Regression and Classification Approaches to Eye Localization in Face Images

Mark Everingham and Andrew Zisserman
Department of Engineering Science, University of Oxford
{me,az}@robots.ox.ac.uk

Abstract

We address the task of accurately localizing the eyes in face images extracted by a face detector, an important problem to be solved because of the negative effect of poor localization on face recognition accuracy. We investigate three approaches to the task: a regression approach aiming to directly minimize errors in the predicted eye positions, a simple Bayesian model of eye and non-eye appearance, and a discriminative eye detector trained using AdaBoost. By using identical training and test data for each each method we are able to perform an unbiased comparison. We show that, perhaps surprisingly, the simple Bayesian approach performs best on databases including challenging images, and performance is comparable to more complex state-of-the-art methods.

1. Introduction

We address the task of localizing the eyes in grey-level face images output by a face detector. Typically the position of the eyes would subsequently be used to warp the input image to a canonical frame for recognition: given the pair of 2-D coordinates a similarity transform is defined which transforms the eyes to fixed image positions e.g. [1]. The task of eye localization has attracted much attention in the face recognition community, not least because the accuracy of popular approaches to face recognition such as PCA or LDA has been shown to degrade with poor localization [15].

We investigate three approaches to the eye localization problem: (i) the *regression* approach directly tries to minimize the distance between the predicted and true positions by learning a functional mapping from the input image to the eye positions. Alternatively, one can aim to minimize the distance *indirectly* by classifying patches of the image correctly as eye or non-eye: (ii) the *Bayesian* approach learns models of the eye appearance and non-eye appearance and applies Bayes rule to produce a "probability of eye" output for patches around each pixel of the input image, from which a prediction can be extracted. (iii) the *discriminative* approach treats the problem as one of classification: a classifier is trained to produce positive output for patches around the eye and negative output elsewhere.

Often it may be difficult to determine from published results if a particular method is successful *per se* or because of differences in training data or test protocol. Here, we train and test each method with identical images, allowing unbiased comparison.

1.1. Previous work

In the area of 'passive' feature localization, requiring no control of illumination, a wide variety of approaches have been proposed. Methods include heuristic rules or hand-built templates [11], Gabor wavelet networks [4], and PCA-based 'Eigeneyes' [8]. Recently many authors have focused on discriminative learning of feature detectors using classifiers including support vector machines (SVM) [1]. There has been particular interest in discriminative methods based on boosting, because of their potential computational efficiency [3, 7, 16]. Methods using boosting have placed emphasis on spatial models of multiple features [3], improving precision by verification [7] or defining more discriminating weak classifiers [16].

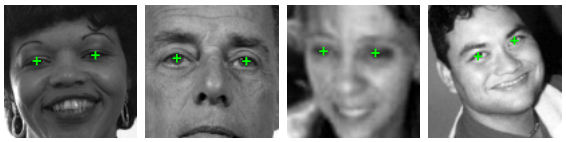
2. Approach

We begin by describing face detection and geometric normalization of the detections used as pre-processing for all methods, followed by description of the three methods investigated.

2.1. Face Detection and Normalization

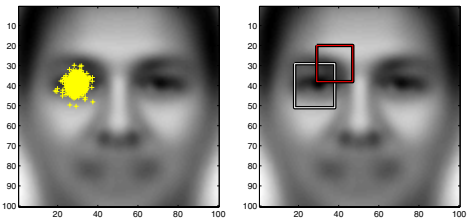
All the localization methods investigated take the region output by a face detector as input. We used the publicly available implementation of the Viola-Jones face detector [14] from the OpenCV library. The face detector outputs a bounding box $\langle x, y, s \rangle$ where $\langle x, y \rangle$ is the predicted centre of the face, and s the scale (half-width of the square). Fig. 1a shows example detections for images in the FERET database (left two) and our own WWW database (see Sec. 3). The detected face

(a) Face detections in original images



(b) Normalized images and ground truth

Figure 1. Face detection and normalization. Detected faces are scaled to size 100×100 pixels for processing. Some variation in translation and scale remains, particularly in the WWW database (right two) due to inaccuracy of the face detector localization with variation in pose.



(a) Training positions (b) Search region

Figure 2. Sliding window approach. (a) The ground truth positions of the right eye plotted on the mean training image. (b) The bounding box of the search region and an example input window.

images are normalized by scaling to a standard size of 100×100 pixels to reduce the amount of translation and scale variation in the images. Note that, particularly in the case of out-of-plane rotation of the head, considerable variation remains.

2.2. Regression Method

The first localization method considered formulates the task in a direct manner as a regression problem. We are given a set of training data $\{\mathbf{x}_i, \mathbf{y}_i\}$ where $\mathbf{x}$ is the input image and $\mathbf{y} = \langle x, y \rangle$ is a corresponding 2-D eye position. A linear regressor is defined which maps from the input image to the predicted eye position:

$$f(\mathbf{x}) = \mathtt{W}^\top \phi(\mathbf{x}) \tag{1}$$

where $\phi(\mathbf{x})$ transforms the input image to some high-dimensional space. The parameters of the regressor are learnt by minimizing the Euclidean distance between the true and predicted eye positions. To avoid over-fitting, a regularization term is added which penalizes solutions with large weights $\mathtt{W}$:

$$E(\mathtt{W}) = \frac{1}{2}\sum_i ||\mathbf{y}_i - \mathtt{W}^\top \phi(\mathbf{x}_i)||^2 + \frac{1}{2}\lambda ||\mathtt{W}||^2 \tag{2}$$

This is known as kernel ridge regression [13]. The solution can be shown to be:

$$f(\mathbf{x}) = \mathtt{Y}(\mathtt{G} + \lambda \mathtt{I})^{-1}\mathbf{z} \tag{3}$$

where columns $\mathtt{Y}_i = \mathbf{y}_i$, $\mathtt{G}_{ij} = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ and $\mathbf{z}_i = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x})$. Since only dot products $\phi(\mathbf{u})^\top \phi(\mathbf{v})$ are required, explicit definition of the mapping $\phi(\mathbf{x})$ is avoided by defining a *kernel* $K(\mathbf{u}, \mathbf{v}) = \phi(\mathbf{u})^\top \phi(\mathbf{v})$ directly. We use the radial basis function (RBF) kernel:

$$K(\mathbf{u}, \mathbf{v}) = \exp\left(-\gamma ||\mathbf{u} - \mathbf{v}||^2\right) \tag{4}$$

Input Representation. In implementing the regression method we have a choice of how to represent the input image $\mathbf{x}$. One possibility is simply to present the entire face image as a vector of pixel values. However, one might expect this to be suboptimal: much of the face image is likely to contain little information about the position of the eye, and the model must learn to 'ignore' this irrelevant variation, which is hard to achieve with limited training data. Instead we make explicit the region of the image considered relevant: the input vector $\mathbf{x}$ is extracted as the concatenation of pixels in a square region of $k \times k$ pixels; the centre of the region is fixed at the mean over the training images of the ground truth eye positions. The vector $\mathbf{x}$ for each image is normalized by subtracting the mean and dividing by the standard deviation; this gives invariance to affine transformations of intensity.

Learning. The regression method has three parameters which must be set during learning: the size of the input region k, the kernel parameter γ (4), and the regularization parameter λ (2). These parameters were set using a validation set: half of the training data was held out as validation data and a grid search over the three parameters was conducted. The parameters minimizing the mean squared error between ground truth and predicted positions on the validation data were selected, and the method re-trained on the complete training set.

2.3. Classification Methods

An alternative view of the localization problem, which has been adopted in much previous work [1, 3, 7, 16] is that of *classification*: a classifier is built which conceptually produces positive output if its input is an eye, or negative output otherwise. The classifier is applied at different locations in the input image and the

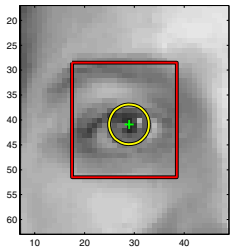


Figure 3. Selecting training examples. Informative negative examples are taken from within the rectangular search region (red) but excluding the uncertain circular region (yellow).

coordinates at which the output is positive are selected as the predicted eye position.

Sliding Windows. We investigate two classification methods which use distinctive approaches to training: Bayesian and discriminative. For both methods the inputs to the classifier are square patches of $k \times k$ pixels extracted around each pixel in a 'sliding window' fashion. Given the set of ground truth positions in the training images one can estimate the region of the face image in which the eye is likely to appear, for example the eyes are unlikely to appear in the lower half of the image. Fig. 2a shows the distribution over the position of the right eye for the training images in the FERET database. The bounding box of the points is used to define the search region (Fig. 2b). The classifier is applied to square patches around each pixel within this region.

Selecting Training Examples. For the classification methods, both positive (eye) and negative (non-eye) example patches are required for training. Fig. 3 illustrates the scheme used for selecting examples. Positive patches are extracted around the ground truth points in the training image (green cross in Fig. 3). For effective training, negative examples should match those likely to be encountered during testing; this suggests selecting negative examples within the search region (red box in Fig. 3). However, errors in the training data may mean that patches near to the ground truth points cannot reliably be considered negative examples. To cope with this a circular "uncertain" region (yellow circle in Fig. 3) is defined, and patches within this uncertain region are not used during training.

2.4. Bayesian Approach

In the Bayesian approach, probabilistic models of the appearance of a patch $\mathbf{x}$ are built independently for positive (eye) and negative (non-eye) classes. Denoting the estimated probability of a patch given the eye model $p(\mathbf{x}|e)$ and the probability given the non-eye model $p(\mathbf{x}|\bar{e})$, the log-likelihood ratio is used to assign confidence that a patch is an eye:

$$llr(\mathbf{x}) = \log p(\mathbf{x}|e) - \log p(\mathbf{x}|\bar{e}) \tag{5}$$

For each class c (eye or non-eye), the distribution over patch appearance is modelled as a Gaussian distribution:

$$p(\mathbf{x}|c) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1}(\mathbf{x}-\boldsymbol{\mu})\right\} \tag{6}$$

where $d = k \times k$ is the dimensionality of the patch.

For a given set of training data it can be shown that the maximum likelihood estimates of the mean $\boldsymbol{\mu}$ and covariance Σ are the empirical mean and covariance

$$\tilde{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{x}_i \tag{7}$$

$$\tilde{\Sigma} = \frac{1}{N-1}\sum_{i=1}^{N} (\mathbf{x}_i - \tilde{\boldsymbol{\mu}})(\mathbf{x}_i - \tilde{\boldsymbol{\mu}})^\top \tag{8}$$

While there is no reason to favour another estimate for the mean, when limited data is available, caution must be applied to the estimate of the covariance since $d(d+1)/2$ parameters must be estimated. We adopt a simple method to regularize the estimate of covariance by adding a scaled identity matrix to the empirical covariance:

$$\Sigma = \tilde{\Sigma} + \lambda \mathtt{I} \tag{9}$$

A positive value of λ has the effect of 'expanding' the distribution. The eigenvalues of the covariance matrix are increased by the constant λ, increasing the variance in each dimension; considering the size of λ relative to each eigenvalue, this also has the effect of diminishing the confidence in the directions of the trailing eigenvectors, which might well be poorly estimated. An intuitive view of the regularization is that of augmenting the training set with a set of 'virtual' examples sampled by adding Gaussian noise with variance λ to each training example. Training with noise is a well-established technique for increasing generalization in a classifier [2].

Localization. The output from the method is a log-likelihood ratio (5) at each pixel in the search region. The estimate of the eye position is taken as the position of the patch yielding the greatest log-likelihood ratio. Other possibilities such as estimating the conditional mean of the eye position are possible but not robust to outliers.

Learning. The Bayesian method has two parameters which must be set during learning[1]: the size of the input patches k, and the regularization parameter λ (9), which is set equally for both eye and non-eye classes. These parameters were set as in the regression case (Sec. 2.2) using a validation set held out from the training data.

Relation to Other Work. The Gaussian model used here has recently been applied to face detection [10]; that work applies a more complicated form of regularization to the covariance matrix. The method here is related to the 'probabilistic' PCA model [8] which models $p(\mathbf{x}|c)$ as Gaussian with a form of regularization obtained by averaging the trailing eigenvalues of the covariance matrix; a key difference is that no negative model $p(\mathbf{x}|\bar{c})$ is used in [8].

2.5 Discriminative Approach

In the discriminative approach, we directly minimize a measure of error in the assignment of patches to positive (eye) or negative (non-eye) classes. This is in contrast to the Bayesian approach which models the classes independently. The original discrete AdaBoost algorithm [5] is used to learn a 'strong' classifier $H(\mathbf{x})$ as a linear combination of 'weak' classifiers:

$$F(\mathbf{x}) = \sum_m w_m f_M(\mathbf{x}) \tag{10}$$

where the sign of $H(\mathbf{x})$ is used to assign examples to positive or negative classes. It can be shown that AdaBoost minimizes the criterion

$$E(F) = \sum_i \exp -y_i F(\mathbf{x}_i) \tag{11}$$

where $y_i \in \{-1, 1\}$ is the class assigned to each training example. This criterion can be viewed as an upper-bound on the misclassification error [12], though other interpretations are possible [6]. In terms of localization error, the 'uncertain' region used for training (Sec. 2.3) can roughly be viewed as defining a 'bounded error' model, though there is no direct connection between the classification and localization errors.

As weak classifiers we use the set of thresholded Haar-like features proposed by Viola and Jones [14]. These have the attraction of being very efficient to compute via the integral image. Full details can be found in [14]. Given the strong classifier output at each pixel, the eye position is estimated as the position with maximum output, as for the Bayesian classifier.

[1]Note in a strict Bayesian framework we should marginalize over these parameters.

Table 1. Errors in right eye localization for FERET and WWW databases. All errors are Euclidean distance in pixels measured in the normalized images. For each method the mean, median (50%) and 90th percentile (90%) are shown.

	FERET			WWW		
	mean	50%	90%	mean	50%	90%
mean	2.41	2.11	4.33	3.60	3.19	6.46
human	–	–	–	1.54	1.38	2.90
regress	1.29	1.08	2.34	1.78	1.51	3.29
generat	1.18	0.98	2.04	1.73	1.30	2.74
discrim	1.36	1.15	2.39	2.40	1.73	3.77

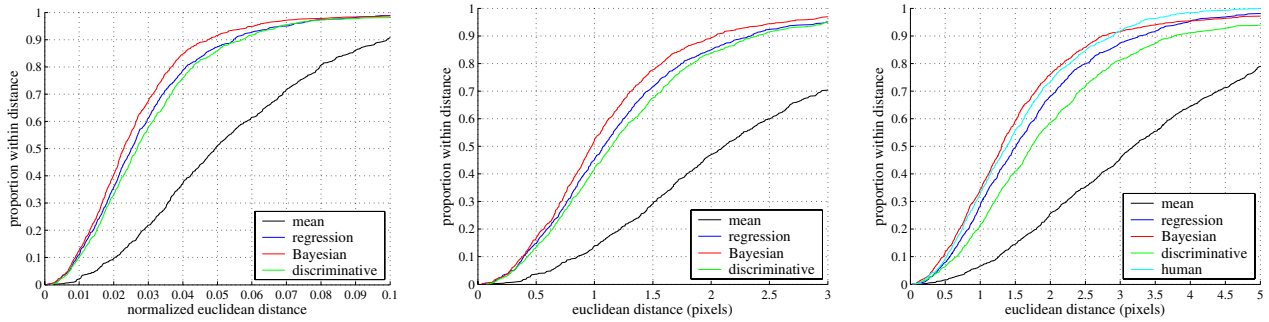
Learning. To avoid the extreme computational expense of cross-validation with boosting, the size of the input patch for the discriminative method was set equal to that selected by validation for the Bayesian method. A single strong classifier (no cascade) was learnt by bootstrapping [2]. For each round of bootstrapping, boosting was halted when the margin of $F(\mathbf{x})$ between positive and negative examples exceeded one. Training was terminated when the false positive rate (per patch) on the training data fell below 10^{-6}.

3 Experimental Results

In this section we report results of the three approaches investigated. All methods were tested on two independent image databases: gray-scale FERET [9] and WWW. The FERET images are all taken indoors, with good resolution, image quality, and limited variation in lighting. Pose of the faces in these images is typically very close to frontal. Of the 3,368 frontal images for which ground truth eye positions are provided, the 3,337 images for which the face detector detected a face correctly were retained for experiments.

The WWW database contains images obtained from the world wide web. Images are typically of much poorer quality than in face recognition databases, with highly variable image resolution, lighting, and pose of the face. The 10,118 images in the database for which the face detector operated correctly were retained for experiments. In each database, 1,000 images were selected randomly as the test set. All parameter estimation for the methods was performed on validation data taken from the training sets.

Evaluation Method. Methods were principally evaluated by graphing the trade-off in each method between localization error and proportion of successful localizations [3], see Fig. 4. On the x-axis is plotted a threshold on the Euclidean distance between the predicted eye position and ground truth position; on the



(a) FERET database (b) WWW database

Figure 4. Results of right eye localization for (a) FERET and (b) WWW databases. The left-most plot shows distances normalized by the inter-ocular distance. The right two plots show distances in pixels measured in the normalized images (note the different scales on the x-axes); we consider this measure more meaningful for the WWW database which contains large pose variation.

y-axis is plotted the proportion of images for which the distance is below this threshold.

Results. For brevity we report here the results of right eye localization; results on the left eye were very similar. Fig. 4 shows the error curves for each method on the two databases. The left plot shows 'normalized' distance reported in some previous work [3]. The other plots show distance in pixels. The curve marked 'mean' is the performance obtained by simply predicting the mean eye position in the training set for every image.

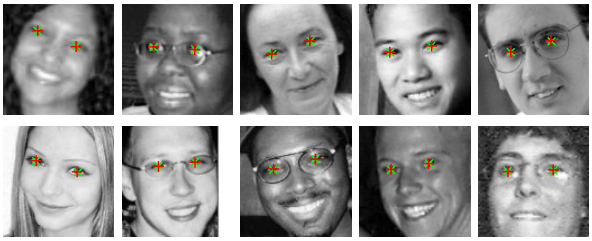
All methods perform significantly better than predicting the mean position. For both databases the Bayesian method consistently performs best, both in terms of mean localization error and bounds on the error up to 90% (Table 1). For the FERET database, in 90% of images the eye is localized to within 2.04 pixels using this method (approximately 4.7% in terms of inter-ocular distance). For the much more variable WWW database, the corresponding localization error is 2.74 pixels. Fig. 5 shows examples of good and bad localization by the Bayesian method on challenging WWW images. The regression method performs second best, and the discriminative method is consistently worst. On the FERET database the difference in mean error between the best method (Bayesian) and worst (discriminative) is only 0.07 pixels. On the more challenging WWW images, the discriminative method is markedly worse than the other methods. It is particularly interesting that the Bayesian approach performs best since this method is conceptually simple, and computationally inexpensive to train. It is also interesting to observe that this method achieves lower mean error than the regression method, which is trained explicitly to minimize this error.

Quantitative comparison with other methods is difficult due to unreported differences in the datasets used. Table 2 offers one salient comparison, with the boosting-based method of Jang & Ji [16], who report results for 400 images of the FERET database. Table 2 shows corresponding results of the Bayesian method evaluated on 1,000 FERET images. The mean displacements for the two methods are comparable, with our method giving slightly worse localization in x and better in y. It is interesting to note that the standard deviation of the displacements for our method is about half that reported in [16], suggesting more stable results.

Table 2. Comparison between Bayesian method and method of Wang & Ji [16]. The mean and standard deviation of the absolute errors in x and y-coordinates are reported, measured in the original FERET images.

	mean(x)	std(x)	mean(y)	std(y)
Wang & Ji	1.27	2.66	1.36	2.46
Bayesian	1.29	1.28	1.04	1.29

For the WWW database the test images have been labelled with eye positions by two individuals. The error curve obtained using the predictions from the second individual is shown in Fig. 4b as 'human'. One might expect this curve to have close to zero errors for a high proportion of the images. This is clearly not the case, due to errors in either individual's marking of the ground truth or disagreements on ambiguous images. The curve for the second individual is essentially indistinguishable from that obtained using the Bayesian method. It is particularly interesting to note that the Bayesian method achieves *lower* error for some parts of the curve than the second individual, suggesting that the method has learnt some specificities of the first individual's labelling.



(a) Good localization

(b) Bad localization

Figure 5. Examples of (a) good and (b) bad localization on images from the WWW database using the Bayesian method. Good localization is achieved in the presence of scale and pose variations. Persistent causes of poor localization include spectacles, hair, eyebrows and extreme lighting.

4 Discussion

In this paper we have investigated three approaches to the eye localization task, addressing the task directly by regression, or indirectly as a classification problem. A simple Bayesian approach was shown to perform best, giving comparable performance to much more complex state-of-the-art methods [16]. Informally our explanation of these results is that in the discriminative model, the small localization errors in the ground truth can affect performance greatly, whereas the simple model used in the Bayesian approach has limited capacity to fit such errors. We do not argue for the use of the Bayesian approach over discriminative approaches, but rather that the training criterion used by discriminative methods needs to be refined for this task to reflect the localization rather then classification aims. We have also investigated other discriminative approaches including SVM methods [1] which have explicit robustness to outliers, but the simple Bayesian model has thus far performed better. This reinforces our conclusion that the indirect nature of the classification approach to localization requires refinement.

Comparison with human labelling suggests that for perhaps 90% of images the limit on localization accuracy may already have been reached, and future work should concentrate on achieving comparable accuracy at even higher recall rates. Two areas seem worthy of research: (i) incorporating a reliable 'rejection' mechanism so that uncertain or multiple detections can be rejected or passed to a user for supervision, and (ii) investigating mechanisms for *automatically* correcting errors or inconsistencies in the training data to improve accuracy.

Acknowledgements. This work was supported by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778.

References

[1] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and faces in the news. In *Proc. CVPR*, pages 848–854, 2004.

[2] C. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[3] D. Cristinacce and T. Cootes. Facial feature detection using adaboost with shape constraints. In *Proc. BMVC03*, pages 231–240, 2003.

[4] R. S. Feris, J. Gemmell, K. Toyama, and V. Kruger. Hierarchical wavelet networks for facial feature localization. In *Proc. FGR2002*, pages 118–223, 2002.

[5] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings*, 1996.

[6] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[7] Y. Ma, X. Ding, Z. Wang, and N. Wang. Robust precise eye location under probabilistic framework. In *Proc. FGR2004*, pages 339–344, 2004.

[8] B. Moghaddam and A. Pentland. Probabilistic learning for object representation. In *Early Visual Learning*. Oxford University Press, 1996.

[9] P. Philips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE PAMI*, 22(10):1090–1104, 2000.

[10] J. Robinson. Covariance matrix estimation for appearance-based face image processing. In *Proc. BMVC05*, pages 389–398, 2005.

[11] J. Rurainsky and P. Eisert. Eye center localization using adaptive templates. In *Proc. CVPR Workshop on Face Processing in Video*, 2004.

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

[13] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[14] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[15] P. Wang, M. Green, Q. Ji, and J. Wayman. Automatic eye detection and its validation. In *Proc. IEEE Workshop on Face Recognition Grand Challenge Experiments*, 2005.

[16] P. Wang and Q. Ji. Learning discriminant features for multi-view face and eye detection. In *Proc. CVPR*, 2005.