A Generative Framework for Real Time Object Detection

Javier R. Movellan, Bret Fortenberry & Ian Fasel
(MPLab TR 2003.03)
Machine Perception Laboratory
Institute for Neural Computation
University of California San Diego

Abstract

Recent imaging studies show that the human brain has structures specialized for the detection of eyes and the recognition of eye-related behavior. There is some evidence that such systems may be innate and play an important role in infant social development. The development of machine perception systems that detect eyes and analyze eye behavior will also enable new approaches to human-computer interaction that emphasize natural, face-to-face communication with the user. For such systems to have an impact in everyday life it is important for them to work robustly in natural, unconstrained conditions.

We formulate a probabilistic model of image generation and derive optimal inference algorithms for finding eyes within this framework. The approach models the image as a collage of patches of arbitrary size, some of which contain the object of interest and some of which are background. The approach requires development of likelihood-ratio models for object versus background generated patches. These models are learned using boosting methods. One advantage of the generative approach proposed here is that it makes explicit the conditions under which the approach is optimal, thus facilitating progress towards methods that model the image generation process in more realistic ways. The approach proposed here searches the entire image plane in each frame, making it resistant to fast, unpredictable motion. The system is robust to changes in lighting, illumination, and differences in facial structure, including facial expressions and eyeglasses. Furthermore, the system can simultaneously track the eyes and blinks of multiple individuals. We also present pilot results using this system for analysis of eye-openness in EEG studies. Finally we reflect on how the development of perceptive systems like this may help advance our understanding of the human brain.

1 Introduction

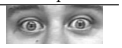
Since its official beginnings in the early 1960s Cognitive Science has fashioned many heated debates: early attention vs. late attention, working memory vs. short term memory, serial vs. parallel processing, analogical vs. propositional representations, symbolic vs. sub-symbolic processing, modular vs. interactive architectures. These debates have turned out undecidable, contributed little to our understanding of the mind, and have not proven relevant for society at large.

The need for methodological reform is clear. Modern approaches and methods to the study of the mind are needed that avoid scholastic debates. One approach, which we have found particularly useful was originally proposed by Marr (1982). The approach focuses on understanding the nature of the problems the brain faces and finding possible solutions to these problems (Edleman and Vaina, 2001). When pursuing this endeavor we have found that probability theory, in particular the use of probabilistic generative models, was a fruitful analytical tool. The third author of this paper referred to this methodological stance as *probabilistic functionalism* (Movellan and Nelson, 2001). One characteristic of probabilistic functionalism is the focus on solving specific problems under general conditions rather than solving abstract problems under restricted laboratory conditions. To focus simultaneously on the specificity of the problem and the generality of the solution is critical, otherwise one can easily get caught on frustrating theoretical debates or on trick solutions that inform us little about the brain. This document can be seen as an application of the methods of functional probabilism to help understand the problem of eye and eye-blink detection. We do so by formulating an analytical model of the problem at hand, studying how optimal inference would proceed under such model, and evaluating the performance of the optimal inference algorithm in natural conditions.

The study of face perception has been revitalized thanks to recent progress in cognitive neuroscience. The advent of modern neuro imaging is revolutionizing the study of the mind and presenting a picture of the human brain far different from a general purpose computing machine. Single neuron recording and imaging studies are showing specific neural systems that play a crucial role in the perception of faces, facial features, and facial expressions. These include the fusiform face area, superior temporal sulcus, orbital frontal cortex, frontal operculum, right somatosensory cortex, and the amygdala (Kawashima et al., 1999; George et al., 2001).

Face perception has been a traditional area of research in developmental psychology, a discipline that studies how the human mind develops from infancy to adulthood. Face processing in general and eye detection in particular is deemed so important in this field that some of its most influential researchers have postulated the need for innate eye detection and gaze processing modules. These ideas are still controversial but recent experiments have shown that from birth human infants are exceptionally sensitive to the eye and to mutual gaze engagement (Farroni et al., in press; Johnson, 2001). These systems may help tune the newborn infant towards interaction with their caregivers (Baron-Cohen, 1995).

In recent years there has been an emerging community of machine perception scientists focused on automatic detection of faces and facial behavior. The special importance of the eyes is becoming quite clear within this community. There are at least two reasons for this: (1) Proper registration. In a recent evaluation of state of the art face recognition system it was proposed that a large proportion of the failures of these system was due to poor alignment and registration of facial features, particularly in outdoors conditions. Good eye detection in realistic environments may thus have a tremendous impact on the accuracy of face perception technolo-

Code	Descriptor	Muscles Involved	Example
AU5	Upper Lid Raiser	Levator Palpebrae Superioris	
AU6	Cheek Raiser	Orbicularis Oculi, Pars Orbitalis	
AU7	Lid Tightener	Orbicularis Oculi, Pars Palebralis	
AU41	Lid Droop	Relaxation of Levator Palpebrae Superioris	
AU42	Slit	Orbicularis Oculi	
AU43	Eyes Closed	Relaxation of Levator Palpebrae Superioris; Orbicularis Oculi, pars Palpebralis	
AU44	Squint	Orbicularis Oculi, pars Palpebralis	
AU45	Blink	Relaxation of Levator Palpebrae Superioris; Orbicularis Oculi, pars Palpebralis	
AU46	Wink	Relaxation of Levator Palpebrae Superioris; Orbicularis Oculi, pars Palpebralis	
AU61	Eyes Turn Left	Lateral and Medial Rectus	
AU62	Eyes Turn right	Lateral and Medial Rectus	
AU63	Eyes Up	Superior Rectus	
AU64	Eyes Down	Inferious Rectus	
AU65	Walleye	Lateral Rectus	
AU66	Crosseye	Medial Rectus	

gies (Phillips, 2003). (2) The Facial Action Coding System (FACS) of Ekman and Friesen (1978) is the most comprehensive standard for coding facial behavior. FACS devotes 15 categories (action units) to describe eye behavior (see Table 1). Only the mouth surpasses the eyes in the number of action units assigned to it. This reflects the fact that eye behavior is extremely rich and particularly informative about the state of the user.

Current work on eye detection divides into approaches based on visible spectrum cameras and approaches based on near-infra-red (NIR) cameras. In indoor and relatively controlled conditions the spectral properties of pupil under near NIR illumination provide a very clean signal that can be processed very fast and accurately (Haro et al., 2000; Ji and Yang, 2001, 2002). While NIR based methods are practical and worth pursuing, it is also important to pursue visual spectrum methods for the following reasons: (1) NIR based methods tend to produce a large number of false positives when used in relatively uncontrolled illumination conditions; (2) NIR based methods do little to further our understanding about the perceptual problem the brain solves when processing faces in natural conditions.

Of all the eye related behaviors perhaps the most important is blinks (action unit 45 in the FACS system). This is due to its relevance in several fields, including neurology, physiology, and psychology. For example, blink rate is known to vary with physiological and emotional arousal, cognitive effort, anxiety, fatigue, and deceit (Holland and Tarlow, 1972; Ekman, 1985; Karson, 1988; Van-Orden et al., 2000; Ji and Yang, 2001). Ji and Yang (2002) presents a state of the art method to detect blinks in real time using NIR imaging. Approaches based on visual spectrum images also exist. Bartlett, Braathen, , Littlewort, Smith, and Movellan (in press) present an approach to detect blinks in indoors environment using Support Vector Machines. Cohn, Xiao, Moriyama, Ambada, and Kanade (in press) describe an approach, that uses hand-coded eye-blink detectors. They report results comparable

to those of Bartlett et al. (in press) on the same testing dataset. These two systems handled out-of-plane rotations of the head by fitting a 3D deformable model of the head and then re-rendering the image into a frontal view.

2 A Generative Model for Images

In this section we frame the problem of finding faces and facial features as a Bayesian inference problem: We formulate a model of how images are generated and then derive an algorithm for making optimal inferences under this model. One advantage of generative models is that probability estimates of the categories of interest are computed explicitly, facilitating integration with other potential sources of information not necessarily considered at design time. In addition generate models force us to make our assumptions explicit, facilitating progress towards more effective algorithms.

Unless otherwise stated capital letters will represent random variables and small letters specific values taken by those variables. When possible we use informal shorthand notation and identify probability functions by their arguments. For example, $p(y)$ is shorthand for the probability (or probability density) that the random matrix Y takes the specific value y.

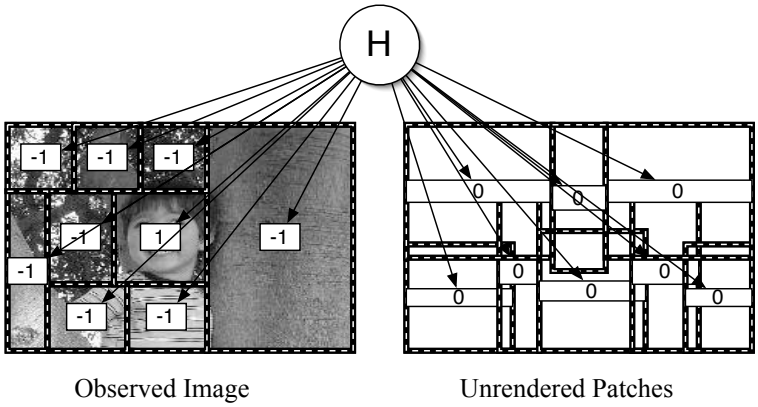


Figure 1: The hidden variable H determines which image patches will render the background (-1) which patches will render the object of interest (1) and which patches will not be rendered (0). The set of rendered patches determine the observed image.

We model the image as a collage of rectangular patches of arbitrary size and location, some patches rendering the object of interest, the other rendering the background. Given an image our goal is to discover the patches that rendered the object. Let Y a random matrix representing an image with a fixed number of pixels. Let y be a specific sample from Y. Let $\mathcal{A} = (a_1, a_2, \cdots, a_n)$ be an enumeration of all possible rectangular image patches, e.g. a_i determines the position and geometry of a rectangle on the image plane. Let y_{a_i} a matrix whose elements are the values of y for the pixels in the rectangle a_i. Let $H = (H_1, \cdots H_n)$ be random vector that assigns each of the n patches to one of three categories: H_i takes the value 1 when the patch a_i renders the object of interest, it takes value -1 when it renders the background, and value 0 when it is not rendered. (see Figures 1 and 2).

The image generation process proceeds as follows (see Figure 1). First a segmentation h is chosen with probability $p(h)$. Then for each patch a_i if $h_i = 1$ then an image of size a_i is chosen from the object distribution $q(\cdot \mid a_i, 1)$ independently of all the other patches. If $h_i = -1$ then a background image y_{a_i} is chosen from the background distribution $q(\cdot \mid a_i, -1)$. If $h_i = 0$ then a_i is not rendered. The observed image y is the collection of the rendered patches.

The model is specified by the prior probabilities $p(h)$ and by the object and background rendering distribution q. The prior is specified by the marginal probabilities $\{p(H_i = 1) \ : \ i = 1 \cdots n\}$, by the constraint that values of h that do not partition the image plane have zero probabilty, and by one of the two following constraints: (I) For cases in which we know there is one and only one object of interest on the image plane, only values of h with a single 1 are allowed. (II) For cases in which there may be an arbitrary number of objects of interest we assume that for $i = 1 \cdots n$ the distribution of $\{H_j \ : \ j \neq i\}$ conditioned on the event $\{H_i \neq 0\}$ is independent of H_i. In other words, we assume that the location of a rendered object does not tell us anything about the location of other rendered objects except for the fact that two different objects cannot render the same pixels.

For a given image y our goal is to detect patches rendered by the object. There are two cases of interest: (I) We know there is one and only one patch rendered by the object ; (II) There is an unknown and arbitrary number of patches rendered by the object model.

2.1 Case I: Single Object

We know there is one and only one patch on the image plane that rendered the object of interest. Our goal is to find the most probable patch $\hat{k} \in \{1 \cdots n\}$ given the image y, i.e,

$$\hat{k} = \operatorname*{argmax}_i p(H_i = 1 \mid y) \tag{1}$$

Using the law of total probability we have that

$$p(H_i = 1 \mid y) = \sum_h p(H_i = 1) p(h \mid H_i = 1) p(y \mid h H_i = 1) \tag{2}$$

Note that

$$p(y \mid h H_i = 1) = \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} Z(h, y) \tag{3}$$

where

$$Z(h, y) = \prod_{i : h_i \neq 0} q(y_{a_i}; a_i, -1) \tag{4}$$

The term $Z(h, y)$ describes how well the image y can be explained by the segmentation h but with all the patches rendering background, no objects. Thus

$$p(H_i = 1 \mid y) = p(H_i = 1) \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \tag{5}$$

$$\sum_h p(h \mid H_i = 1) Z(h, y) \tag{6}$$

$$= p(H_i = 1) \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} E(Z(H, y) \mid H_i = 1) \tag{7}$$

The conditioning event $\{H_i = 1\}$ discards all the partitions that do not render a_i. Thus $E(Z(H, y) \mid H_i = 1)$ represents how well the image y can be explained

as a mosaic of background patches, provided one of those patches is a_i. If the background model includes all possible wrongly shifted and scaled versions of the object of interest then $E(Z(H,y) \mid H_i = 1)$ should be small for the patch that actually rendered the object, and large otherwise. This is due to the fact that the patch that includes the object will be hard to explain by the background model (see Figure 2). More formally if $E(Z(H,y) \mid H_{\hat{k}} = 1) \leq E(Z(H,y) \mid H_i = 1)$ for $i = 1, \cdots n$ then

$$\hat{k} = \underset{i}{\operatorname{argmax}}\, p(H_i = 1 \mid y) = \underset{i}{\operatorname{argmax}}\, p(H_i = 1) \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \tag{8}$$

$$= \underset{i}{\operatorname{argmax}}\ \log p(H_i = 1) + \log \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \tag{9}$$

The optimal inference algorithm prescribes scoring all possible patches in terms of a function that includes the prior probability of that patch containing an object and a likelihood ratio term. The patch that maximizes this score is then chosen.

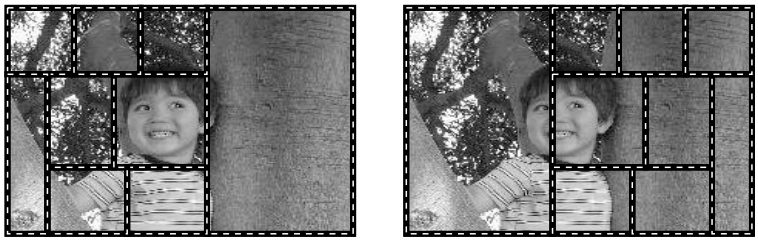


Figure 2: The segmentation on the right contains the patch that generated the object of interest (i.e. the face). It will be hard for this segmentation to explain the image as a collection of background patches. The segmentation on the left does not contain the object patch. Since the background model includes wrongly shifted version of faces it will be easy to explain the image as a collection of object patches.

2.2 Case II: Arbitrary Number of Objects

This case applies, for example, in face detection problems for which we do not know apriori how many faces may appear on the image plane. To formalize the problem we define a function Φ that measures the degree of match between any two arbitrary segmentations h and h'

$$\Phi(h, h') = \sum_{i=1}^{n} \rho(h_i, h'_i) \tag{10}$$

$$\rho(h_i, h'_i) = \delta_{h_i, h'_i} (\delta_{h_i, 1} + \delta_{h_i, -1}) \tag{11}$$

where δ is the Kroenecker delta function. Φ counts the number of patches for which both h and h' assign the same "object" or "background" label and disregards all the patches that are not rendered by h. Our goal is to find a partition $\hat{h}$ that optimizes the expected match

$$E(\Phi(H, \hat{h}) \mid y) = \sum_i p(h_i \mid y) \rho(h_i, \hat{h}_i) \tag{12}$$

The optimal assignment follows

$$\hat{h}_i = \begin{cases} 1 & \text{if } p(H_i = 1 \mid y) > p(H_i = -1 \mid y) \\ -1 & \text{else} \end{cases} \tag{13}$$

Thus to find the optimal assignment we need to scan all possible image patches $a_1 \cdots a_n$, compute the log posterior probability ratio

$$\log \frac{p(H_i = 1 \mid y)}{p(H_i = -1 \mid y)} \tag{14}$$

and assign "object" labels to the patches for which this ratio is larger than 0.

Using the law of total probability we have that

$$P(H_i = 1 \mid y) = \sum_h p(H_i = 1) p(h \mid H_i = 1) p(y \mid hH_i = 1) \tag{15}$$

where

$$p(y \mid hH_i = 1) = q(y_{a_i}; a_i, 1) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j) \tag{16}$$

Thus

$$P(H_i = 1 \mid y) = P(H_i = 1) q(y_{a_i}; a_i, 1) \sum_h p(h \mid H_i = 1) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j) \tag{17}$$

and

$$P(H_i = -1 \mid y) = P(H_i = -1) q(y_{a_i}; a_i, -1) \sum_h p(h \mid H_i = -1) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j) \tag{18}$$

Due to the fact that $\{H_j \ : \ j \neq i\}$ are independent of H_i given $\{H_i \neq 0\}$ then

$$p(h \mid H_i = 1) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j) = p(h \mid H_i = -1) \prod_{j \neq i : h_j \neq 0} q(y_{a_j}; a_j, h_j) \tag{19}$$

for all s, h, i. Thus

$$\log \frac{P(H_i = 1 \mid y)}{P(H_i = -1 \mid y)} = \log \frac{P(H_i = 1)}{P(H_i = -1)} + \log \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} \tag{20}$$

In order to make optimal inferences all we need is a model for the prior probability of object locations and a model for the log-likelihood ratios of image patches of arbitrary geometry. In Section 3 we will see how these models can be learned using boosting methods.

3 Learning Likelihood Ratios using GentleBoost

The inference algorithm presented above requires a likelihood ratio model for object versus background generated patches. Given an image patch y_i the model should give us the probability ratio of the patch given the object model vs background model. In this paper we learn these likelihood ratios using GentleBoost, a boosting algorithm recently developed by Friedman et al. (1998). Boosting (Freund and Schapire, 1996, 1999) refers to a recent family of machine learning algorithms for learning classifiers by sequential accumulation of experts that focus on the mistakes made by previous experts. Friedman et al. (1998) showed that boosting methods can be reinterpreted from the point of view of sequential statistical estimation. Based on this point of view they proposed "GentleBoost", a boosting algorithm that optimizes a "gentle" version of the binomial likelihood function.

During training we are given labeled examples of two categories and the goal is to learn a model for the log posterior probability ratios of the categories. The model used in GentleBoost is of the following form:

$$\log \frac{p(y)}{1 - p(y)} = 2 f(y) \tag{21}$$

where $p(y)$ is the probability that input y belongs to one of the two categories of interest, and

$$f(y) = \sum_{i=1}^{t} h_i(y) \tag{22}$$

adds the opinion of t experts. As explained in the next section, the "experts" used in our system consist of two elements: (1) a simple Haar-like wavelet; and (2) a non-linear tuning function that takes the output of the wavelet and produces an opinion about the category of the input. This opinion can take any real value between 1 and 1]. A value of-1 indicates that this wavelet is certain that y belongs to the background. A value of $+1$, indicates that the wavelet is certain the input belongs to the object of interest (see Figure 10).

GentleBoost can be seen as an application of Newton-Raphson optimization algorithm to minimize the following chi-square error:

$$\rho = \sum_i \frac{t(y_i) - p(y_i)}{\sqrt{p(y_i)(1 - p(y_i))}} \tag{23}$$

where $t(y) \in \{0, 1\}$ is the category label for the i^{th} training input y_i and

$$p(y) = \frac{1}{1 + e^{-2f(y)}} \tag{24}$$

Note $p(y_i)$ is the probability of a Bernouilli random variable with mean $p(y_i)$ and standard deviation $\sqrt{p(y_i)(1 - p(y_i))}$. Thus ρ can be seen as a the number of standard deviations between the observed label and the average label value. As the number of examples in the training set increases, minimizing the chi-square error becomes identical to maximizing the likelihood. However when the number of samples is small, chi-square estimators can be more efficient than maximum likelihood estimators.

GentleBoost chooses a set of experts $h_1, h_2 \cdots$ in a sequential manner. For a given set of already chosen experts, GentleBoost selects the expert that maximally reduces the current chi-square error. In practice this can be done in a variety of ways. We use the following method: We start with a very large pool of wavelets (see Section 5). For each wavelet we use kernel regression methods to estimate the function $h : \mathbb{R} \to [-1, 1]$ that minimizes ρ if that particular wavelet were added to the pool of already chosen wavelets. We call this function the *tuning curve* for the wavelet. After we found the optimal tuning curves for all the wavelets in the original pool, we choose the particular wavelet and corresponding tuning curve, that minimizes the current value of ρ. The process is iterated, each time adding a new wavelet and tuning curve, until ρ no longer decreases.

At the end of the training process, if we give the system an image patch y_{a_i} the model will provide us with an estimate of the probability that the patch belongs to one of the two categories of interest (i.e., object vs. background)

$$p(y_{a_i}) = \frac{1}{1 + e^{-2f(y_{a_i})}} \tag{25}$$

This posterior probability estimate reflects the proportion π of examples of a given category in the training sample. The inference algorithm in (27) requires log-likelihood ratios, not log-posteriors. These can be easily derived from (25) using Bayes rule

$$\log \frac{q(y_{a_i}; a_i, 1)}{q(y_{a_i}; a_i, -1)} = \log\left(\frac{1 - \pi}{\pi}\right) + \log\left(\frac{p(H_k = 1 \mid y_{a_k})}{p(H_k = -1 \mid y_{a_k})}\right) = \log\left(\frac{1 - \pi}{\pi}\right) + 2f(x) \tag{26}$$

Combining and (9) and (25) we get

$$\hat{k} = \max_i p(H_i = 1 \mid y) = \max_i \log p(H_i = 1) + 2f(y_{a_i}) \tag{27}$$

4 Situation Based Inference

One common approach to eye detection is based on the operation of a set of independent feature detectors (Huang and Wechsler, 1999; Fasel et al., 2000). The output of these detectors (e.g., a detector for the left eye, a detector for the right eye, a detector for the tip of the nose, etc.) is integrated by looking for configurations that match the distribution of interfeature distances typical of the human face (Wiskott et al., 1997; Leung et al., 1995; Kothari and Mitchell, 1996). Suppose our goal is to find the center of an eye with 1 pixel accuracy. This requires for our background model to include examples of eyes shifted by 1 pixel from the center position. In practice, a detector efficient at distinguishing eyes slightly shifted from center is also likely to produce a large number of false positives when scanning general backgrounds that do not include faces. Unfortunately in the approaches described above the search problem scales exponentially with the number of false alarms, rendering them impractical in situations with unconstrained background conditions.

The approach we propose here is based on the idea of a bank o situational or context dependent experts operating at different levels of specificity. For example, since the eyes occur in the context of faces, it may be easier to detect eyes using a very large context that include the entire face and then formulate feature detectors specifically designed to work well under such context. While we may think of these as face detector, we can also think of them as eye detectors that happen to have very large receptive fields. This form of eye detection works under very general context conditions, avoiding the proliferation of false alarms, but provides poor information about the precise location of the eyes. These eye detectors are complemented by context-specific eye detectors that provide very precise information about the position of the eyes.

More formally, let y represent an observed image, S represent a contextual situation (e.g., the location and scale of a face on the image plane), and O represent the location of the left eye of that face on the image. Using the law of total probability we have that

$$p(o \mid y) = \int p(s \mid y) p(o \mid sy) dh \tag{28}$$

where $p(s \mid y)$ represents a situation detector. In our case its role is to find regions in the image plane that are likely to contain eyes due to the fact that they contain faces. The $p(k \mid sy)$ is a situation based eye detector. For example it may work when the location and scale of the face on the image plane is known. In this example s partitions the image pixels into those belonging to the face, $y_f(s)$, and those belonging to the background . Once the position and scale of the face are known, the background provides no additional information about the position of the eye, i.e.,

$$p(o \mid ys) = p(o \mid y_f(s) s) \tag{29}$$

where $q(k|yh)$ is a model for face patches that are not eyes. The situational approach proposed here can be iterated, where one first detects a general context, followed by detection of a context within a context, each time achieving higher levels of precision and specificity allowed by the fact that the context models become smaller and smaller on each iteration.

5 Real-time system architecture

In the next sections we describe and evaluate an algorithm that performs optimal inference under the assumptions of the generative model described above. The current system utilizes two types of eye detectors: The first type, which can be thought of as a face detector, starts with complete uncertainty about the possible location of eyes on the image plane. Its role is to narrow down the uncertainty about the location of the eyes while operating in a very wide variety of illumination and background conditions. The second type of detector operates on the output of the first detector. As such it can assume a restricted context and achieve high location accuracy. Once the most likely eye location is chosen, the image patch surrounding the eyes is passed to a blink detection for analysis. The flowchart for this procedure is shown in Figure 3.

While the system described here operates on video images in real time, it currently treats each frame as independent of the previous frames, making it equally useful for static images as for video. Treating each video frame independently allows the system to simultaneously code eye location and behavior on multiple faces that may come in and out of the image plane at random times.

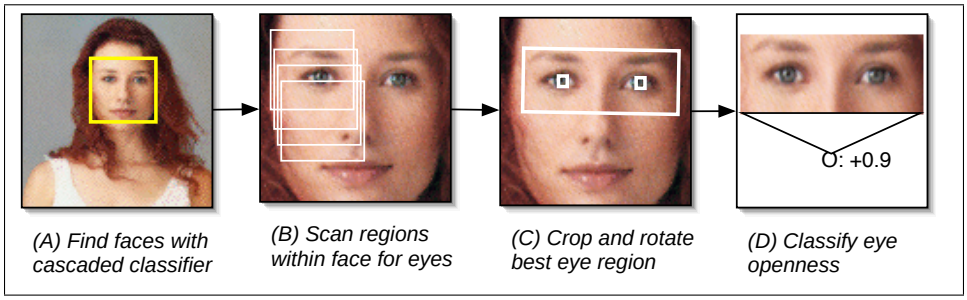


Figure 3: Flowchart for face, eye, and blink detection

5.1 Stage I: Eye detection in general background conditions

As described above the first component of the inference process locates regions of the image plane that contain faces, and thus eyes. This module operates under very general background and illumination conditions and greatly narrows down the plausible location eyes on the image plane.

We decide for the smallest face of interest to be 24×24 pixels large. We developed a likelihood-ratio model for this scale using a dataset of Web images provided by Compaq Research Laboratories. This dataset contains 5000 images of frontal upright faces scaled to fit a 24×24 pixels square. A mirror image of each patch was obtained for a total of 10000 face images. 8 billion examples of non-face patches were obtained by randomly selecting patches from a dataset of 8000 images were collected from a dataset of non-face images. All the patches were square, of arbitrary size, and at arbitrary locations in the images of the dataset. All the patches were then scaled down to 24×24 pixels.

The likelihood-ratio model was trained using the GentleBoost method described in Section 3. GentleBoost sequentially chooses wavelets from a large pool and combines them to minimize a chi-square error function. The pool of wavelets we choose from was based on Viola and Jones (2001); Shakhnarovich et al. (2002) and

consists of Haar-like wavelets. The main reason for their use is that their output can be computed very fast by taking the sum of pixels in two, three, or four equal-sized, adjacent rectangles and taking differences of these sums. To this original set we add a center-surround type wavelets and mirror image wavelets that are sensitive to patches symmetric about vertical axis (see Figure 5). It is very computationally expensive to perform an exhaustive search over all these wavelets– in a 24×24 pixel window, there are over 160,000 possible wavelets of this type. To speed up training, we break the wavelet selection step into two stages. First, at each round of boosting, we take a random sample of 5% of the possible wavelet. For each wavelet we find the tuning curve that minimizes the loss function ρ if that particular wavelet were added to the pool of already chosen wavelets. In step two, we refine the selection by finding the best performing single-wavelet classifier from a new set of wavelets generated by shifting and scaling the best wavelet by two pixels in each direction, as well as composite wavelets made by reflecting each shifted and scaled wavelet horizontally about the center and superimposing it on the original. Using the chosen classifier as the weak learner for this round of boosting, the weights over the examples are then adjusted using to the GentleBoost rule. This wavelet selection process is then repeated with the new weights, and the boosting procedure continues until the performance of the system on a validation set no longer decreases.

The inference algorithm calls for likelihood ratio models at multiple scales. Likelihood ratios for larger image patches are obtained by linearly scaling the patches down to 24×24 pixels and then applying the likelihood ratio model trained on that particular scale. Thanks to the choice of Haar-like wavelets for the higher level image representation, this interpolation step can accomplished in constant time if the scale factor is an integer.

Following Viola and Jones (2001), rather than training a "monolithic" classifier which evaluates all its wavelets before it makes a decision, we divided the classifier into a sequence of smaller classifiers which can make an early decision to abort further processing on a patch if its likelihood-ratio falls below a minimum threshold. We can think of this as a situational cascade where each level of the cascade is trained only on patches that survived the previous levels. After each element of the cascaded is trained, a boot-strap round (*ala* Sung and Poggio (1998)) is performed, in which the full system up to that point is scanned across a database of non-face images, and false alarms are collected and used as the non-faces for training the subsequent strong classifier in the sequence. Figure 10 shows the first two wavelet chosen by the system along with the tuning curves for those wavelets.

At recognition time the inference algorithm calls for scanning the entire image plane and looking for square patches of arbitrary scale and location with large likelihood-ratios. In practice we start scanning 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 1.2 times the previous scale, and the corresponding offsets are scaled by the same proportion, for an additional $(n - 24s) \times (m - 24s)/s$ patches per scale. For a 640×480 pixel image, this produces over $400,000$ total patches.

Because the early layers in the cascade need very few wavelets to achieve good performance (the first stage can reject 60% of the non-faces using only 2 wavelets, using only 20 simple operations, or about 60 microprocessor instructions), the average number of wavelets that need to be evaluated for each window is very small, making the overall system very fast while still maintaining high accuracy. Performance on the CMU-MIT dataset (a standard, public data set for benchmarking frontal face detection systems) is comparable to Viola and Jones (2001). While

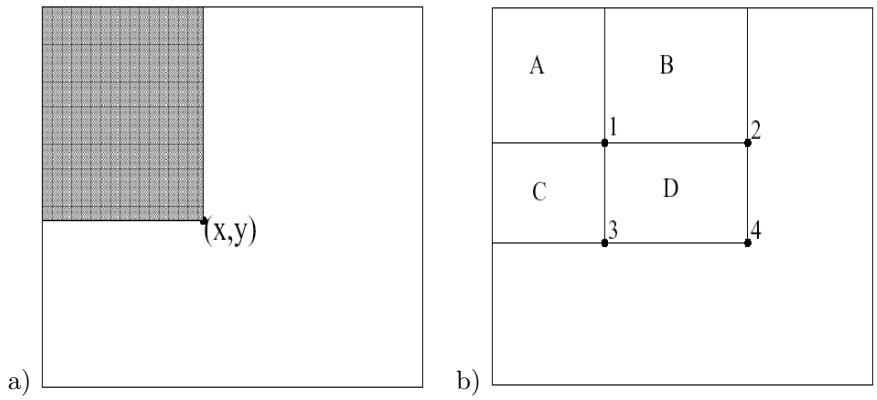


Figure 4: The Integral Image (after Viola & Jones, 2000).: (a) The value of the pixel at (x, y) is the sum of all the pixels above and to the left. (b) The sum of the pixels within rectangle D can be computed as $4 + 1 - (2 + 3)$.

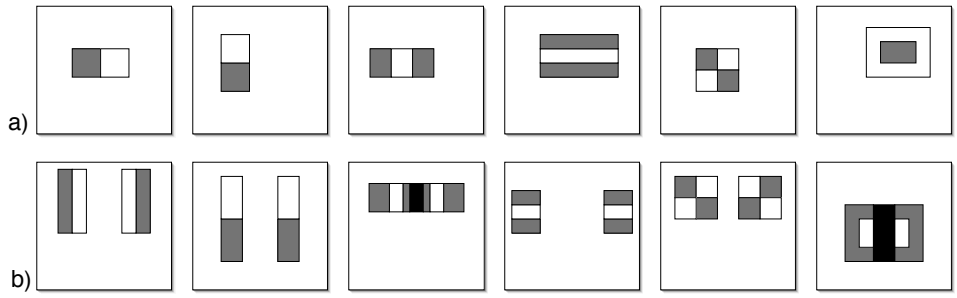


Figure 5: Each wavelet is computed by taking the difference of the sums of the pixels in the white boxes and grey boxes. (a) Wavelets types include those in (Viola and Jones, 2001), as well as a center-surround type wavelet. (b) During the refinement step, the same wavelet types superimposed on their reflection about the Y axis are also possible.

CMU-MIT contains wide variability in the images due to illumination, occlusions, and differences in image quality, the performance in controlled environments, such as in the BioID dataset (used later in this study), containing faces that are frontal, focused and well lit, with simple background, is often close to 100% hit rate with few, if any, false alarms. We made the source code for this stage available at http://kolmogorov.sourceforge.net.

Figure 6: Examples of faces and nonfaces used in training the face detector

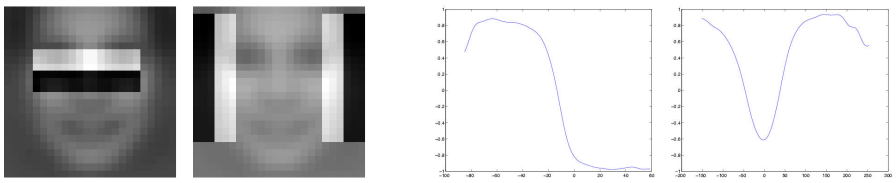


Figure 7: The first two wavelets (left) and their respective tuning curves (right) for face detection. Each wavelet is shown over the average face. The tuning curves show the evidence for face (high) vs. non-face (low), as a function of the output of the wavelet, shown increasing from left to right. The first tuning curve shows that a dark horizontal region over a bright horizontal region in the center of the window is evidence for an eye, and for non-eye otherwise. The second tuning curve is bimodal, with high contrast at the sides of the window evidence for a face, and low contrast evidence for nonface.

5.2 Stage II: Eye Detection in the Context of Faces

The first stage in the eye detection system specialized on finding general regions of the image plane that are highly likely to contain eyes. The output of the system is very resistant to false alarms but does not specify well the precise location of the eyes. The second stage specializes on achieving high accuracy provided it operates on the regions selected by the previous stage. This stage uses the same searching techniques as the previous stage: all patches within a sub-region of the face, restricted in both location and scale, are classified as eye versus not-eye.

The data used for training was from the CMU-MIT face database and the Compaq face database, this time with all positive examples containing eyes at a canonical scale and location within the 24×24 pixel patch.

We experimented with several possibilities for the choice of the location and size of the center of the eye with respect to the patch in the training set. Positive training samples were prepared by cropping example images such that the distance from the center of the eye to the left and upper edges of the cropping window were a fixed ratio r of the distance between the eyes of the source face, then scaling this sample to 24×24 pixels. If we let d be the distance between the eyes, t be an offset parameter and q be a scale parameter, then we can rewrite $r = q(d + td)$. Choosing q to be small results in a small receptive field with high resolution, while choosing q large results in a large receptive field with relatively low resolution. The choice of t shifts the location of the eye with respect to the center of the patch. From the perspective of the contextual cascade approach, it is arguable that pixels which are generated by background contain relatively little additional information once we know we are within a face, thus we should choose a t and q that maximizes the likely number of pixels in the positive example patch that are generated by the face. However, given a fixed input size of 24×24, it is possible that smaller values of q (resulting in higher resolution examples with less surrounding context) allow us to maximally benefit from the information in pixels generated by the eye only. We present results on varying these parameters experimentally to find an optimal choice of offset parameter t and scale parameter q in section 6.

The contextual cascade approach also allows us to constrain how we choose non-eye examples: We model our prior belief about the eye location π as a normal distribution, with parameters for the mean and standard deviation of the true eye position and scale with respect to the window chosen by the face detector taken

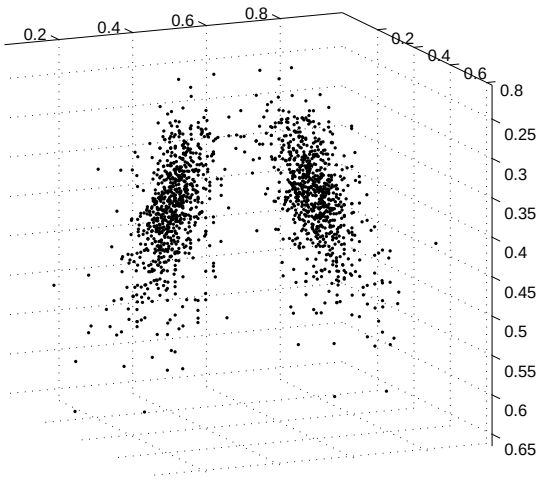


Figure 8: Location of the eyes with respect to the face detection window (x- and y-axes) as the size of the detection window with respect to the distance between the eyes varies from small (positive z-axis) to large (negative z-axis).

from the training set. In figure 8, we show the locations of eyes with respect to the size of the face detection window for some example data. Down on the vertical axis shows increasing ratio of the size of the face detection window to the distance between the eyes. When the face detector selects a small window relative to the true face size, resulting in a small detection width to eye distance ratio, the eyes tend to be far apart with respect to the detection window. When the face detector selects a large window compared to the distance between the eyes, the eyes tend to be located closer together, near the center of the detection window.

Using these statistics about the true eye positions with respect to the estimated face location, we can restrict the set of patches for searching – and thus for training also – to have a minimum Mahalanobis distance M from the mean location and scale of each eye. Choosing $M = 16.27$ gives a 99.9% confidence interval for one of the patches containing the eye (see Appendix B).

Using these criteria, for each example face, we created two positive training examples (one for each eye), and six negative training examples, where the negative examples were selected randomly from the set of patches satisfying the minimum distance from the mean eye patch size and location criterion. To make best use of our data, we flipped the positive and negative examples from the right eye about the *horizontal* axis and combined them with the left eye examples to train a single left eye detector. Then this left eye detector was flipped about the *horizontal* axis to get a right eye detector. Examples of eyes and non-eyes used in training is shown in Figure 9.

Once we have collected a set of positive and negative examples, training the eye detector uses GentleBoost as described above. We found that it is possible to achieve excellent performance with only 50-100 wavelets without over-fitting, as tested on a validation set. Thus, we chose to forgo the potential speed and accuracy benefits of the attentional-cascade and boot-strap techniques for the current experiments in favor of simpler and faster training. Figure 7 shows example wavelets and their corresponding tuning curves for one of the best eye-detectors.

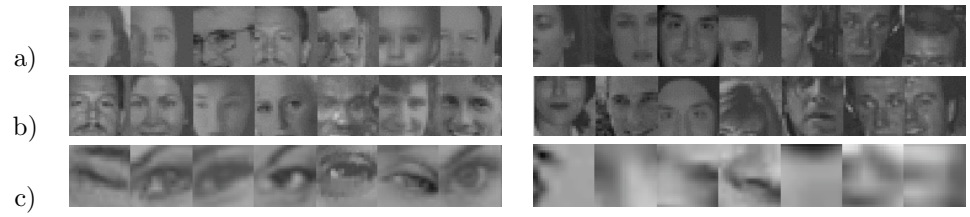


Figure 9: Examples of positive example patches (left) and negative examples patches (right) used for training three different eye detectors. Each patch is 24×24 pixels. (a) For this detector, positive examples were chosen to be centered on the eye ($t = 0$), with scaling factor $q = 1$. (b) This eye detector uses the same scaling factor in (a), but with offset parameter t chosen such that the eye is off center to maximize pixels generated by face. (c) Here, we use a smaller value of $q = .22$, so that the eye fills the window. Changing the offset parameter t would not change the number of pixels that are generated by face, so there is no corresponding off-center condition.

While the Stage I of our system (face detection) makes no assumptions about the number of faces on the image plane, the second Stage (precise location of the eyes) assumes that there is one patch rendering the left eye and one patch rendering the right eye. If the goal is to maximize the probability of choosing the correct rendering patch optimal inference requires choosing the patch that maximizes the log posterior ratio (see (26)). However if the goal is to minimize the minimize the expected squared distance from the eye optimal inference asks for computing the mean of the posterior distribution. Both approaches can be seen as examples of a more general algorithm that chooses the N patches with highest log posterior ratios and producing a weighted average of the opinions of those patches about the location of the feature of interest. In Section 6 we present accuracy results using different values of N.

5.3 Stage III: Blink Detection

Like face detection and eye detection, blink detection is done with a boosted classifier. In this case, the task is a binary classification task over a single patch per image, thus there is no need to perform a search across multiple patches. Instead, we use estimates of the eye locations to create a 44×22 pixel patch containing the eyes, doing scaling and rotation with simple linear interpolation. Training data was collected from 120 eye-open images and 120 eye-closed images collected from the Web by using the eye detector to label the eye locations, then cropping and rotating the region around the eyes to an upright frontal view. The dataset will be available at http://mplab.ucsd.edu. Figure 11 shows examples of the training data collected this way. GentleBoost is then used to select wavelets and tuning curves for this discrimination task. Figure 12 shows example wavelets and their corresponding tuning curves for the best blink detector.

6 Experimental Results

6.1 Testing Datasets

We tested the performance of the eye detector on two different types of datasets. The first dataset was the BioID dataset (Frischholz and Dieckmann, 2000; Jesorsky et al., 2001), a freely available collection of face images with eyes labeled. This

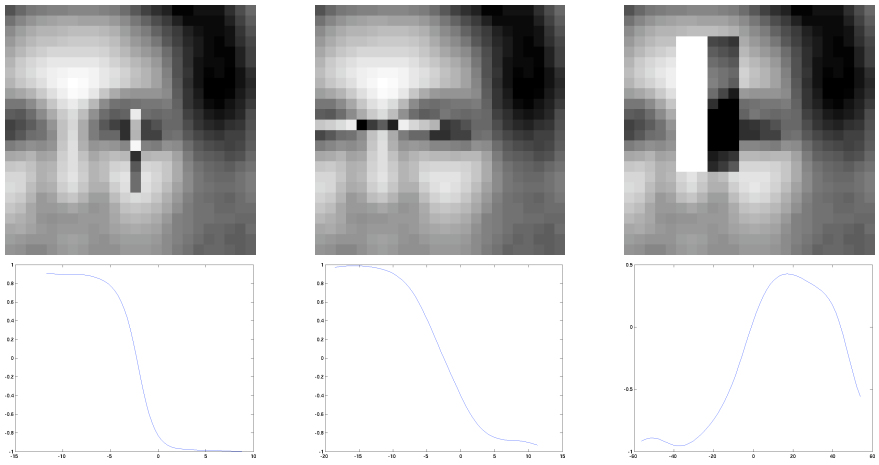


Figure 10: The first, third, and sixth wavelets (top) and their respective tuning curves (bottom) for the left eye detector centered on the eye with scale factor $q = 1$. Each wavelet is shown over the average positive (eye) example. The tuning curves show the evidence for eye (high) vs. non-eye (low) as the wavelet output increases (shown increasing from left to right). The first tuning curve shows that a dark vertical region over a bright vertical region in the center of the window is evidence for an eye, and for non-eye otherwise. The middle tuning curve looks for a horizontal band that goes dark-light-dark towards the left of the window as evidence for an eye, which appears to be testing for the bridge of the nose. The rightmost wavelet also can be interpreted as a bridge of the nose detector, however it also indicates that *toomuch* difference between the left and right parts of the wavelet are evidence *against* eye.

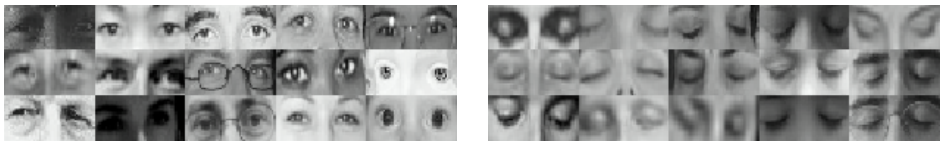


Figure 11: Example open eyes (left) and closed eyes (right) used to train the blink detector. About 120 images of each type were taken from the web to include a wide variety of lighting conditions, facial types, glasses, and cameras. The eye detection system was used to automatically crop, scale and rotate the image patches to an upright frontal view.

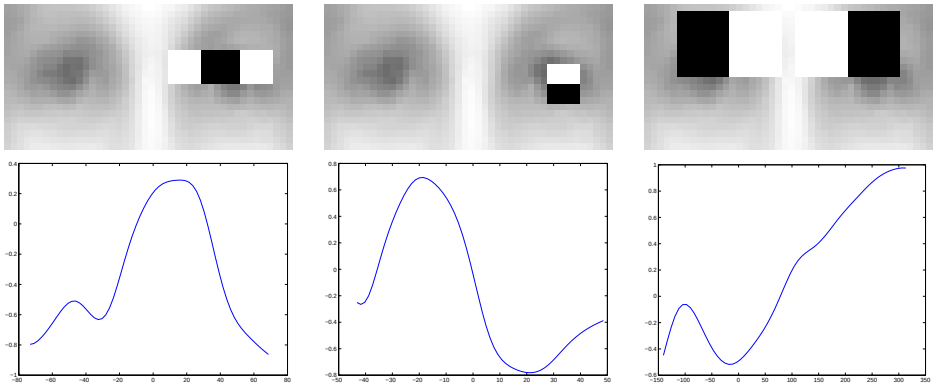


Figure 12: Features superimposed on the average open eye image (top) and their respective tuning curves (bottom) for the blink detector.

dataset contains 1521 images with good lighting conditions and frontal faces, and most subjects had their eyes open. This was to make it easier to compare our results with other eye-detection systems. The second dataset was more challenging, consisting of 400 images collected from the Web and digital cameras. We are making this dataset available at http://mplab.ucsd.edu. These images varied widely in image quality, lighting condition, background, facial expression, and head orientation, and contained 200 eyes-open and 200 eyes-closed examples. Measuring performance on this dataset allows us to compare how different parameter choices affect the quality of the system in unconstrained situations.

6.2 Eye Detection Experiments

We performed several experiments to measure the effects of different choices for the resolution and relative eye location parameters t and q for training the boosted classifier, as well as to evaluate the different techniques for estimating the actual location of the eyes (e.g., maximum posterior, vs. posterior mean).

To compare the effects of changing patch size and resolution, we tested six different choices of q, which is expressed as a ratio of the distance between the eyes, from .22 to 2.5. To see the effect of the amount of non-face image visible within the patch, the offset parameter t was chosen either (a) to keep the eye at the center of the patch, or (b) to maximize the area of the face covered by the patch while still keeping the eye as close as possible to the center. In condition (b) the choice of t and q only interact for two of the tested values for q, because for q much larger than 1.5 the entire face is always contained within the patch regardless of the offset parameter, and for values of q much smaller than 1, any offset that keeps the patch covering the entire eye contains only face pixels. Thus we have a total of eight conditions.

Varying patch size from small enough to cover just the iris ($q = .22$) to large enough to cover an area four times the size of the head ($q = 2.5$) results in a U-shaped curve, with the best performance coming from the patch with size $q = 1$, which covers about 80% of the face. The Median accuracy with this patch size is 1/5 of an iris on the BioID dataset and 1/3 of an iris on the difficult dataset from the Web. Choosing an offset parameter so that the patch is centered on the face rather than the eye did not seem to improve results.

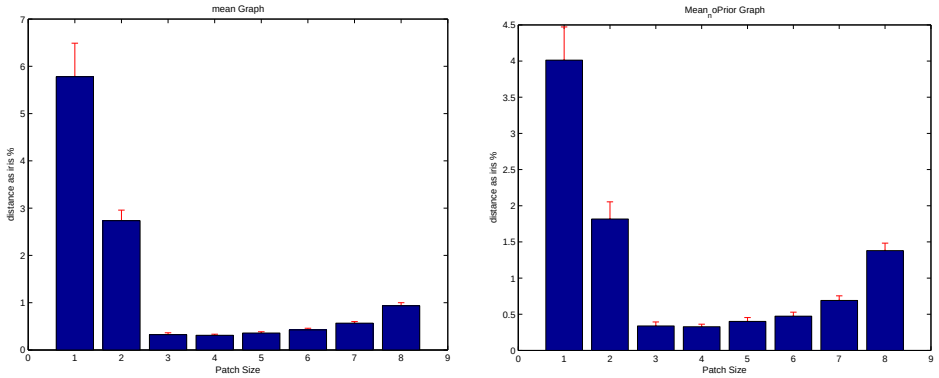


Figure 13: Median distance from center of labeled eye positions on the Web data-set as the scale parameter q and offset parameter t are varied. The graphs show the result using only the boosted classifier (left) and the full likelihood score, which combines the prior and posterior (right). In both cases, the centroid of the top 10 patches are used. From right to left, conditions are (1) $q = .22$, eye centered, (2) $q = .44$, eye-centered, (3) $q = .5$, eye-centered, (4) $q = 1$, eye-centered, (5) $q = 1$, face-centered, (6) $q = 1.5$, eye-centered, (7) $q = 1.5$, face-centered, (8) $q = 2.5$, eye-centered.

post	$q = .22$ eye-centered	$q = .44$ eye-centered	$q = .5$ eye-centered	$q = 1$ eye-centered	$q = 1$ face-centered	$q =$ eye-ce
$\text{argmax}_k(f(y_k))$	4.66 ± 0.19	2.25 ± 0.14	0.30 ± 0.03	0.27 ± 0.01	0.41 ± 0.02	$0.35 \pm$
$\text{centroid}(f(y_k))$	3.40 ± 0.23	2.07 ± 0.16	0.24 ± 0.04	0.21 ± 0.02	0.33 ± 0.02	$0.31 \pm$
$\text{argmax}_k \log p(y_k) + 2f(y_k)$	10.43 ± 0.34	2.68 ± 0.11	0.29 ± 0.02	0.26 ± 0.01	0.41 ± 0.02	$0.36 \pm$
$\text{centroid}(\log p(y_k) + 2f(y_k))$	9.47 ± 0.45	2.81 ± 0.16	0.24 ± 0.03	0.21 ± 0.01	0.31 ± 0.02	$0.28 \pm$

Table 1: Results on the BioID dataset of eye detection under different choices of patch size, offset and post-processing conditions

Tables 1 and 2 show the results for each of the patch conditions using different decision methods. These include choosing the maximum likelihood patch, taking the average of the 10 most likely patches, taking the maximum posterior patch, and taking the average of the 10 patches with the largest posterior. This fourth technique yielded the best overall results. Figure 19 shows examples of this system at work.

It is useful to visualize the log-posterior ratio maps at all search locations at several scales within an image (shown for the left eye only). In Figures 14 and 15, we show the locations of a face image that were processed by two different boosted classifiers at three of the searched scales. Each point indicates the location that would be at the center of the eye if the corresponding patch were the best eye patch. The scale of the patch increases from left to right. Thus, on the left, we see the locations where candidate patches are small, i.e., to test for the case where the distance between the eyes is small compared to the width of the face box, and on the right we see large candidate patches, which would be the scale of the true eye patch in the case where the distance between the eyes is large compared to the face detection window.

The image show the 99.9% confidence using the prior distribution for location and size of the eye with respect to the face detection window. Note the search for small

post	$q = .22$ eye-centered	$q = .44$ eye-centered	$q = .5$ eye-centered	$q = 1$ eye-centered	$q = 1$ face-centered	$q =$ eye-ce
$\text{argmax}_k(f(y_k))$	4.64 ± 0.38	2.13 ± 0.19	0.38 ± 0.04	0.37 ± 0.03	0.48 ± 0.05	$0.52 \pm$
$\text{centroid}(f(y_k))$	4.01 ± 0.46	1.82 ± 0.24	0.34 ± 0.05	0.33 ± 0.03	0.40 ± 0.05	$0.47 \pm$
$\text{argmax}_k \log p(y_k) + 2f(y_k)$	6.28 ± 0.75	2.81 ± 0.23	0.38 ± 0.05	0.36 ± 0.03	0.43 ± 0.03	$0.50 \pm$
$\text{centroid}(\log p(y_k) + 2f(y_k))$	5.78 ± 0.71	2.73 ± 0.22	0.32 ± 0.04	0.31 ± 0.02	0.36 ± 0.03	$0.42 \pm$

Table 2: Results on the Web dataset of eye detection under different choices of patch size, offset and post-processing conditions

Figure 14: Activation maps for the eye-detector with scale parameter q chosen so that the receptive field just covers the eye. Dots over the image indicate the center of checked patches. Patch size increases from left to right. The top row is the activation of the boosted classifier, the bottom row is the log posterior ratio, which combines the prior and log likelihood ratio.

eyes with respect to the face detection window is restricted to a small region near the center, and the search for larger eyes is closer to the edges, the search for eyes at the mean scale is much larger. On the top row of each figure, the color of the dot indicates the output of the boosted classifier for that patch, where brighter red indicates higher activation. On the bottom row, the color is the final likelihood score after combining the prior and the posterior information. In Figure 14, the choice of q was small enough to cover just the eye, and was trained with examples as in row (c) of Figure 9. At this scale, the detector produces many false alarms, as can be seen from the mottled activation map in the top row. While the use of the prior often helps significantly, as in this image, it often cannot overcome high activations of the boosted classifier. However, in Figure 15, trained with $q = 1$, with training samples like row (a) of Figure 9, shows that the detector is much more selective overall, and the use of the prior only serves as a small bias in the final decision.

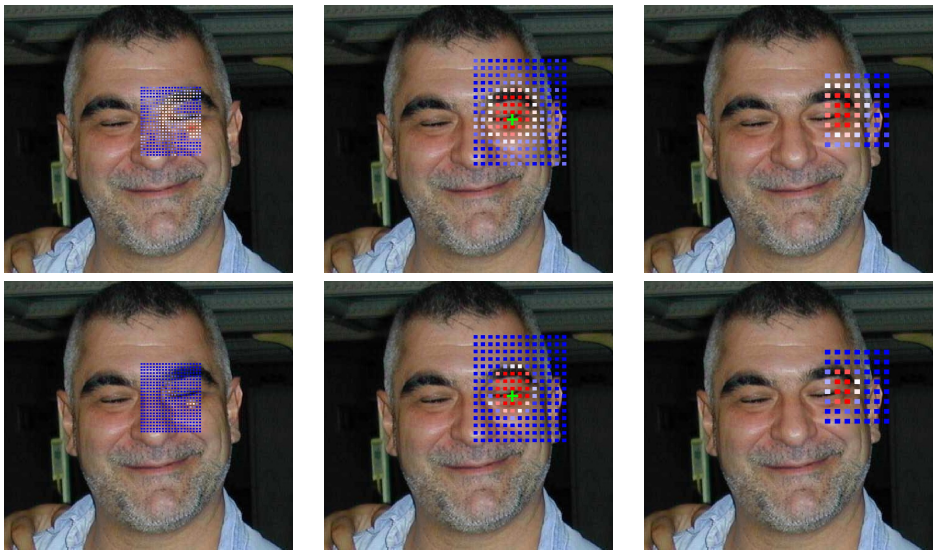


Figure 15: Activation maps for the eye-detector with $q = 1$, centered over the eye. Dots over the image indicate the center of checked patches. Patch size increases from left to right. The top row is the activation of the boosted classifier, the bottom row is the log-posterior ratio, which combines the prior and the output of the classifier. This eye detector is more accurate than the one in figure 14, thus the addition of the prior information has little effect on the final decision.

6.3 Blink detection

The best performing eye detection, with scale parameter $q = 1$ and zero offset from the center of the eye, was used to automatically crop, scale and rotate 120 examples of closed eyes and open eyes. These examples were used to train a blink detector. We stopped training after 500 wavelets and tuning curves had been chosen. The resulting classifier was then used to classify an additional 120 eyes-open and eyes-closed faces taken from the web and labeled by hand.

To assess the effects of precise localization of the eyes we compared systems that found the eyes based on the output of Stage I alone (face detection) and systems that located the eyes using Stage I and II. The effects were dramatic: adding stage II increased performance from $56.53\% \pm 8\%$ to $83.48\% \pm 6\%$.

7 Deployment: Using Eye Blink detection in an EEG fatigue study

It is our experience that realistic evaluation of machine perception systems must go beyond testing on canned datasets. We have begun evaluating the use of the system described in this document in a fatigue study in which video analysis and EEG are combined. One crucial aspect of the study is recognition of eyeblinks, as the number and duration of eyeblinks is highly correlated with the level of fatigue (Ji and Yang, 2002). The experimental setup consists of a subject fitted with a 256 electrode EEG cap, seated in front of a computer monitor, with 4 synchronized digital video cameras arranged in several view to capture face information. In Figure 16, we show a typical sequence of images in which a subject closes his eyes.

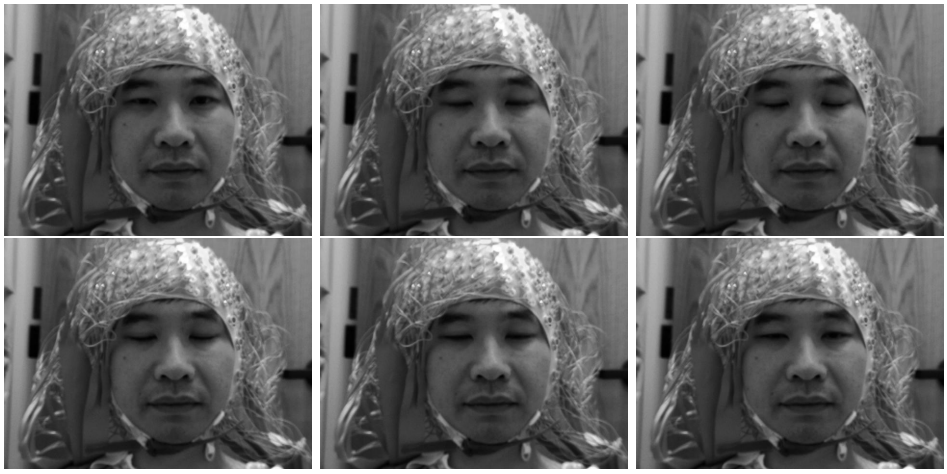


Figure 16: EEG signals are recorded with a 256 electrode cap while subjects are observed with video cameras. This shows a sequence of six consecutive frames in which a subject blinks.

In order to synchronize the cameras and the EEG recordings, the subjects were asked to blink five times at the beginning and end of the trial. In Figure 17 (top), these blinks can be seen as short spikes of increased positivity in the signal, averaged across all 256 channels. Figure 17 (bottom) shows the output of the blink detector on the corresponding video sequence, aligned with the EEG signal, for about 35 seconds. In this graph, positive indicates a high posterior likelihood of closed-eye, while negative indicates higher likelihood of open-eye. The blink detector does an extremely good job of tracking the openness of the eyes, both for voluntary and involuntary blinks. Surprisingly, although the blink detector was only trained with examples labeled as open- versus closed-eyes, the relative magnitude of the output closely tracks the dynamics of the eyeblinks as well. Figure 18 shows a closer view of the first five eyeblinks of the EEG trial in Figure 17. We can see that as the EEG signal increases and decreases in positivity during the timecourse of the eyeblink, the blink detector closely parallels the change in positivity as the eyelid covers more and less of the eye.

This level of performance has exciting implications. Current EEG research has relied entirely on the EEG and EOG signal itself for automatic eyeblink detection, in order to either reject contaminated trials, or to decompose the EEG signals into blink, muscle movement, and other artifact and brain signal components using, e.g., ICA (Jung et al., 1998). One drawback to these techniques is that contamination components could potentially also contain brain components, which is especially bad when the subject of study is directly related to eyeblink behavior itself! The blink detector, however, cannot contain any brain components, as it relies entirely on the video. Integrating a highly accurate video-based blink detection system into EEG analysis will thus allow better artifact rejection, as well as the ability to study the relationship between eyeblinks and brain signals through independent measures.

8 Conclusions

One advantage of generative models is that they force us to make explicit the conditions under which inference algorithms are optimal. Boosting approaches are

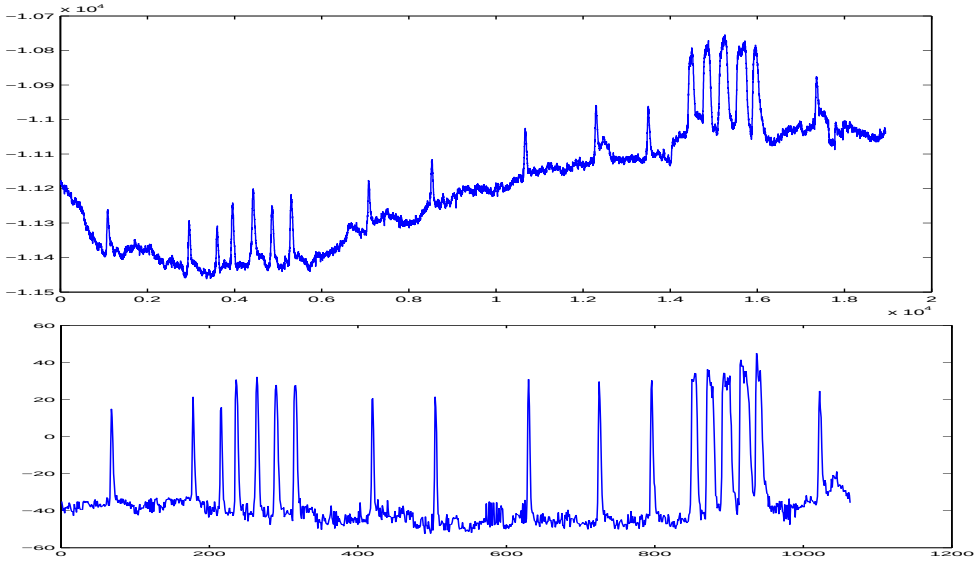


Figure 17: 35 seconds of an EEG fatigue study. The EEG signal (the average across all 256 channels is shown on top) and the blink detector output (bottom) are synchronized using the five consecutive blinks at the beginning and end of a segment.

commonly thought of as discriminative and thus discriminative approaches have been for the most part divorced from the generative approaches. In this paper we helped integrate these approaches by presenting a generative model for images under which discriminative boosting methods can be used to develop the models needed for optimal inference. The approach was applied to the problem of detecting eyes and eye-blinks in natural conditions using visible spectrum cameras. In line with the methodological stand of probabilistic functionalism (Movellan and Nelson, 2001) our emphasis was on robustness under natural conditions.

In the process of developing the eye detection system we learned several lessons that may help us understand the problems the brain needs to solve when detecting eyes:

(1) The nature of the features underlying face processing has become the subject of a heated issue in cognitive science (Cottrell et al., 2003) running the risk of becoming another undecidable debate. A popular answer is that the features used in recognizing faces are holistic in nature (Farah et al., 1988). By this is meant that during perception the face is not decomposed into specific features that are then glued together using joint feature distance statistics. As functionalists, instead of positioning ourselves on this debate we focus on understanding the nature of the problems we found when detecting faces and facial features:

First we have found that it is very difficult to analyze eye behavior (e.g., blinks) without explicitly localizing the eyes. Based on our previous work on expression recognition we think eye localization with precision in the order of 1/4 of an iris may be necessary for reliable recognition of facial expressions. Thus it seems reasonable to think that the brain may allocate resources to precisely locate facial features, including the eyes.

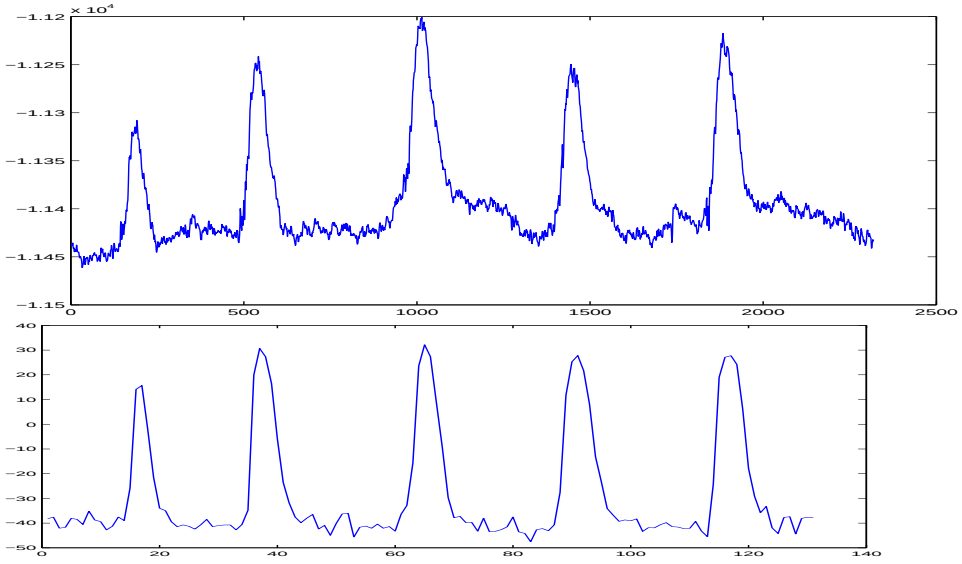


Figure 18: The EEG signal (top) and blink detector (bottom) for the first five blinks in a trial. The blink detector accurately captures the shape of the blink signal – gradually increasing and decreasing proportional to the openness of the eye.

We found that it is very difficult to develop detectors that work in very general conditions and provide high levels of accuracy about the location of the eyes. There is a trade-off between robustness and accuracy. Eye detectors that localize the eyes precisely within the face exhibit unacceptable false-alarm rates when operating outside the face. Eye detectors that avoid false-alarm rates in cluttered environments, are not sufficiently precise about the location of the eyes. We explored a solution to this tradeoff, based on a cascade of detectors that operate at different levels in the robustness/localization trade-off. Some of these detectors capture the general context in which one may find eyes. By doing so they minimize false alarms at the cost of precise position information. Precision is achieved by detectors that operate in specific contexts. If this is the strategy adopted by the brain, one would expect to find ensembles of neurons. Some of these neurons would respond to entire faces under difficult conditions. However such neurons would not be sufficient to precisely localize features. We also expect to find neurons specialized on detecting eyes in the context of faces, i.e. they should maximally excited by eyes precisely aligned and maximally inhibited by small deviations from alignment.

(2) In this paper we developed the necessary likelihood-ratio and prior models using supervised learning methods. It would be of interest to investigate whether such models can be learned using unsupervised learning methods. Another possibility is that evolution took care of developing such models. Provided a set of useful wavelets is available, our face detector requires in the order of 50 Kbytes to code the weights assigned to these wavelets. It takes an additional 2 KBytes to find eyes within faces. The complexity of the models may be greatly reduced if one could assume the conditions that occur in early mother-infant interaction. However this is an empirical question at this point that could be answer by collecting video of the world viewed from the point of view of a human infant.

(3) We focused on a system specialized on detection of eyes in a particular pose:

upright frontal. In many cases (e.g., detection of fatigue in car drivers) analysis of upright-frontal views is all is needed since frontal orientations are nominal and deviations from such orientation typically indicate fatigue or lack of attention (Ji and Yang, 2002). There are several ways one could generalize the system to work under rotations in depth. One approach we experimented with in the past fits 3D morphable models and warps them into frontal views (Bartlett et al., in press). While this method is very effective under controlled illumination conditions, it is expensive computationally and brittle when exposed to outdoor conditions. The approach we are currently exploring utilizes a collection of systems each specialized on a narrow set of face views. The approach is partially inspired by experiments showing the existence of view specific face detection neurons in infero-temporal cortex (IT) in monkeys (Logothetis and Poggio, 1994). Development of such systems is currently difficult due to the lack of labeled datasets that include sufficient number of images in multiple poses and illumination conditions. Collecting such databases is critical to accelerate progress in this field.

Appendices

A Examples

Figure 19: Examples of the eye detection system at work

B Gaussian Confidence Regions

Let Z be n-d Gaussian, zero mean with covariance I_n. Let σ a covariance matrix, with eigevectors p and eigenvalues λ, i.e. $\sigma = p\lambda p^T$. Let $\mu \in \mathbb{R}^n$. Let $Y = p(\lambda)^{1/2}Z + \mu$. Thus Y is Gaussian with covariance Σ and mean μ.

For a given $\alpha > 0$ We want the probability that $(Y-\mu)^T\Sigma^{-1}(Y-\mu)$ takes values smaller or equal to α. Now note

$$(Y-\mu)^T\Sigma^{-1}(Y-\mu) = Z^TZ = \sum_{i=1}^{n} Z_i^2 \tag{30}$$

which is a chi-square random variable with n degrees of freedom. This is the key to obtaining confidence intervals.

B.1 Example

Suppose $n = 3$, Y is gaussian with mean μ and covariance σ and we want to calculare the value α such that

$$P((Y-\mu)^T\sigma^{-1}(Y-\mu) < \alpha) = 0.001$$

, i.e., we want a volume that captures 99.9 % of the probability. First we go to the chi-square disribution with 3 degrees of freedom and find that the critical value for 1/1000 is 16.27. Thus

$$P((Y-\mu)^T\sigma^{-1}(Y-\mu) < 16.27) = P(Z^TZ < 16.27) = 1/1000 \tag{31}$$

Thus the 99.9 % confidence region for Y is given by the set of values y such that

$$(y-\mu)^T\sigma^{-1}(y-\mu) \leq 16.27 \tag{32}$$

References

Baron-Cohen, S., 1995. Mindblindness. MIT Press, Cambridge, MA.

Bartlett, M. S., Braathen, B., , Littlewort, G., Smith, E., Movellan, J. R., in press. An approach to automatic recognition of spontaneous facial actions. In: Advances in Neural Information Processing Systems. No. 15. MIT Press, Cambridge, Massachusetts.

Cohn, J. F., Xiao, J., Moriyama, T., Ambada, Z., Kanade, T., in press. Automatic recognition of eye blinking in spontaneously occurring behavior. Behavior Research Methods, Instruments, and Computers.

Cottrell, G. W. G. W., N., D. M., Padgett, C., Adolphs, R., 2003. Is all face processing holistic? the view from UCSD. In: Wenger, M., Townsend, J. (Eds.), Computational, Geometric, and Process Perspectives on Facial Cognition: Contexts and Challenges. Erlbaum.

Edleman, S., Vaina, L. M., 2001. David marr. International Encyclopedia of the Social and Behavioral Sciences.

Ekman, P., 1985. Telling Lies: Clues to Deceit in the Marketplace, Politics, and Marriage, 1st Edition. W.W. Norton, New York.

Ekman, P., Friesen, W., 1978. Facial Action Coding System: A Technique for the Measurement of Facial Movement. Consulting Psychologists Press, Palo Alto, CA.

Farah, M. J., Wilson, K. D., Drain, M., Tanaka, J. N., 1988. What is special about face perception? Psychological Review 105 (3), 482–498.

Farroni, T., Csibra, G., Simion, F., Johnson, M. H., in press. Eye contact detection in humans from birth. Proceedings of the National Academy of Sciences of the United States of America.

Fasel, I. R., Smith, E., Bartlett, M. R., Movellan, J. R., 2000. A comparison of Gabor filter methods for automatic detection of facial landmarks. In: Proceedings of the 7th Symposium on Neural Computation. California Institute of Technology.

Freund, Y., Schapire, R., 1999. A short introduction to boosting.
URL `citeseer.nj.nec.com/freund99short.html`

Freund, Y., Schapire, R. E., 1996. Experiments with a new boosting algorithm. In: International Conference on Machine Learning. pp. 148–156.
URL `citeseer.nj.nec.com/article/freund96experiments.html`

Friedman, J., Hastie, T., Tibshirani, R., 1998. Additive logistic regression: a statistical view of boosting.
URL `citeseer.nj.nec.com/friedman98additive.html`

Frischholz, R., Dieckmann, U., Feb. 2000. BioID: A multimodal biometric identification system. IEEE Computer 33 (2).

George, N., Driver, J., Dolan, R. J., 2001. Seen gaze-direction modulates fusiform activity and its coupling with other brain areas during face processing. Neuroimage 6 (13), 1102–12.

Haro, A., Flickner, M., Essa, I. A. (Eds.), 2000. Detecting and Tracking Eyes by Using Their Physiological Properties, Dynamics, and Appearance. IEEE Computer Society.

Holland, M. K., Tarlow, G., 1972. Blinking and mental load. Psychological Reports (31), 119–127.

Huang, J., Wechsler, H., 1999. Eye detection using optimal wavelet packets and radial basis functions (RBFs). International Journal of Pattern Recognition and Artificial Intelligence 7 (13).

Jesorsky, O., Kirchberg, K., Frischholz, R., 2001. Robust face detection using the hausdorff distance. In: Bigun, J., Smeraldi, F. (Eds.), Audio and Video based Person Authentication - AVBPA 2001. Springer, pp. 90–95.

Ji, Q., Yang, X., 2001. Real time visual cues extraction for monitoring driver vigilance. Second International Workshop on Computer Vision Systems (ICVS2001).

Ji, Q., Yang, X., 2002. Real-time eye, gaze, and face pose tracking for monitoring driver vigilance. Real-Time Imaging (8), 1077–2014.

Johnson, M. H., 2001. The developmental and neural basis of face recognition: Comment and speculation. Infant and Child Development 10, 31–33.

Jung, T.-P., Humphries, C., Lee, T.-W., Makeig, S., McKeown, M. J., Iragui, V., Sejnowski, T. J., 1998. Extended ICA removes artifacts from electroencephalographic recordings. In: Jordan, M., Kearns, M., Solla, S. (Eds.), Advances in Neural Information Processing Systems. Vol. 10. MIT Press, Cambridge, MA, pp. 894–900.

Karson, C. N., 1988. Physiology of normal and abnormal blinking. Advances in Neurology 25-37 (49), 119–127.

Kawashima, R., Sugiura, M., Kato, T., Nakamura, A., Hatano, K., Ito, K., Fukuda, H., Kojima, S., Nakamura, K., 1999. The human amygdala plays an important role in gaze monitoring: A PET study. Brain 122 (4), 779–783.

Kothari, R., Mitchell, J., 1996. Detection of eye locations in unconstrained visual images. ICIP96.

Leung, T. K., Burl, M. C., Perona, P., 1995. Finding faces in cluttered scenes using random labeled graph matching. Fifth Intl. Conf. on Comp. Vision.

Logothetis, N. K., Poggio, T., 1994. Viewer-centered object recognition in monkeys. Tech. Rep. A.I. Memo 1473, Artificial Intelligence Laboratory, M.I.T.

Marr, D., 1982. Vision. Freeman, New York.

Movellan, J. R., Nelson, J., 2001. Probabilistic functionalism: A unifying paradigm for the cognitive sciences. Behavioral and Brain Sciences 24 (4).

Phillips, J., 2003.

Shakhnarovich, G., Viola, P., Moghaddam, B., 2002. A unified learning framework for real-time face detection and classification. International Conference on Automatic Face and Gesture Recognition, 14–21.

Sung, K. K., Poggio, T., 1998. Example based learning for view-based human face detection. IEEE Trans. Pattern Anal. Mach. Intelligence 20, 3951.

Van-Orden, K., Jung, T. P., Makeig, S., 2000. Eye activity correlates of fatigue. Biological Psychology 3 (52), 221–40.

Viola, P., Jones, M., 2001. Robust real-time object detection. Tech. Rep. CRL 20001/01, Cambridge ResearchLaboratory.

Wiskott, L., Fellous, J. M., Krüger, N., von der Malsburg, C., 1997. Face recognition by elastic bunch graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7), 775–779.