In-Plane Rotational Alignment of Faces by Eye and Eye-Pair Detection

M.F. Karaaba[1], O. Surinta[1], L.R.B. Schomaker[1] and M.A. Wiering[1]

[1]*Institute of Artificial Intelligence and Cognitive Engineering (ALICE), University of Groningen*
Nijenborgh 9, Groningen 9747AG, The Netherlands
{*m.f.karaaba, o.surinta, l.r.b.schomaker, m.a.wiering*}*@rug.nl*

Keywords: Eye-pair Detection, Eye Detection, Face Alignment, Face Recognition, Support Vector Machine

Abstract: In face recognition, face rotation alignment is an important part of the recognition process. In this paper, we present a hierarchical detector system using eye and eye-pair detectors combined with a geometrical method for calculating the in-plane angle of a face image. Two feature extraction methods, the restricted Boltzmann machine and the histogram of oriented gradients, are compared to extract feature vectors from a sliding window. Then a support vector machine is used to accurately localize the eyes. After the eye coordinates are obtained through our eye detector, the in-plane angle is estimated by calculating the arc-tangent of horizontal and vertical parts of the distance between left and right eye center points. By using this calculated in-plane angle, the face is subsequently rotationally aligned. We tested our approach on three different face datasets: IMM, Labeled Faces in the Wild (LFW) and FERET. Moreover, to compare the effect of rotational aligning on face recognition performance, we performed experiments using a face recognition method using rotationally aligned and non-aligned face images from the IMM dataset. The results show that our method calculates the in-plane rotation angle with high precision and this leads to a significant gain in face recognition performance.

1 INTRODUCTION

Alignment of a face after the detection from a still image has crucial importance before the image is given to any face recognition algorithm to obtain accurate results. In particular, rotational alignment is necessary after locating the face, since in unstructured environments the face can appear in any angle rather than frontal. There are three types of rotation angle parameters which determine the pose of a face: roll (in-plane), yaw and pitch. Since the roll angle exists in 2D (hence it is also called in-plane), aligning of it is easier than the other angle parameters. Yaw and pitch angles exist in 3D, and aligning faces which are transformed by such rotations is much harder, because the aligning method has to deal with invisible or deformed parts of the face. We here propose an in-plane alignment of a face using eye coordinates that are automatically found in a face image. In this way we aim to obtain in future work high recognition results with a face recognition algorithm, without the need to use full 3D modeling techniques.

Related Work. For aligning a face image, three general methods have been used: statistical appearance modeling methods, local features methods and geometric calculation methods.

In the first approach, two related methods called Active Shape Models (ASM) (Cootes et al., 1995) and Active Appearance Models (AAM) (Cootes et al., 1998) are popular where statistical information obtained from sample training data is used. The simplest of these methods is ASM. In the ASM method, one manually labels a number of facial landmarks as salient points on example faces used for training the system. These landmark points are then used to model the facial shape. Since positions of these points are correlated, the PCA method is further applied to obtain principal components describing the variances of the point distributions and to make the further calculations computationally more efficient. Since shape information is not sufficient for modeling some complex face data, the AAM method, which is an extension of ASM, has been proposed. AAM combines the shape model with texture information for improving the face recognition system. With both approaches, especially with the latter one, promising results have been obtained. Nevertheless, an intensive labeling effort to obtain all salient points in the training images is required to train these systems.

In the second approach, one uses local features by implementing a local feature extractor without examining global information. An example method for this approach, proposed recently in (Anvar et al., 2013), utilizes the Scale Invariant Feature Transform (SIFT)

(Lowe, 2004) algorithm as a local feature detector. Here, only one face is labeled with two reference points (the mid-point between two eyes and the tip of the nose) and using the reference information, the rest of the training face images are described automatically using SIFT features. Then a Bayesian classifier is trained on the patches, which are composed of face and non-face SIFT patches, to eliminate non-face features. Since SIFT features include orientation information for each facial feature found, this information is used to estimate the rotation angle. However, high-quality face images, which are not available for every application field, are generally a prerequisite for the SIFT algorithm to perform accurately.

In the third approach, some landmark points localized by detectors are used to determine the correct alignment position of a face. The points used to align a face are usually central points of the eyes, and sometimes the mouth and tip of the nose. After locating these points by a corresponding detector, the face rotation angle can be estimated and the face can be rotated geometrically. In this approach, because the performance of the aligner will depend on the performance of the detectors, detector design becomes an important part of the method. There are two different approaches for detectors: the ones which are implemented using mathematical operators describing object specific information and the others which learn object specific information from sample images. While the methods using the former approach are also called shape-based models, the methods which are based on the latter approach are called appearance-based models. While the former one is faster, its performance strictly depends on the specification of the object to be found. The latter one is slower but more robust to illumination and other noise sources that exist in real-world data (Hansen and Ji, 2010).

To localize an object, using two or more layered systems has been shown to obtain a performance improvement. In (Li et al., 2010), such an approach has been used to align faces. In that paper, a two-layered eye localization method is adopted such that in the first layer a mathematical operator named Fast Radial Symmetry Transform is implemented to find the points with high radial symmetry in a given image. After locating eye candidate points by this operator, the eye classifier of Castrillon (Castrillón-Santana et al., 2008) is applied to eliminate false candidate points and to finally locate the eyes in a face image. After the localization, the in-plane rotation angle is estimated by using the central points of the left and right eye. In (Monzo et al., 2011), another hierarchical method is implemented. Here, in the first layer the Adaboost classifier using Haar-like features supplies many promising eye candidates to the second layer. Then the second layer implementing the histogram of oriented gradients (Dalal and Triggs, 2005) and a Support Vector Machine (SVM) is used to localize eyes.

Contributions. In this paper, we propose a simple yet robust automatic face rotational alignment method in which the in-plane rotation angle of a face is estimated using the eye locations found by eye and eye-pair detector systems. Eyes are localized by the eye detector that searches for eyes in an eye-pair patch obtained with our previously proposed eye-pair detector (Karaaba et al., 2014). The eye detector is implemented by using a feature extractor and a classifier. The method for each detector is based on a sliding window approach. We make use of the restricted Boltzmann machine (RBM) (Hinton, 2002) and the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005) to extract features from the patches belonging to the sliding window. Then the extracted features and presented to a support vector machine classifier (SVM) (Vapnik, 1998). The eye-pair detector is implemented by using an RBM and an SVM. In this paper, we compare the effects of the HOG and the RBM to study their utility for eye detection.

After locating the eyes in a face image, the in-plane angle is calculated geometrically with the arctangent formula using x and y distances between the two detected eyes. Finally, the face is rotated by using that angle. We have tested our method on (subsets of) three different face datasets, namely IMM (Nordstrøm et al., 2004), FERET (Phillips et al., 1998) and LFW (Huang et al., 2007). Our datasets contain 240, 230 and 450 face images, respectively. We have chosen to use subsets in order to save time on preparation of the datasets and on testing of the methods. We evaluate the performance of our method based on two different evaluation criteria: eye localization error and rotation error. The results show that the RBM feature extraction method performs slightly better than the HOG method on in-plane angle estimations. Moreover, we have also compared the use of rotationally aligned faces to non-aligned faces using a simple but robust face recognition system. The results of that experiment prove that rotational alignment of a face has a high impact on the recognition performance.

Paper Outline. The rest of the paper is organized as follows: In Section 2, the feature extraction techniques are described in detail. In Section 3, the eye-pair and eye detectors are described together with the method used for computing the rotation angle. In Section 4, the experimental platform, the evaluation methods, and the results of the experiments are presented. In Section 5, we conclude this paper.

2 FEATURE EXTRACTION

We will explain in this section the Restricted Boltzmann Machine (RBM) (Hinton, 2002) and the histogram of oriented gradients (HOG) (Dalal and Triggs, 2005), which are used as feature extraction methods.

2.1 Restricted Boltzmann Machines

An RBM is an energy-based neural network model used for suppression of noise and reducing the dimensionality of the input data. It is composed of two layers: an input layer and a hidden layer, which are connected to each other through (symmetric) weighted connections. There are many possible implementation methods of these layers depending on the structure of the data to be modeled. While the two layers can be implemented with the same layer type, different activation functions in different layers can also be used. The binary stochastic layer is the most prevalent implementation. We adopted in this paper, however, a linear layer for the input units and a logistic layer for the hidden units as this obtained the best performance in our experiments. The mathematical description of the RBM is briefly given below.

Let v_i be the value of input unit i and h_j be the activity value of hidden unit j that models the input data and $\hat{v}_i$, $\hat{h}_j$ are reconstructed input and hidden values. h_j is computed from the input vector by:

$$h_j = f(b_j + \sum_i v_i w_{ij}) \tag{1}$$

$\hat{v}_i$ and $\hat{h}_j$ are computed as:

$$\hat{v}_j = f(a_j + \sum_i h_i w_{ji}), \quad \hat{h}_j = f(b_j + \sum_i \hat{v}_i w_{ij}) \tag{2}$$

where $f(\cdot)$ is the activation function, a_j is the bias for input unit j, b_j is the bias value for hidden unit j and w_{ij}'s are weights connecting input and hidden units. For the linear function $f(x) = x$ and for the logistic function $f(x) = \frac{1}{1+exp(-x)}$.

To build a model using RBMs, the weight vector w is to be optimized. The most often used method to find the best weight vector, proposed by Hinton (Hinton, 2002), is the contrastive divergence algorithm. In this algorithm, the weight vector w is optimized according to the following update rule:

$$\Delta w_{ij} = \eta(\langle v_i h_j \rangle - \langle \hat{v}_i \hat{h}_j \rangle) \tag{3}$$

where η is the learning rate, $\hat{v}$ are reconstructed values of the input data and $\hat{h}$ are reconstructed values of the hidden units. The angle brackets denote the expected value of any v_i, h_j pair, which are computed using a batch of training examples. Biases are updated by:

$$\Delta a_i = \eta(\langle v_i \rangle - \langle \hat{v}_i \rangle), \quad \Delta b_j = \eta(\langle h_i \rangle - \langle \hat{h}_i \rangle) \tag{4}$$

After the optimization process, values of h_j are computed with the RBM given the input vector and then given to a classifier as a feature vector.

2.2 Histograms of Oriented Gradients

The histogram of oriented gradients, proposed first by (Dalal and Triggs, 2005) for pedestrian detection, is a feature extraction technique which computes the oriented gradients of an image using gradient detectors. It has been applied since then in many other object detection systems such as for faces (Zhu and Ramanan, 2012) and on-road vehicles (Arróspide et al., 2013), as well as for object recognition like for recognizing faces (Déniz et al., 2011), emotions (Dahmane and Meunier, 2011) and even actions (Wang et al., 2011).

The mathematical description of the HOG is briefly presented below:

$$G_x = I(x+1,y) - I(x-1,y) \tag{5}$$

$$G_y = I(x,y+1) - I(x,y-1) \tag{6}$$

where $I(x,y)$ is the intensity of the pixel at position (x,y), and G_x and G_y are the horizontal and vertical components of the gradients, respectively.

$$M(x,y) = \sqrt{G_x^2 + G_y^2} \tag{7}$$

$$\theta_{x,y} = \tan^{-1} \frac{G_y}{G_x} \tag{8}$$

While $M(x,y)$ is the magnitude of gradients, $\theta_{x,y}$ is the angle of the gradient at the given location. There are mainly two HOG descriptor calculation methods: Circular HOG (C-HOG) and Rectangular HOG (R-HOG). In this paper, we used the R-HOG method. In the R-HOG method, the image to be processed is divided into *blocks* which are composed of pixels. For each block a separate histogram is constructed after which all histograms are concatenated to form the feature vector.

As seen from the equations, angles and magnitudes are calculated from the gradients. In the HOG descriptor angles are grouped using *orientation bins*. The *orientation bins* are used to select angles for which magnitudes of gradients are collected. The appropriate bin b_θ for some angle $\theta_{x,y}$ is computed by:

$$b_\theta = \lceil \frac{\theta_{x,y} B}{2\pi} \rceil, \quad 0 \le \theta \le 2\pi, \quad 0 \le b_\theta \le B \tag{9}$$

where B is the bin size.

The calculated contributions of each pixel to the appropriate bin are weighted using the magnitudes and summed up in the final histogram.

3 EYE AND EYE-PAIR DETECTION

Here, our novel hierarchical detector system based on eye-pair and eye detectors is explained. In this system, it is assumed that a face is detected in a picture by a face detector, therefore we focus only on the eye-pair and eye detection process before the alignment. The system is comprised of two detection layers. In the first layer, the eye-pair detector searches for an eye-pair in an image containing a face. After the eye-pair is found, the eye detector, which is in the second layer, looks for the eyes in the eye-pair region. So, the eye detector assumes its input image is an eye-pair image rather than a face image. Decreasing the search space hierarchically like described above has as advantage that false positives can be greatly reduced in number. Both detectors use a sliding window method to locate the object of their interest and use a detector frame of fixed resolution. On the other hand, an input image is rescaled in a predefined range of resolutions preserving the aspect ratio of the detector frame.

3.1 Training Set Construction

To train the eye-pair and eye detector, we first created a face image dataset manually by collecting images containing human faces from the Internet. Although the faces in the images we collected are in different zoom levels, we kept the face-to-image zoom ratio always bigger than 0.5 during cropping. In addition, the collected faces are in various positions and illumination levels making them useful for eye-pair and eye detection purposes in uncontrolled environments (Karaaba et al., 2014). We will now present details about the training dataset collection for the eye detector and additional dataset collection for the eye-pair detector to make it more robust to rotated faces.

Eye Detector Dataset. To construct the eye dataset, we first cropped eye regions of the faces which are around 400 in number. We then added mirrored versions of them to the eye dataset. To obtain negatives, we have used two different methods. The first one is automatic non-eye image collection using initial eye ground truth information and the second one is obtaining the negatives by testing the system with our initially trained detector. We used approximately two times more image patches (for both the positive and negative set) than for the eye-pair dataset used in (Karaaba et al., 2014).

Further Additions. To make the system more robust to rotated faces, we have rotated the face samples in the training sets using angles of ±5°, ±10°, ±15°, ±20° using the initial in-plane angle of the faces computed from the manually selected eye coordinates. After this automatically cropped eye-pair and eye regions using the ground truth information of original cropped patches are added to the training set. After we aggregated around 1,200 new eye-pairs, we tested the systems (eye and eye-pair detector) on the training set of face images and collected more negatives. The final amount of images in the eye-pair and eye detector datasets increased to 7,000 and 13,500, respectively.

Sample eye-pair pictures used to train the eye-pair detector (in original resolution) are shown in Figure 1. Sample eye and non-eye pictures (in original resolution) are shown in Figure 2.

To locate the eyes, the SVM is invoked on all windows of the sliding window with the appropriate feature vector extracted from the window patch, and finally the highest outputs of the SVM are selected as the locations of the eyes.

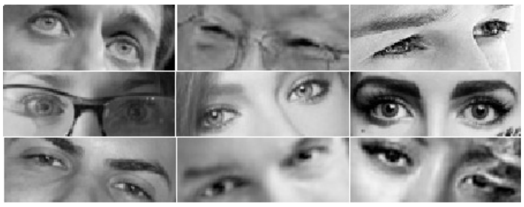
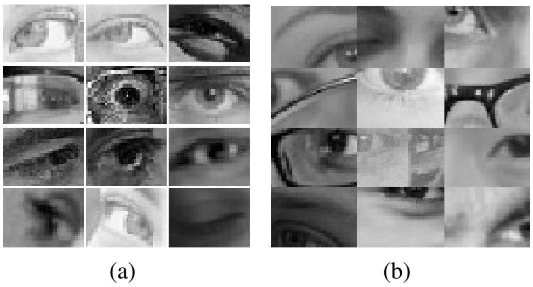


Figure 1: Sample eye-pair regions for training the eye-pair detector.



(a) (b)

Figure 2: Sample eye (a) and non-eye (b) regions cropped from eye-pair image patches. Note that the non-eye regions may still contain eyes, but they are not very precisely located in the center.

3.2 Calculating the Roll Angle

After locating the two eyes, the arctangent formula is used for roll angle calculation:

$$\text{angle} = arctan(\frac{y}{x}) \tag{10}$$

Where

$$y = eye(left)_y - eye(right)_y \tag{11}$$

$$x = eye(left)_x - eye(right)_x \tag{12}$$

Where eye(left) and eye(right) denote the central points of the two eyes. In Figure 3 a graphical representation of the roll angle estimation and the face alignment method can be seen.

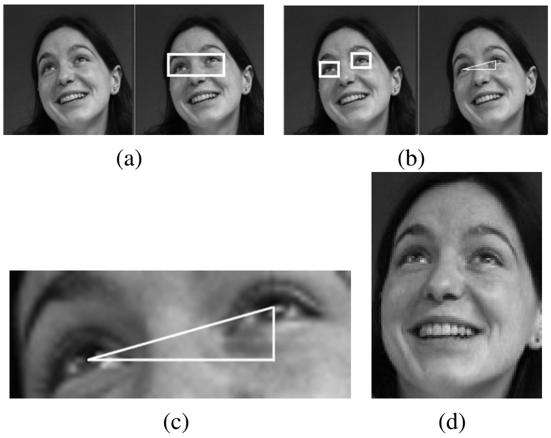


Figure 3: Rotation angle estimation stages: (a) finding eye-pair, (b) finding eyes from eye-pair, (c) computing the angle from central coordinates of eyes (17.5° in this example), (d) rotationally aligned face.

4 EXPERIMENTAL SETUP AND RESULTS

In this section general experimental parameters, the face datasets which are used in the experiments, the formulas used for evaluation, and finally the eye detection and in-plane rotation angle estimation results are given. In our experiments, an SVM classifier (Vapnik, 1998) has been employed and the RBF kernel is used as non-linear kernel due to its separability power and suitability to the datasets we used.

4.1 Experimental Parameters

For the eye-pair detector we used the same aspect ratio as in (Karaaba et al., 2014). For the eye detector the ratio of a frame is selected as 1.38. The resolution used in the eye detector which uses the RBM as the feature extractor is 18×13 and it is 36×27 for the eye detector which uses HOG. We use 50 hidden units for the RBM and around 100 epochs are employed to train the model. We use a starting learning rate as 0.03 and normalized the input data between 0 to 1 before giving them to the RBM. As for HOG, we chose 4×3×6 (4×3 as block partitioning and 6 bins). According to our observations, while higher feature dimensions for HOG gave slightly better accuracy at the expense of increased computation time, lower feature dimensions gave poorer performance in comparison to the current HOG parameters.

4.2 Datasets

For the tests, the IMM (Nordstrøm et al., 2004), the FERET (Phillips et al., 1998) and the Labeled Faces in the Wild (LFW) (Huang et al., 2007) face datasets are used. We note that the images in these datasets were only used in the testing stage. The IMM face dataset belongs to the Technical University of Denmark and is composed of 240 images with 40 individuals. The FERET dataset was created by the Defense Advanced Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST) for the purpose of testing face recognition algorithms. The full dataset is composed of 2,413 facial images with 856 individuals. We use 230 facial samples of the full dataset selected from the first 100 individual folders for our experiments. The LFW dataset is known for containing face images collected in totally unconstrained environments. It contains approximately 13,000 images of around 6,000 people. We selected alphabetically the first 450 images from this dataset. For all the selected images, we determined the rotation angles using the manually established eye coordinates. For some sample face pictures of these test datasets, see Figure 4.

Pose differences caused by yaw and roll angle changes are more prevalent in the IMM than in the FERET dataset. The LFW dataset, on the other hand, includes high variability of illumination and pose differences which makes it very challenging for computer vision algorithms.

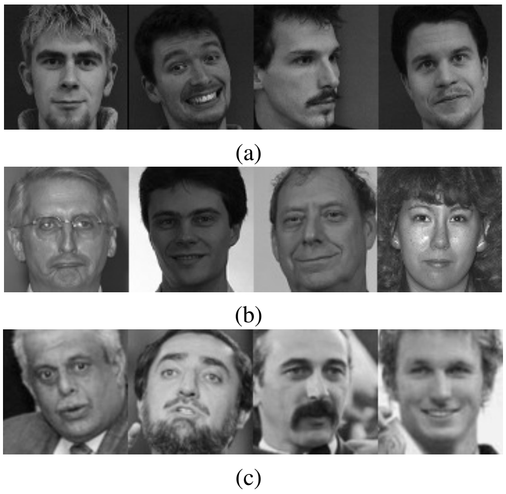


Figure 4: Sample face images of the test datasets (a) IMM, (b) FERET and (c) LFW.

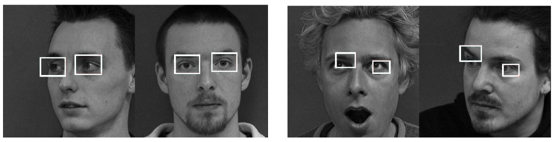
4.3 Evaluation Methods

We have used two evaluation methods for our face alignment method. The first one is the eye localization error which is calculated by dividing the pixel localization error by the eye-pair distance. The eye-pair distance is here the Euclidean distance between the central points of the two eyes. The localization error is calculated as follows:

$$e = \frac{d(d_{eye}, m_{eye})}{d(m_{eye_l}, m_{eye_r})} \tag{13}$$

where $d(\cdot,\cdot)$ in (13) denotes the Euclidean distance in 2D and in pixel units, d_{eye} denotes the (center) coordinates of the detected eye, m_{eye} are the coordinates of the manually cropped eye, m_{eye_l} represents the coordinates of the left eye, and m_{eye_r} is the same for the right eye. Some examples of face images where eyes are localized with an error lower or higher than a threshold of 0.2 are depicted as rectangles in Figure 5.

The second evaluation method is the angle estimation error which is calculated as the absolute value of the difference between manually obtained and automatically estimated angles (in degrees).



(a) (b)

Figure 5: Eyes localized with less (a) and more (b) than a localization error of 0.2.

4.4 Results

In this section we will show the results using the RBM and HOG feature extraction methods with the SVM as classifier.

We first show the eye localization errors in Table 1 and the rotation angle estimation errors in Table 2. The average localization errors and rotation estimation errors were computed on the natural data without doing any additional artificial rotation. Instead we computed the average errors from all the images we selected for the datasets.

Table 1 shows the results for localizing the eyes. The two feature extraction methods perform similarly. The average localization errors are very small (much smaller than the threshold of 0.2 shown in Figure 5). This also makes the angle estimation errors in Table 2 very small, although the rotation errors are quite sensitive to small errors in eye localization.

Table 1 also shows that, while we obtain the lowest localization errors for the IMM dataset, the performance of the method deteriorates when the method is applied to the FERET and LFW datasets. Another point is that error results on FERET are close to LFW which is known as one of the hardest datasets due to its realistic nature. The main reason for this is that although LFW possesses complex backgrounds and relatively low contrasted images, the images of FERET vary much more in illumination than the images of the other datasets (see Figure 4).

When we examine Table 2, the average rotation errors are quite small. Meanwhile, although a correlation can be seen between Table 1 and Table 2, lower position errors do not directly imply lower rotation errors. For instance, although average position error results of RBM are a bit higher than HOG results, average rotation estimation results look the opposite. This observation suggests that calculation of rotation angles are sensitive to stability of position information. In this way, we can say that the RBM feature extraction method gives more stable position information than the HOG method.

Table 1: Average Localization Error±Standard Error

Method	**Dataset**	left eye	right eye	average
RBM	**IMM**	.046±.002	.043±.002	.044±.002
	LFW	.071±.004	.069±.005	.070±.004
	FERET	.069±.009	.079±.011	.074±.01
HOG	**IMM**	.044±.006	.041±.004	.042±.005
	LFW	.066±.003	.071±.005	.069±.004
	FERET	.064±.009	.071±.01	.067±.009

Table 2: Average Rotation Error ±Standard Error

Method	**Dataset**	average error	successful rotations <2.5° (%)
RBM	**IMM**	1.35±.066	90.0±1.9
	LFW	2.30±.083	65.5±2.3
	FERET	2.38±.118	80.9±2.6
HOG	**IMM**	1.47±.082	80.0±2.6
	LFW	2.46±.096	63.4±2.3
	FERET	2.64±.12	76.5±2.8

The results on the LFW dataset are quite promising when compared to previous results. We only found one paper describing localization errors on LFW, in (Hasan and Pal, 2011) average eye localization errors on LFW are 0.081 for the left and 0.084 for the right eye. In this study, we obtained lower error rates as can be seen in Table 1.

As for a general comparison with other works, the survey paper (Song et al., 2013) presents a lot of eye detection results obtained with many other possible methods. Our methods (using HOG and RBM feature extraction methods) outperform some of these methods, although the results of the best methods presented in (Song et al., 2013) are better than the results obtained with our method. To compare to those results, we want to mention that our best method obtained 95% (96.9%) correctly detected eyes on Feret with a eye localization threshold of 0.1 (0.25), and 85.4% (99%) on LFW with a threshold of 0.1 (0.25).

We also show the plot of the average angle estimation errors in Figure 6. To construct the plot in Figure 6, we first rotated every single face image in one of the experimental datasets (IMM) to 0° degrees using the manually annotated coordinates of the eye centers. Then, we rotated every image from -30° to 30° in steps of 2° and for each angle we computed the average rotation estimation error.

The error rates of the method are lowest between -20° and 20° which corresponds to the range of angles encountered in the training set for the eye detector. Besides, a similar observation already seen in Table 1 and Table 2 about the performance of the two feature extraction methods can also be noticed here. To conclude from all of these observations, the RBM seems to better handle angle estimations than HOG.

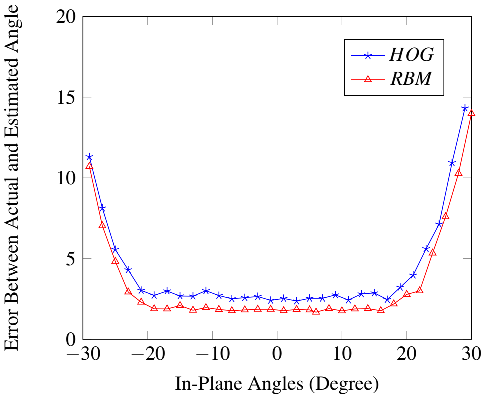


Figure 6: Angle estimation errors on the artificially rotated IMM dataset, as a function of artificial face rotation angles from -30° to 30° in steps of 2°

Face Recognition. We also show the effect of rotational alignment on the performance of a face recognition system. To make this comparison, we cropped all face images in the IMM dataset according to the eye coordinates. First, we created the *Non-Rotated* dataset, see Figure 7(a), by cropping using detected eye positions, without using angle information from eye positions to rotationally align the faces. In this way, the eye detection systems using HOG or RBM still operate in a slightly different way.

Second, we made an *Automatically Rotated* dataset, see Figure 7(b), by cropping after rotating by using the angle information using the found eye positions.

Then, we used HOG with 3×3×9 parameter settings (3×3 as block resolution and 9 bins) and 60x66 pixels resolution as the input dimension to train the face recognition system. As the IMM dataset contains 6 images per person (6×40 = 240), we selected 4 images for each class as training data and 2 for testing. Then we have in total 160 images for training and 80 images for testing. We subsequently gave the computed HOG features to an SVM *1-to-All* approach and used grid search to find the best meta-parameters to train the model. We selected HOG for this face recognition experiment particularly due to its easy training properties and its relative robustness to illumination variations. These results, however, should not be interpreted as results of an optimally working face recognition system. With this experiment, we aim to show the influence of rotational alignment. Additionally, we examine the individual effect of each feature extraction technique used in eye detection. Table 3 shows that using automatically rotated faces gives around 6 to 8 percent improvement in recognition performance. If rotated faces are compared by eye detection technique, the use of RBM in the eye detection system gives a slightly better performance than HOG and also gives the highest overall performance.

Table 3: Face Recognition Results on IMM Dataset

	detected by RBM+SVM (%)	detected by HOG+SVM (%)
Non-Rotated	74.50	75.50
Auto. Rotated	**82.75**	81.75
Improvement	8.25	6.25

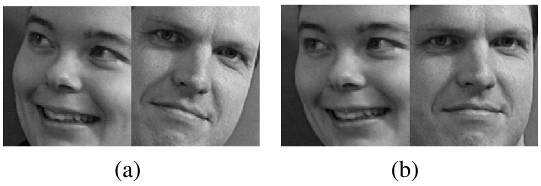


Figure 7: (a) faces in original angle and (b) faces rotated using the eye coordinates found by our best performing method.

5 CONCLUSION

Face alignment is an important step to obtain good results with a face recognition system. In this paper, we have presented a novel face alignment method based on two detectors that operate hierarchically. In this method, first the eye-pair location is found in the face image by the eye-pair detector. Then an eye detector uses the search region, which the eye-pair detector returned, to find the locations of the eyes. This location information is subsequently used to align faces by using a simple geometrical formula. For the eye detector, we also compared results of two feature extraction techniques in eye localization and rotation angle estimation. The results on three different datasets show that the RBM feature extraction technique is better at handling rotation angle estimation than HOG. This is also supported by the angle estimation error plot created by using artificially created angles. We finally examined the effect of rotational alignment in a face recognition experiment in which we compare the use of rotationally aligned and non-aligned faces in a simple face recognition system. The results show that the RBM feature extraction method gives the best angle estimation performance and this in-turn results in better performance in a face recognition system. In future work we will primarily focus on optimizing the face recognition algorithm, which will make use of the rotation alignment method presented in this paper.

REFERENCES

Anvar, S., Yau, W.-Y., Nandakumar, K., and Teoh, E. K. (2013). Estimating In-Plane Rotation Angle for Face Images from Multi-Poses. In *Computational Intelligence in Biometrics and Identity Management (CIBIM), 2013 IEEE Workshop on*, pages 52–57.

Arróspide, J., Salgado, L., and Camplani, M. (2013). Image-based on-road vehicle detection using cost-effective histograms of oriented gradients. *Journal of Visual Communication and Image Representation*, 24(7):1182–1190.

Castrillón-Santana, M., Déniz-Suárez, O., Antón-Canalís, L., and Lorenzo-Navarro, J. (2008). Face and Facial Feature Detection Evaluation. In *Third International Conference on Computer Vision Theory and Applications, VISAPP08*, pages 167–172.

Cootes, T. F., Edwards, G. J., and Taylor, C. J. (1998). Active Appearance Models. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 484–498. Springer.

Cootes, T. F., Taylor, C. J., Cooper, D. H., and Graham, J. (1995). Active Shape Models - Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59.

Dahmane, M. and Meunier, J. (2011). Emotion recognition using dynamic grid-based HoG features. In *Automatic Face Gesture Recognition and Workshops (FG 2011), 2011 IEEE International Conference on*, pages 884–888.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893.

Déniz, O., Bueno, G., Salido, J., and la Torre, F. D. (2011). Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603.

Hansen, D. W. and Ji, Q. (2010). In the Eye of the Beholder: A Survey of Models for Eyes and Gaze. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 32(3):478–500.

Hasan, M. K. and Pal, C. J. (2011). Improving Alignment of Faces for Recognition. In *Robotic and Sensors Environments*, pages 249–254. IEEE.

Hinton, G. E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800.

Huang, G. B., Ramesh, M., Berg, T., and Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst.

Karaaba, M. F., Wiering, M. A., and Schomaker, L. (2014). Machine Learning for Multi-View Eye-Pair Detection. *Engineering Applications of Artificial Intelligence*, 33(0):69 – 79.

Li, H., Wang, P., and Shen, C. (2010). Robust face recognition via accurate face alignment and sparse representation. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 262–269.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60:91–110.

Monzo, D., Albiol, A., Sastre, J., and Albiol, A. (2011). Precise eye localization using hog descriptors. *Machine Vision and Applications*, 22(3):471–480.

Nordstrøm, M. M., Larsen, M., Sierakowski, J., and Stegmann, M. B. (2004). The IMM face database - an annotated dataset of 240 face images. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, DTU.

Phillips, P. J., Wechsler, H., Huang, J., and Rauss, P. (1998). The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306.

Song, F., Tan, X., Chen, S., and Zhou, Z.-H. (2013). A literature survey on robust and efficient eye localization in real-life scenarios. *Pattern Recognition*, 46(12):3157 – 3173.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Wang, H., Klaser, A., Schmid, C., and Liu, C.-L. (2011). Action recognition by dense trajectories. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3169–3176.

Zhu, X. and Ramanan, D. (2012). Face detection, pose estimation, and landmark localization in the wild. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2879–2886.