}$. In this paper, we use a piecewise constant function whose parameters are chosen by the GentleBoost algorithm. This allows each weak classifier to output arbitrary real values in the range $[-1, +1]$ rather than simply binary decisions.

3. Database Description

Here we briefly describe some of the most commonly used databases for training and testing facial feature detectors. FERET (frontal images)[22] is a free, publicly available database with 3880 images taken in controlled settings with no background clutter and little variation in illumination. XM2VTS[29] and BANCA-C/WorldModel[1] are commercially available databases. XM2VTS contains 1180 high quality frontal face images. BANCA-C/WorldModel contains 2380 frontal face images with no background clutter and some variation in illumination. Images in the aforementioned databases were taken in controlled environments with uniform background.

The BANCA-D/A and BioID[2] databases attempt to simulate real-world conditions. BANCA-D/A contains 4160 frontal face images with cluttered backgrounds, variable illuminations and head pose variation. The free and publicly available BioID database contains 1521 frontal face images that vary with respect to illumination, background, scale and head pose. Based on results from the literature,[9, 15, 32]

Fig. 3. Sample of cropped images from the GENKI database.

these databases are considered more challenging than FERET, XM2VTS and BANCA-C/WorldModel databases.

While these databases have helped to advance research in the area of facial feature detection and face recognition, they do not represent the variety of illumination and rendering conditions found in many real life applications, such as consumer cameras, surveillance systems or social robots, in which factors such as illumination and background clutter cannot be controlled. For this reason we collected a new database, named GENKI, that contains approximately 70,000 images collected from the World Wide Web (see Fig. 3), a portion of this database has been made available to the public and is refereed to as GENKI-4K[13] in this paper. This collection is highly varied with respect to illumination, background clutter, head pose, age, ethnicity, partial occlusions, image compression artifacts, and image resolution. Images in the database were hand labeled for the location of the temporal and nasal corner of right and left eyes, the center of the tip of the nose, mouth corners, mouth center (defined as the estimate of the location of the intersection between the line defined by the labial furrow and the curve defined by the end of the upper teeth) and pose: roll, pitch, and yaw. Head pose ranges are: pitch $-30°$, $+48°$; yaw $+/-62°$; roll $-60°$, $+53°$. For our purposes the center of the eye is defined as the midpoint between the labeled temporal and nasal corner of the eye.

4. Empirical Studies

Here we present a series of experiments whose goal is to investigate the effects of intervening factors on system performance, and identify trends for improving performance. Because the set of all possible combinations of parameter settings is too large to search exhaustively, we proceeded in a sequential manner, identifying

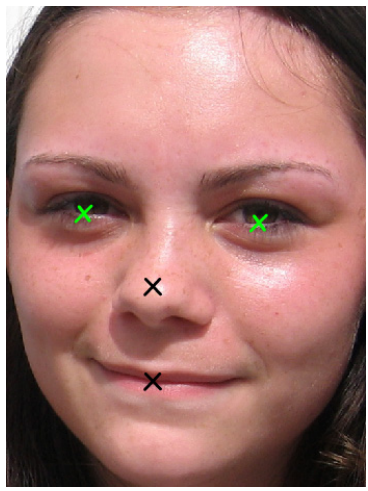


Fig. 4. Face with hand labeled features.

promising parameter values, fixing them and then varying other parameters, and repeating back to earlier parameters several times. Parameter values which remain fixed throughout the work are based on preliminary tests and previous work.[9] We present performance on three types of facial features: (1) the center of the eyes (here we only report left eye performance, since right eye performance is equivalent), (2) tip of the nose, and (3) center of the mouth (see Fig. 4). All classifiers were trained using examples from the GENKI database. Several different subsets were used for training (as described below), however when testing on GENKI, all experiments were tested on the same 10,000 image subset of images, never seen during training.

4.1. *Performance statistics*

A recent trend in the literature is to report performance on facial feature localization in terms of interocular distance,[3,6,15] the distance between the centers of the eyes. An important aspect of this unit of measure is that it is scale independent. A drawback is that it is relatively large, on the order of several centimeters, and as a consequence not intutively appealing. We found that in many cases it was more intuitive to compare the precission in terms of iris widths. Using a sample of images from the GENKI database we found that the average interocular distance was 6.63 times larger than the diameter of the iris. For simplicity we chose the closest integer to 6.63 as a standard. Thus, we defined a **Standard Iris Diameter (SID)** as 1/7 of the interocular distance, and use it as the basic unit of measurement to present performance values. This unit is intuitively appealing while also being easy to convert to the more traditional interocular distance standard.

When comparing the output of the feature detector to the hand labeled feature locations, the "error" of a specific feature detection is defined as the euclidean

distance (in SIDs) between the system's final output and the hand labeled location. Here we report two error statistics: the root mean square error (RMSE), and the median absolute error (MAE), which is less sensitive to outliers. Another useful measure is the proportion of times that the absolute distance between the system's output and the hand labels is below a threshold. Unless otherwise stated, we use a relatively lax threshold of 1.75 (0.25 interocular distance) SID, a common detection criteria.

In the remainder of this section we report on a series of experiments designed to investigate the effects of the following factors: (1) size of the search region, (2) receptive field size, (3) sampling rate (4) training set size, (5) selection of positive and negative examples, (6) number of rounds of training, (7) information integration, and (8) pose variation.

4.2. *Size of the search region*

As described in Sec. 2.1 and illustrated in Fig. 1, we use a sliding-window approach to search for target facial features at different locations and scales. First, a context detector segments a region of the image likely to contain a face. In this initial face detection stage, we first scan patches of size 24×24, the minimum scale of interest, and shift one pixel at a time until all possible patches of this size are scanned. Each larger scale is chosen to be approximately 1.2 times the previous scale, and the window shifting amount is scaled by the same proportion.

After the face is found, we search for facial feature points (FFPs) within the face. Rather than search within the entire face region for a particular FFP, we limit our search to the areas most likely to contain the desired feature. To this end we model the joint prior distribution of the offset in horizontal and vertical directions, along with scale, as a three-dimensional Gaussian distribution. The parameters for this distribution are simply the empirical mean and covariance of these values found in a sample from the GENKI database. The resulting model was used to determine search regions for the contextual feature detectors. We refer to these search regions as *regions of interest*, or *ROI*s. The size of a ROI is determined by a maximum allowed Mahalanobis distance from the most probable prior location of the target features. We tested the effect of varying this maximum threshold over the following Mahalanobis distances: $\{1.215, 2.366, 4.108, 6.251, 16.275, 21.101, 25.902, 30.665\}$. As shown in Fig. 5, performance increased with larger search regions, but improvements were very small for search regions with Mahalanobis distances beyond 15. Therefore in the remaining experiments we fixed the ROI threshold distance to 15.

4.3. *Receptive field size*

The size and resolution of the detector influences the localization performance and it is unclear *a priori* what the optimal values should be. For example, should eye

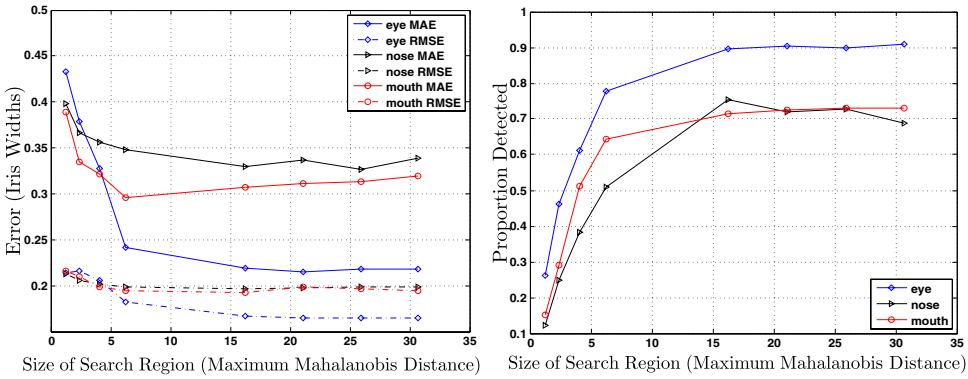


Fig. 5. Effect of search region size.

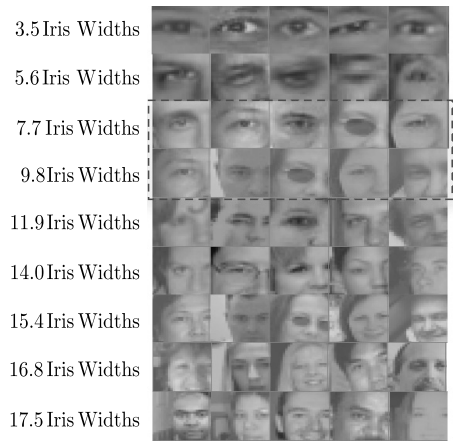


Fig. 6. Examples of varying receptive field size (optimal sizes in dashed box).

detectors search only for the pixels making up the iris, should they use information from the entire eye region, or should they use information far beyond the eye region? To study this issue we tested the effect of varying the size of the detector's *receptive field*, defined as a square image patch of fixed width relative to the size of the detected face. The tested receptive field widths, measured in SIDs, were $\{3.5, 5.6, 7.7, 9.8, 11.9, 14.0, 15.4, 16.8, 17.5\}$ (see Fig. 7).

While the amount of face context varied within the receptive field (see Fig. 6), the size of training patches were always scaled to 24 $\times$ 24 pixels. Thus in our approach there is a context versus resolution trade-off — larger receptive fields mean less resolution, while smaller receptive fields mean greater resolution.

Performance that resulted in both high detection rates and low localization error was obtained by using relatively large receptive fields that include a significant amount of the face (see Fig. 7). This result, which is somewhat

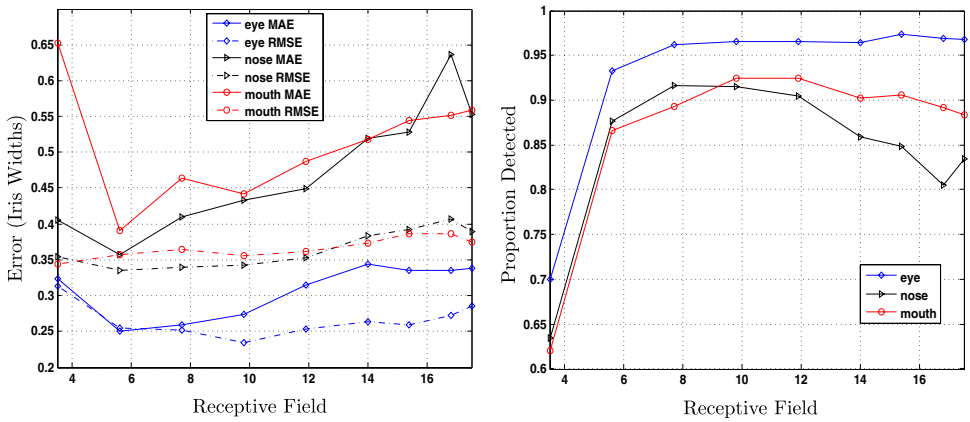


Fig. 7. Effect of receptive field size.

unintuitive, replicates our previous results on a smaller dataset.[9] It should be pointed out that other approaches typically use much smaller receptive field sizes, typically about 3 SIDs.[3,6,19,25]

4.4. *Sampling rate*

Because of the sliding-window approach, performance is influenced by the distance between each application of the detector window. We refer to this distance as s for *sampling distance*. Given a ROI for a particular target feature, we first choose the minimum $K \times K$ scale allowed by the ROI and applied the classification window at every point on a grid within the ROI whose nodes are spaced every $s = \frac{K}{24}$ pixels, rounded to the nearest integer. K is then incremented to the next positive integer multiple of 24 and the process is repeated, until K is larger than the maximum specified by the ROI. In practice, the output of the feature detectors is sensitive to translations and scales smaller than this base resolution. Since the true location of a FFP may be between the regions scanned by the sliding window, we tried increasing the sampling rate by a factor of 2, i.e. for a particular scale K choose windows spaced $\frac{s}{2}$ pixels apart. Table 1 shows the effect of increasing the sampling rate: doubling the sampling rate reduced error rates by a few percentage points for the nose and mouth detectors. However, the eye detectors improved by less than 1%. Throughout the remainder of experiments we sampled every $\frac{s}{2}$ pixels.

4.5. *Training set size*

We investigated the effects of the training set on performance. The number of positive examples was varied from 100 to 20,000 according to the following schedule: $\{100, 300, 500, 1000, 2000, 3000, 5000, 8000, 10{,}000, 15{,}000, 20{,}000\}$. For each condition, the number of negative examples was three times the number of positive examples.

Table 1. Performance as a function of sampling rate.

	MAE		RMSE		Detection Rate	
Feature	$\frac{s}{2}$	s	$\frac{s}{2}$	s	$\frac{s}{2}$	s
Left eye	0.2290	0.2780	0.2338	0.2573	0.9644	0.9623
Nose	0.3689	0.3977	0.3164	0.3341	0.9185	0.9093
Mouth	0.3391	0.5258	0.3311	0.3849	0.8992	0.8638

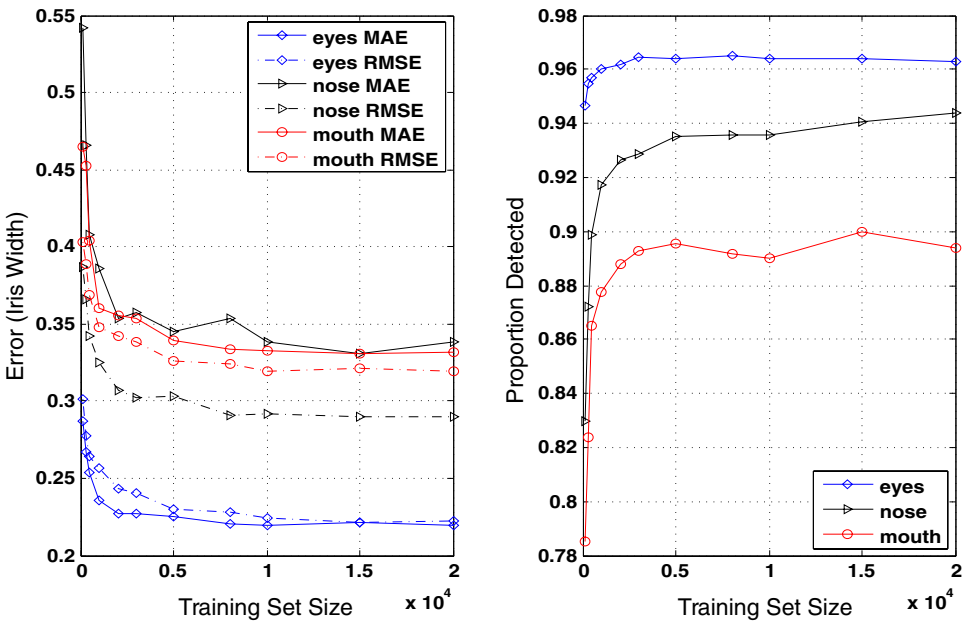
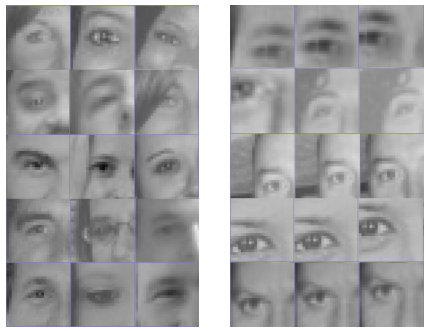


Fig. 8. Effect of training example set size.

Results are presented in Fig. 8. While there are some performance gains for training sizes beyond 5000 examples, they are minimal. Across all feature types, training set sizes between 1000 and 5000 examples perform well. This was a surprisingly small number considering that improvements made in other problems, such as face detection and smile recognition, required increasing the number of training examples beyond $10,000$.[24,26]

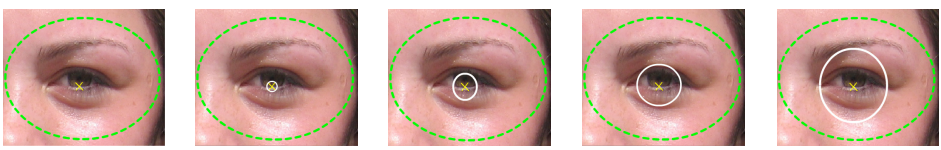
4.6. *Selection of positive and negative examples*

The training patches were chosen using the sliding-window described in Sec. 4.4. For each image in the training set we first create a "candidate list" of all locations visited by the detection window, restricted to the region of interest (shown by the green dashed outer ellipse in Fig. 10). Thus the scanning procedure to select training patches is identical to the scanning procedure used at run-time to detect target features. Once we have a collection of candidate patches, we select the patch



(a) (b)

Fig. 9. Training examples for receptive field of size 7.7 SIDs. Exclusion range of 2 with a bias selection to patches nearer the ROI boundary: (a) positive training examples, (b) negative training examples.



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

Fig. 10. Left eye region of interest: (a) no exclusion region, (b) exclusion region of 0.5, (c) exclusion region of 1, (d) exclusion region of 2 and (e) exclusion region 3.

with minimum Euclidian distance from the human labeled feature point (yellow X in Fig. 10), and add it to the set of positive training patches. This ensures that the system is trained with positive examples that have the FFP centered and slightly shifted from center with respect to location and scale in a manner similar to how they will be encountered at run-time. This is particularly important since patches visited in the sliding-window approach are sometimes separated by several pixels, depending on the size of the face.

To select negative examples, we first create an exclusion region (white inner ellipse in Fig. 10) around the labeled feature location and take a sample of patches inside the ROI but outside of the exclusion region. In addition we biased the sampling process in three ways: (a) no bias, (b) bias towards selection, of patches close to the exclusion boundary, and (c) bias towards selection of patches nearer to the ROI boundary. Experimental results indicate method (c), in conjunction with a moderate exclusion range [Fig. 10, examples (c) and (d)], performed best with regard to detection and localization accuracy.

4.7. *Number of training rounds*

Boosting is a sequential machine learning procedure, in which a new filter is added to the classifier after each round of training. The amount of training, i.e. the number

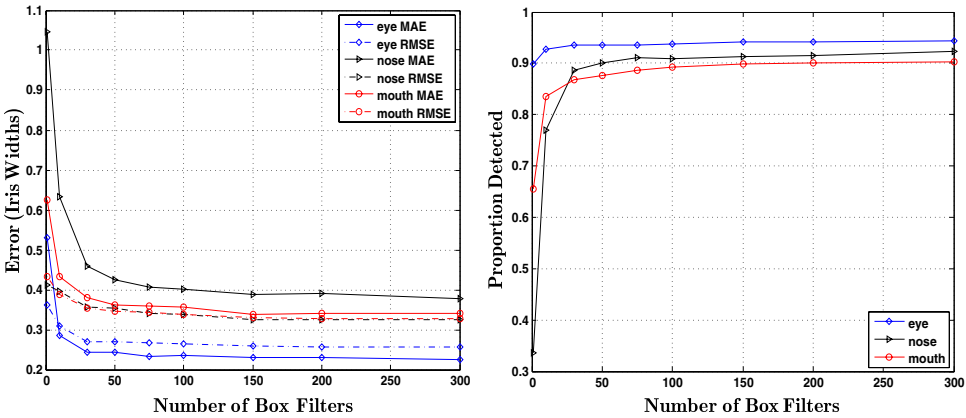


Fig. 11. Performance with respect to the number of box filters chosen to construct the strong classifier.

of filters chosen to construct a classifier, influences training time, run-time, detection rate and localization accuracy. We investigated the effect of varying the number of training rounds on classification performance. Experimental results (see Fig. 11) show best levels of performance occur between 30 and 150 training rounds, with only nominal improvements in performance beyond that. Other problems, like face detection, typically require several thousand training rounds,[24] indicating that feature detection in the context of faces is a less complex problem. Based on these results, we limited the rounds of training to 150 for all subsequent experiments.

4.8. *Information integration*

For each image patch visited by the multiscale sliding window approach, the detector returns the log-likelihood ratio that the patch was generated by the target class versus the background class. The result is a list of candidate patches that can be assessed in terms of the likelihood that they are of the target class. The log-likelihood ratio for a particular feature can be combined with the prior probability that the feature is located at a particular location to form a posterior probability estimate at every location.

We investigated several simple methods, which perform surprising well, for integrating the results of the likelihoods and posterior probability estimates of detected patches to make a final decision about the FFP location: (1) choosing the maximum log-likelihood patch; (2) choosing the maximum log-posterior patch; (3) choosing the mean of the k highest log-likelihood patches; (4) choosing the mean of the k highest log-posterior patches; (5) choosing the median of the k highest log-likelihood patches; (6) choosing the median of the k highest log-posterior patches; (7) choosing the weighted average of all log-likelihood patches; (8) choosing the weighted average of all log-posterior patches. In preliminary tests values of k between 20–25 gave best performance, so for these experiments we report results for $k = 23$.

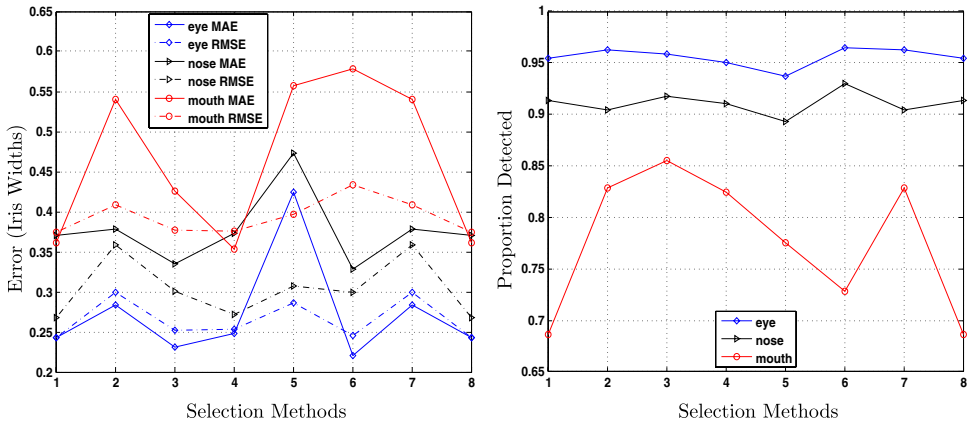


Fig. 12. Information integration methods: (1) maximum log-likelihood; (2) maximum log-posterior; (3) mean of the k highest log-likelihoods; (4) mean of the k highest log-posteriors; (5) median of the k highest log-likelihoods; (6) median of the k highest log-posteriors; (7) weighted average of all log-likelihoods; (8) weighted average of all log-posteriors. In these graphs, $k = 23$.

Figure 12 shows the results. The eyes and nose perform best by taking the mean of the top $k = 23$ highest log-likelihood patches (method (3) above). Based on our results we use method (3) for eyes and nose information integration and method (4) for mouth information integration for the remaining experiments.

4.9. *Pose invariance*

Accurate eye localization is an important step for many approaches to AU detection, face recognition and pose estimation.[5,7,20,27] The difficulty of face detection and subsequent emotion or face recogniton as pose deviates from frontal view is well known and still an open problem within the computer vision community.[5,7,9,21] If facial features are to be used as part of the overall process of image registration it is important to understand how localization performance varies over head pose. To better understand the relationship between head pose and feature location accuracy we studied the performance of feature localization as a function of head pose (roll, pitch and yaw).

Experiments were conducted on five grouped head pose ranges: $(-5°, 5°)$, $(\pm 5°, \pm 15°]$ and $(\pm 15°, \pm 60°]$ measured from frontal view for roll, pitch and yaw. Each grouping contained 150 images selected from the GENKI dataset (see Fig. 13). Experiments demonstrate that face detection becomes fragile beyond $\pm 15°$, with respect to roll and pitch. Detection of the tip of the nose and mouth were quite sensitive to pose variations in roll and yaw. However, given the face is detected, eye detection and localization proved to be robust to pose variation up to $\pm 60°$ in roll, yaw and pitch.

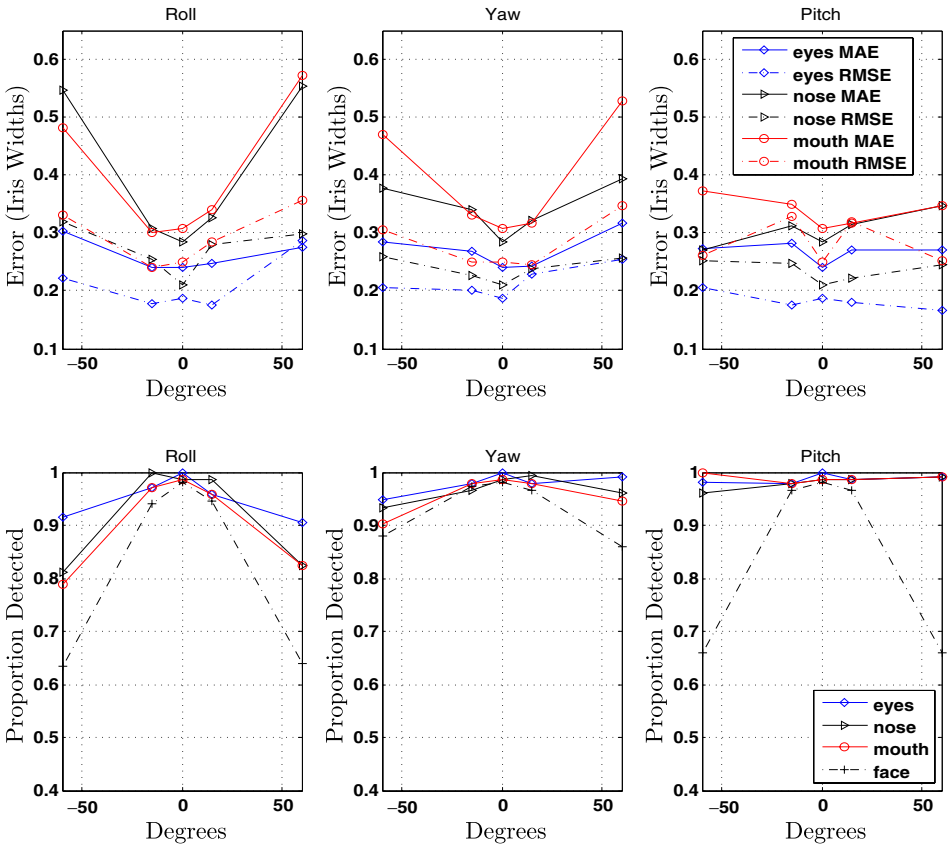


Fig. 13. Effects of pose on performance.

5. Prior Work

We compared the performance of our system with those systems reporting the best performance with respect to detection rate and localization error published to date,[3,6,8] including a previous version of our system.[9]

5.1. *Adaboost with active appearance model*

Cristinacce *et al.*[6] present results on an approach similar to the one presented here: a Viola–Jones style face detector is used to locate faces in the image plane. Once a face is localized, a similar style detector is applied to selected regions of the face — resulting in a set of candidate feature points. This is followed by the application of active appearance models (AAM) to the candidate points to refine the final location estimate, and infer missing points. Because of the significance of Ref. 6, we felt it important to compare our work with theirs. Unfortunately, we do not have access to the XM2VTS dataset. After consultation with Dr. Cristinacce we agreed that the

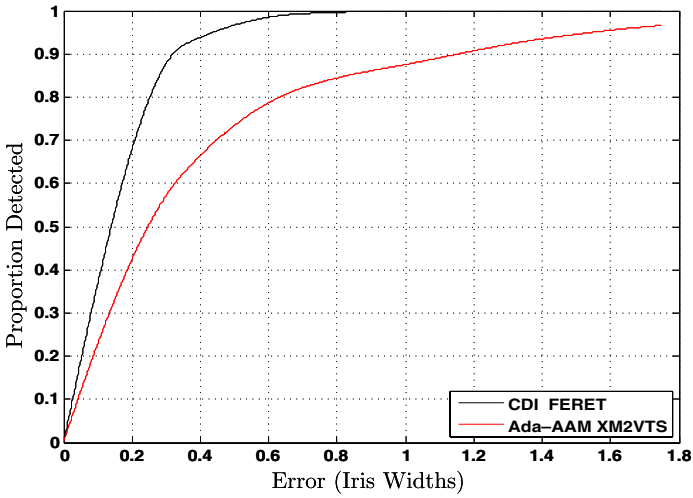


Fig. 14. Comparison between the Appearance Model of Ref. 6, indicated by Adaboost-AAM, and the current system, labeled CDI for Context Dependent Inference. Performance measured on the XM2VTS database by Ref. 6 and measured on the FERET database for this system.

FERET and XM2VTS data sets were of similar level of difficulty, as they are both frontal data sets taken under controlled illumination conditions. One difference is that the XM2VTS data set contains more images of persons with beards, which could affect performance for features on the lower part of the face. With this caveat in mind, Fig. 14 shows a comparison with results presented in Ref. 6.[a]

5.2. *Multimodule SVM*

Campadelli *et al.* presented a general-to-specific model for eye detection that can be applied to the output of any face detector that returns a rough estimate of the face location.[4] Once the face has been detected, eye localization is performed in a two step process by Support Vector Machine (SVM) modules: (1) the eye detector and (2) the eye localizer. The first SVM module performs a rough estimation of eye location by evaluating a subset of points in the face region. The second SVM module is then applied to the candidate points from the first step to refine localization accuracy. The authors of Ref. 4 provided us with their most current, yet to be published, performance data. Figure 15 shows performance comparisons on the FERET and BioID databases.[b]

[a]Results shown for Ref. 6 were extracted from published graphs, thus the exact numerical values may differ slightly.
[b]Results shown for Ref. 4 were obtained from the authors and have yet to be published in other literature.

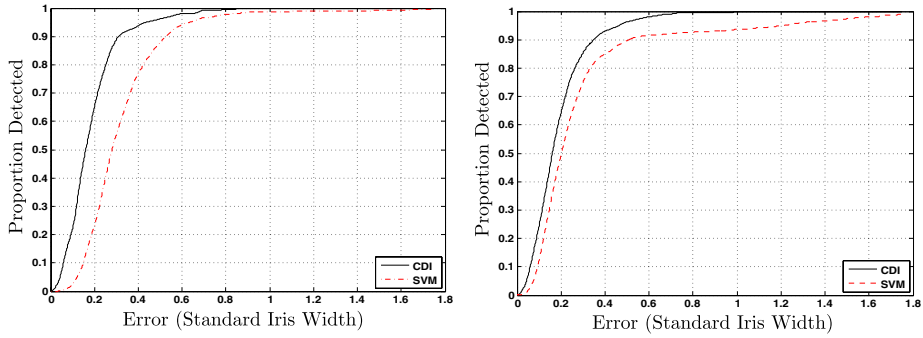


Fig. 15. Comparison with multimodule SVM of Ref. 3, indicated by SVM, and the current system, labeled CDI. Performance is given for FERET database (left) and BioID database (right).

5.3. *Bayesian, regression and discriminative classifier*

Everingham *et al.*[8] compare three approaches to eye localization: regression, Bayesian and discriminative classification. Their results indicate that the Bayesian approach performs best, detecting 90% of the eyes within approximately 0.329 SIDs. A recent field survey[5] described this level of performance as "remarkable" considering the state of the art in 2007. The system we are presenting here localized 92% of the eyes within 0.329 SIDs. While in absolute values, a 2% improvement may appear small, it represents a 20% reduction of error rate over.[8] Figure 16 shows a comparison of our system with the results in Ref. 8 on the FERET database.[c]

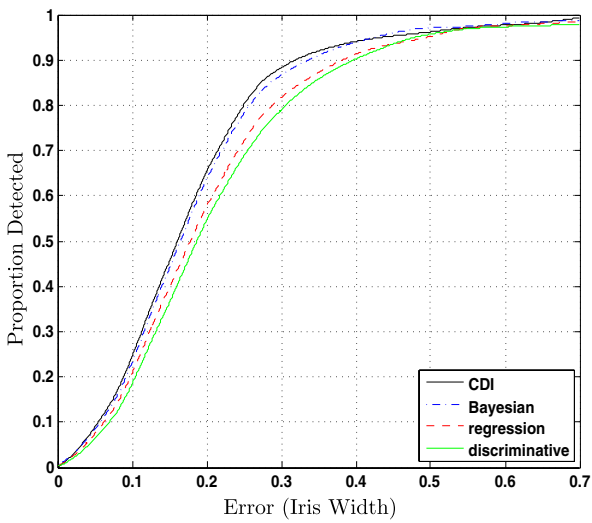


Fig. 16. Comparison with regression, discriminative classifier and Bayesian approach presented by Ref. 8, indicated by regression, discriminative and Bayesian figure labels, and the current system, labeled CDI for Context Dependent Inference. Performance given for the FERET database.

[c]Results shown for Ref. 8 were extracted from graphs provided by the author.

5.4. *Previous version of context dependent inference*

The current approach to feature detection was based on an eye detection system we had previously developed in Ref. 9. We compare performance of the current system to Ref. 9 on the FERET, BioID and GENKI databases. "CDI-0" indicates[9] and "CDI-1" indicates our current system. Figure 17 shows that the current system is a significant improvement over the previous system. On FERET, we detect 94.0% of eyes within 0.4 SIDs, while the previous system only detected 74.8% at this level. On BioID, we detect 92.8%, within 0.4 SIDs while the previous system only detected 70.9% within this distance. On the GENKI database, we detect 75.9% within 0.4 SIDs, while the previous system manages only 56.2% within the same distance.

5.5. *Comparison with human performance*

In the previous sections we compared performance of our automated system using the human labels as ground truth. However, since the human labels used for training

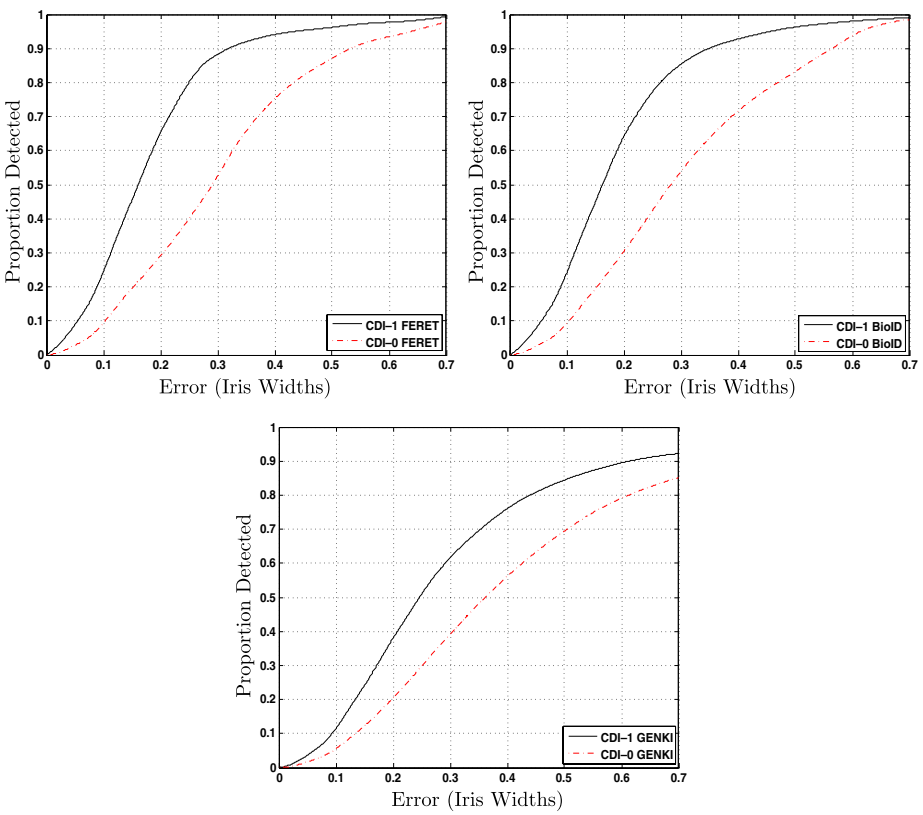


Fig. 17. Comparison with Ref. 9, indicated by CDI-0, with the current work, indicated by CDI-1. Performance given for FERET database (top left), the BioID database (top right) and the GENKI database (bottom center).

Table 2. Human versus machine: MAE.

Feature	Human	Machine	Machine/Human
Left eye	0.1159	0.1867	1.61
Nose	0.1596	0.3138	1.97
Mouth	0.1225	0.3908	3.19

Table 3. Human versus machine: RMSE.

Feature	Human	Machine	Machine/Human
Left eye	0.0748	0.1508	2.02
Nose	0.1037	0.3908	3.77
Mouth	0.0789	0.7495	9.50

may be noisy, it is unclear whether the system is already achieving the best possible performance given the quality of the labels. To clarify this issue, seven people hand labeled the eyes, nose and mouth in 100 randomly selected images from the GENKI database. This test set was also labeled automatically by our feature detection system. The average value, for each feature, computed from the seven human labels was deemed the ground truth value for the feature. We compare both the machine and individual human labelers error from the ground truth label. Tables 2 and 3 display these results. Human labelers outperformed the automated system by a factor of about 1.6 for the eyes, a factor of 2 for the nose, and a factor of 3 for the mouth. While this indicates that there is room for improvement, the performance of the eye detectors is already remarkably close to that of humans.

6. Conclusions

Robust and accurate localization of facial features is critical for an emerging generation of practical applications of machine perception technology applied to the human face. A key challenge in feature detection systems is the need to address an inherent trade-off between robustness and precision. Detectors that are robust to variations in illumination and imaging conditions tend to provide poor localization. Detectors trained to localize features precisely tend to produce a large number of false alarms.

Here, we address this tradeoff by refining the context dependent inference architecture previously proposed in Ref. 9. First, robust detectors are used to detect the general context in which features appear, and then precise detectors are used that operate within that context.

The approach explored in this document outperformed previous methods[3, 6, 8, 15] in terms of detection and localization accuracy. Our experience with a new database of images from the Web leads us to believe that the current benchmark databases used in the literature are too easy and no longer useful for assessing performance in the challenging situations needed for many practical real-world applications. To help drive the field and facilitate comparison with our work, we have released a

dataset (GENKI-4K) of 4000 images from the Web with corresponding ground truth feature labels.

We presented empirical studies of the different factors affecting performance within this architecture. These experiments showed: (1) Significant performance levels can be achieved with 1000–5000 training examples. (2) The negative examples within the ROI should be highly dissimilar from the positive examples. (3) The optimal receptive fields are relatively large, from 7.7 to 9.8 SIDs. (4) Evaluating detectors on test images at a higher sampling rate than the resolution of the detector reduces error. (5) A high degree of accuracy can be achieved with only 50 to 100 box filters. (6) Careful integration of the outputs of detectors over the ROI provides significant performance gains. (7) Eye detectors are relatively robust to pose variation, maintaining good performance levels with deviations from frontal pose of up to 60°. (8) The performance levels of the eye detectors approximate human levels of accuracy, and are ready for practical applications.

Acknowledgments

The study was funded by the UC Discovery Grant #10202 and by the NSF Science of Learning Center grant SBE-0542013.

References

1. The BANCA Database. http://www.ee.surrey.ac.uk/CVSSP/banca/.
2. The BioID Database. http://www.bioid.com/downloads/facedb.
3. P. Campadelli, R. Lanzarotti and G. Lipori, Percise eye localization through a general-to-specific model definition, *Proc. British Machine Vision Conf.* (2006).
4. P. Campadelli, R. Lanzarotti and G. Lipori, Automatic facial feature extraction for face recognition, *Face Recognition* (Publisher I-Tech Education, 2007).
5. P. Campadelli, R. Lanzarotti and G. Lipori, Eye localization: a survey, *The Fundamentals of Verbal and Non Verbal Communication and the Biometrical Issues*, NATO Science Series, Vol. 18 (Amsterdam, 2007).
6. D. Cristinacce and T. Coots, Facial feature detection using adaboost with shape constraints, *Proc. British Machine Vision Conf.* (2003).
7. A. Pnevmatikakis, E. Rentzeperis, A. Stergiou and L. Polymenakos, Impact of face registration errors on recognition, *3rd IFIP Conf. Artificial Intelligence Applications & Innovations (AIAI)* (2006).
8. M. Everingham and A. Zisserman, Regression and classification approaches to eye localization, *Proc. 7th Int. Conf. Automatic Face and Gesture* (2006).
9. I. Fasel, B. Fortenberry and J. Movellan, A generative framework for real time object detection and classification, *Comput. Vis. Imag. Underst.* **98** (2005) 182–210.
10. I. R. Fasel, M. S. Bartlett and J. R. Movellan, A comparison of Gabor filter methods for automatic detection of facial landmarks, *Proc. 5th Int. Conf. Automatic Face and Gesture Recognition* (Washington DC, 2002).
11. Y. Freund and R. E. Schapire, Experiments with a new boosting algorithm, *Proc. 13th Int. Conf. Machine Learning* (Morgan Kaufmann, 1996), pp. 146–148.
12. J. Friedman, T. Hastie and R. Tibshirani, Additive logistic regression: a statistical view of boosting, *Ann. Statist.* **28**(2) (2000) 337–374.
13. The GENKI-4K Database. http://mplab.ucsd.edu.

14. J. Huang and H. Wechsler, Eye detection using optimal wavelet packets and radial basis functions (RFFs), *Int. J. Patt. Recogn. Artif. Intell.* **7**(13), 1999.
15. O. Jesorsky, K. J. Kirchberg and R. W. Frischholz, Robust face detection using the hausdorff distance, *Proc. Audio and Video Based Person Authentication* (2001).
16. B. H. Juang, L. R. Rabiner and J. G. Wilpson, On the use of bandpass liftering in speech recognition, *IEEE Trans. Acoustics, Speech, Signal Proc.*, SSSP-**35**(7) (1987) 947–954.
17. R. Kothari and J. Mitchell, Detection of eye locations in unconstrained visual images, *ICIP96* (1996).
18. T. Leung, M. Burl and P. Perona, Finding faces in cluttered scenes using random labeled graph matching, *5th Int. Conf. Computer Vision* (1995).
19. Y. Ma, X. Ding, Z. Wang and N. Wang, Robust precise eye location under probabilistic framework, *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition* (2004), pp. 339–344.
20. M. S. Barlett, G. Littlewort, M. Frank, C. Lainscsek, I. Fasel and J. Movellan, Fully automatic facial action recognition in spontaneous behavior, *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition* (2006).
21. Q. Ji and P. Wang, Learning discriminant features for multi-view face and eye detection, *Computer Vision and Pattern Recogniton* (2005).
22. P. J. Phillips, H. Wechsler, J. Huang and P. Rauss, The FERET database and evaluation procedure for face recognition algorithims, *Imag. Vis. Comput.* **16**(5) (1998) 295–306.
23. G. Shakhnarovich, P. Violaand and B. Moghaddam, A unified learning framework for real-time face detection and classification, *Int. Conf. Automatic Face and Gesture Recognition* (2002).
24. P. Viola and M. Jones, Rapid object detection using a boosted cascade of simple feature, *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2001).
25. P. Wang, M. Green, Q. Ji and J. Wayman, Automatic eye detection and its validation, *IEEE Conf. Computer Vision and Pattern Recognition* (2005).
26. J. Whitehill, M. Bartlett, G. Littlewort, I. Fasel and J. Movellan, Developing a practical smile detector, *Proc. IEEE Int. Conf. Automatic Face and Gesture Recognition* (2008).
27. J. Whitehill and J. R. Movellan, A discriminative approach to frame-by-frame head pose estimation, *Int. Conf. Automatic Face and Gesture Recognition* (2008).
28. L. Wiskott, J. M. Fellous, N. Krüger and C. von Der Malsburg, Face recognition by elastic bunch graph matching, *IEEE Trans. Patt. Anal. Machine Intelligence* **19**(7) (1997) 775–779.
29. The XM2VTS Database. http://www.ee.surrey.ac.uk/Research/VSSP/xm2vtsdb.
30. S. Lucey, Y. Wang and J. Cohn, Enforcing convexity for improved alignment with constrained local models, *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)* (2008).
31. A. L. Yuille and H. H. Bulthoff, Bayesian decision theory and psychophysics, Technical Report 2, Max Planck Institut fur Biologische Kybernetik, Arbeitsgruppe Bulthoff (1993).
32. Z. H. Zhou and Y. Jiang, Projection functions for eye detection, *Patt. Recogn. J.* (2004).

Micah Eckhardt received his B.S. at UC San Diego in cognitive science with minors in computer science and mathematics. After graduation he spent two years as a research associate with the Machine Perception Laboratory at UC San Diego, where he worked on machine vision and social robotics. He is currently a graduate student at the Massachusetts Institute of Technology.

His current research investigates the integration machine vision and other technologies as assistive and learning tools for people with autism.

Ian Fasel joined the Computer Science Department at the University of Arizona in 2009.

In 2007–08, Dr. Fasel was a postdoctoral fellow at the University of Texas at Austin in the laboratory of Prof. Peter Stone, where he worked on reinforcement learning and human-teachable artificial agents. During that time he also held appointments at Osaka University where he worked on developmental robots with Profs. Minoru Asada and Hiroshi Ishiguro. Dr. Fasel received his Ph.D. in cognitive science at the University of California, San Diego, in the Machine Perception Lab under Dr. Javier Movellan and Dr. Marian Bartlett. He received a B.S. in electrical engineering and a B.A. in Plan II Honors Liberal Arts, with highest honors, from the University of Texas at Austin in 1999, where his thesis was chosen as a "model thesis".

His primary research is in machine learning for vision, audio, touch, cognitive and developmental robotics, and human-robot and human-computer interaction.

Javier Movellan received his Ph.D. at UC Berkeley in developmental psychology, in 1989. He was a Research Associate at Carnegie Mellon University from 1989–1993, where he pursued research on machine learning. Javier is currently a research professor at the California Institute of Telecommunications and Information Technology at the University of California San Diego, where he leads the Machine Perception Laboratory.

He is currently focusing on the problem of developing robots that learn to perceive and interact with the physical and social world. Javier is a founder member of Machine Perception Technologies Inc., a company whose goal is to apply machine perception technologies to daily life environments.