Subject-Oriented Image Classification based on Face Detection and Recognition

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this work, a subject-oriented image classification is implemented where images are classified based on the detection of a target face within the images. This is a binary classification problem which is broken down into two phases. In the first phase, the faces within an input image are detected and segmented from the background. In the second phase, which is the focus of this work, the extracted face images are passed to a face recognition module which is implemented in Python using the AdaBoost algorithm. The performance of the face recognition module is evaluated independently on the provided test set. Finally, the face recognition module is used in conjunction with the face detection and segmentation unit to perform the image classification on the provided example set of images.

1 Introduction

Face detection and recognition systems are currently one of the most important and widely used applications of image understanding and pattern recognition. They have a wide range of application in service and rescue robots, biometrics, information security, and surveillance [1].

Face detection is considered as a binary classification problem with different approaches. One is the sampling-based approach where local visual descriptors are extracted directly from the training data and do not typically conform to any template. A renowned sampling-based learning framework was proposed by Viola and Jones [2] that uses boosting to learn discriminant features. Based on their work, many improvements or extensions have been proposed [3].

There is no single state-of-the-art face recognition system. The main reason for this is because there are many different face recognition applications that each have different requirements and constraints [4]. One widely influential face recognition algorithm is that of Moghaddam and Pentland [5]. They use a statistical approach where the intra-personal and extra-personal distributions are assumed to be Gaussian, and are approximated with principal eigenvectors which are global. One key advantage of this approach is that learning is used to focus on important differences between individuals. Another widely influential algorithm is that produced by von der Malsburg and colleagues [6]. They use a set of complex image features they call Gabor Jets. Since this approach does not use learning it is immediately applicable to new types of images. However, it cannot learn to ignore intrapersonal variations.

The method used in this work is based on Jones and Viola's work [7] for face recognition which is similar to their renowned face detection AdaBoost method [2]. Their system learns to distinguish between intrapersonal and extrapersonal variations without any distributional assumption. Unlike von der Malsburg's work, they use simple features where their scale, orientation, and form is learned by boosting.

In this work, a subject-oriented image classification problem is investigated as an application of face detection and recognition techniques. Searching for images of a particular subject among a large set of images can be an exhaustive task. The motivation behind this work is automated classification of images without human intervention. A binary classification problem is considered where the images are categorized in two different classes. The desired images are the ones that contain the target face.

1.1 Our Approach

The solution to the problem described above is broken into two steps; (1) Face detection and extraction and (2) classifying faces using a face recognition algorithm. The algorithm takes a face image of the target identity along with an input image for classification. In the first phase, the faces within the input image are detected and segmented from the background. Each cropped face is saved as a face image with a label that indicates its source image. Several face images may be extracted from an input image. The second step is classification of extracted faces. If any of the faces within the original input image is classified as the target face, the image is classified as the desired image. For classifying the faces, each face image together with the target face image is passed to a face recognition module. The face recognition algorithm decides whether the extracted face belongs to the target person.

The focus of this work is more on implementing the face recognition module. For the implementation of the face detection in the first step, the existing open source SNFaceCrop software is used. The software is developed in C++ and can detect and extract the faces in an image. It uses a Haar cascade classifier method based on the work by Viola and Jones [2].

In section 2, the face recognition method is described. The implementation results are provided in section 3. Section 4 includes conclusions and discussion on the results.

2 Face Recognition

Once the faces in the input image are detected and segmented using the SNFaceCrop software, each one of them is passed to the face recognition module as a probe face to determine if it belongs to the target person. The face recognition module comprises a binary classifier which takes a pair of a probe face and a target face as input. The image pair (I_1, I_2) is classified as a positive or negative pair based on a face similarity function that is constructed by the learning algorithm. The face similarity function is a real valued function denoted by $F(I_1, I_2)$. The pair (I_1, I_2) is classified as $sign(F(I_1, I_2))$ where $+1$ means that the face images in the pair belong to the same person. In this paper, the face similarity function is obtained as a sum of weak classifiers by using an AdaBoost learning algorithm. The method is based on Jones and Viola's method on face recognition [7]. They use an AdaBoost method based on the work of Schapire and Singer [8] that uses confidence-rated predictions. In section 2.1, the AdaBoost learning algorithm is described. The weak classifiers and selected features are introduced in section 2.2.

2.1 The Learning Algorithm

In the most popular boosting algorithm AdaBoost.M1 [9], the weak classifier has the restricted range $[-1, 1]$ whereas the AdaBoost that uses confidence-rated predictions [8] considers weak classifiers that can range over all $\mathbb{R}$. The sign of the weak classifier is the predicted label -1 or $+1$ and the magnitude is interpreted as the confidence in this prediction. Here, the confidence-rated AdaBoost method is considered in this paper.

In the face recognition module, the face similarity function $F(I_1, I_2)$ is a sum of weak classifiers $h_n(I_1, I_2)$ expressed as $F(I_1, I_2) = \sum_{n=1}^{N} h_n(I_1, I_2)$. Each weak classifier $h_n(I_1, I_2)$ consists of a filter ϕ_n that acts on both face images. Let

$$f_n(I_1, I_2) = |\phi_n(I_1) - \phi_n(I_2)|\,, \tag{1}$$

Then the weak classifier $h_n(I_1, I_2)$ is defined as

$$h_n(I_1, I_2) = \begin{cases} \alpha_n, & f_n(I_1, I_2) \le t_n \\ \beta_n, & \text{otherwise} \end{cases}, \tag{2}$$

where $t_n \in \mathbb{R}$ is a feature threshold, ϕ_n is a scalar filter and α_n and β_n are constants. The predicted label by the weak classifier is $\mathrm{sign}(h_n(I_1, I_2))$. When the output of the weak classifier h_n is α_n, the predicted label is $+1$ and the absolute value of α_n measures the confidence in this prediction. The absolute value of β_n measures the confidence in predicting the label -1. Consider S training sample pairs (x_i, y_i) where $x_i(I_1^{(i)}, I_2^{(i)})$ and $y_i \in \{-1, +1\}$ for $i \in \{1, \ldots, S\}$. The sample weights in the round n of the boosting algorithm are denoted by $\{w_i^{(n)}\}_{i=1}^{S}$. The best weak classifier h_n which is characterized by $(\phi_n, t_n, \alpha_n, \beta_n)$ is selected such that the exponential loss function is minimized:

$$h_n = \arg\min_{h} \sum_{i=1}^{S} w_i^{(n)} \exp(-y_i h(x_i)) \tag{3}$$

Similar to the AdaBoost.M1 method, the above optimization is broken into two major parts [10]. Initially, the filter ϕ_n and the feature threshold t_n is selected by minimizing the weighted error. To this end, for each filter ϕ, the threshold t is optimized by minimizing the weighted error ϵ obtained as

$$\epsilon = \sum_{i: y_i = -1,\, f(x_i) \le t} w_i^{(n)} + \sum_{i: y_i = +1,\, f(x_i) > t} w_i^{(n)}, \tag{4}$$

where $f(\cdot)$ is defined in (1) and $x_i = (I_1^{(i)}, I_2^{(i)})$. Then, the best filter ϕ_n is selected as the filter with minimum error among all the filters. Once the best filter ϕ_n and its threshold t_n are obtained, the optimal values for α_n and β_n are obtained by minimizing the exponential loss function defined in (3). The optimal values of α and β for the optimal filter ϕ_n and threshold t_n are obtained as [7]

$$\alpha_n = \frac{1}{2} log \left(\frac{W_n^{++}}{W_n^{-+}} \right), \quad \beta_n = \frac{1}{2} log \left(\frac{W_n^{+-}}{W_n^{--}} \right), \tag{5}$$

where W_n^{++}, W_n^{-+}, W_n^{+-}, and W_n^{--} are defined as

$$W_n^{++} = \sum_{\substack{i \,:\, y_i \,=\, +1, \\ f_n(x_i) \,\le\, t_n}} w_i^{(n)}, \quad W_n^{-+} = \sum_{\substack{i \,:\, y_i \,=\, -1, \\ f_n(x_i) \,\le\, t_n}} w_i^{(n)},$$

$$W_n^{+-} = \sum_{\substack{i \,:\, y_i \,=\, +1, \\ f_n(x_i) \,>\, t_n}} w_i^{(n)}, \quad W_n^{--} = \sum_{\substack{i \,:\, y_i \,=\, -1, \\ f_n(x_i) \,>\, t_n}} w_i^{(n)}$$

and $f_n(x_i) = \left| \phi_n(I_1^{(i)}) - \phi_n(I_2^{(i)}) \right|$. The steps of the AdaBoost algorithm are described as follows:

AdaBoost algorithm:

- Given S samples $(x_1, y_1), \ldots, (x_S, y_S)$ where $y_i \in \{-1, +1\}$ and $x_i = (I_1^{(i)}, I_2^{(i)})$ is a pair of face images,
- Initialize weights $w_i^1 = \frac{1}{S}$.
- Let R be the number of rounds to boost before resampling.
- For $n = 1, \ldots, N$:

1. For each filter ϕ_j in the filter library, compute the optimal threshold t_j that minimizes the error ϵ in (4) for each potential filter.
2. Choose ϕ_n with the lowest error and save the corresponding threshold t_n.
3. Choose α_n and β_n according to (5). The weak classifier h_n in (2) is now defined.
4. If n is a multiple of R, then resample to generate a new training set with new weights. Otherwise update the weights and normalize them: $w_i^{(n+1)} = w_i^{(n)} \exp(-h_n(x_i) y_i)$ and $w_i^{(n+1)} \longleftarrow \frac{w_i^{(n+1)}}{\sum_{i=1}^{S} w_i^{(n+1)}}$.

- The final strong classifier is $F(x) = \mathrm{sign}\left(\sum_{n=1}^{N} h_n(x) \right)$

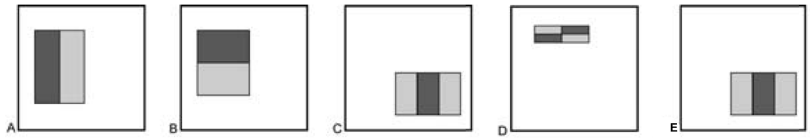


Figure 1: The types of rectangle filters [7] used in this work . The complete set of filters ranges over all scales, aspect ratios and locations in the analysis window.

2.2 Feature Selection

The potential filters ϕ, defined in (1), in the construction of face similarity function are a set of rectangle features similar to those described by Viola and Jones [2]. A rectangle filter is a type of Haar filter which quantizes an image region to a scalar [11]. The set of rectangle filters used in this work are shown in figure 1. A rectangle filter is computed by summing the intensities of all pixels in the dark regions and subtracting the sum of the intensities of all pixels in the light regions. A multiplier may be factored in to make the total number of pixels in the dark rectangles equal to the total number of pixels in the light rectangles. The computation of rectangle features are sped up by using the integral image representation of the input image [2]. The integral image of all the images in the database can be computed and stored once. Using the stored integral values, the computation of rectangle filters can be performed in constant time.

The goal in face recognition is to distinguish between the intrapersonal and extrapersonal variations in the images. The form, scale, orientation, and location of the rectangle features is learned using the training images so that the final constructed filter can distinguish between the intrapersonal and extrapersonal variations. In each round of the boosting algorithm, the rectangle filter that best classifies the intrapersonal and extrapersonal variations is selected among the library of rectangle filters. Each filter measures a particular property, at a given location, scale, and aspect ratio, and is assigned a weight. The feature thresholds determine which variations are acceptable [7]. If a region of the face, such as the hair, is not a good indicator of face similarity in the training images, then it is likely that no filter will be chosen in this region by the learning algorithm.

3 Results

The face recognition algorithm is trained and tested using the face database of AT& T Laboratories Cambridge available at [12]. The images are taken at different times, illumination, and facial expressions. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position.

After training the face recognition algorithm, we used it in conjunction with the SNFaceCrop software for face detection to classify 210 pictures from a personal database.

3.1 Face Recognition Results

The AT& T face database includes 40 distinct subjects each with 10 different images. The images are taken at different times with different illumination. Also the facial expression changes from open to closed eyes, smiling to not smiling. Some facial details are also varying. For example, there are glasses in some of the images of some subjects. The subjects are in an upright, almost frontal position. The pose angle varies in the range $[-23, 23]$ degrees where the zero reference point is the frontal position. In this work, the images are cropped to 102 pixels high and 80 pixels wide to focus on more informative regions.

Half of the database is used for training. There are 20 subjects each one with 10 images in the training set. Therefore, the training set yields only $20 \times \binom{10}{2} = 900$ same pairs out of $\binom{200}{2} = 19900$ total pairs. In order to keep the balance in the training process, at any round only 900 different pairs and all 900 same pairs are used for training. At each resampling round in the algorithm, 900

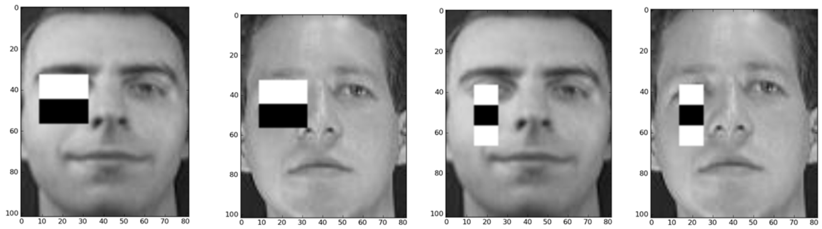
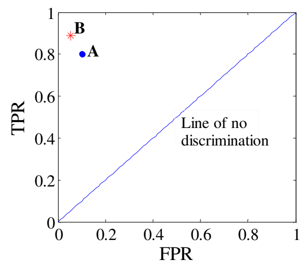


Figure 2: The first and second rectangle filters learned in the AdaBoost algorithm: The scale and location of these filters are demonstrated on two images of the training set. The filters examine the eye and the areas around it. The first filter is a two-rectangle filter and the second one has three rectangular regions.



	A	B
FPR	0.10	0.05
TPR	0.80	0.89
T	200	400
R	50	100

Figure 3: The false positive rate (FPR) and true positive rate (TPR) of the implemented face recognition algorithm is evaluated for two cases. The number of total rounds or total weak classifiers in the AdaBoost algorithm is denoted by T. The number of boosting before resampling is R. In the algorithm, when the round number is a multiple of R, 900 different pairs are resampled from 19000 different pairs in the training set. The 900 same pairs remain unchanged in the rounds.

different-pair samples are taken from the total 19000 different pairs in the training set. After training, the entire database of 40 subjects which includes a total of 400 images is used as the test set.

There are five types of rectangle filters shown in figure 1. The first two filters learned by boosting are shown in figure 2. These filters examine the eye and parts of cheek below the eye. The first filter is a two-rectangle filter and the second one has three rectangular regions.

For evaluating the classification performance, the false positive rate (FPR) and the true positive rate (TPR) should be evaluated. TPR determines a classifier performance on classifying positive instances correctly among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. Different values of TPR and FPR can be obtained by changing the boosting rounds R and T in the algorithm. These values are illustrated in the ROC space [13] in figure 3.

In a binary classification problem, the best theoretical prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100 % sensitivity and 100 % specificity. A completely random guess would give a point along a diagonal line from the left bottom to the top right corners. The distance from the random guess line is the best indicator of how much predictive power a method has.

The entire AT & T database is used as the test set in evaluating the TPR and FPR values. Two cases are considered in evaluating the error rates. In case A, the number of total boosting is equal to $N = 200$ and the sum of 200 filters are used for classification. The number of boosting R before resampling is set to $R = 50$ in this case. Case B corresponds to $N = 400$ and $R = 100$. The obtained TPR and FPR values are shown in figure 3.

3.2 Subject-Oriented Image Classification Results

The test images for the original image classification problem are selected from a personal database where there are more than 20 subjects in a total of 210 photos. Initially, all the faces in all images are detected, segmented and saved using the SNFaceCrop software in batch mode. The file name of each saved face has a label that indicates the source image. Then each face image and the target face image are provided to the face recognition algorithm obtained in case B in section 3.1. The program categorizes the source images in two different directories by saving each image into the directory of the predicted class.

There are 32 desired images that contains the target face. Out of this number, 21 images are classified properly which infers that the true detection rate is almost 0.7 in this case. Also, 18 undesired images are miss-detected.

4 Conclusions and Discussion

In this work, a subject-oriented image classification problem is solved by implementing an AdaBoost algorithm for face recognition. Rectangle filters are used as the features to obtain the best weak classifier in each round of the AdaBoost algorithm. To provide the face recognition module with face images, an existing face detection software is used to extract face images from background. The results in section 3 show that the implemented face recognition algorithm has acceptable error rates on the provided test set. The performance improves by increasing the number of boosting rounds in the algorithm from 200 to 400.

Regarding the classification results in section 3.2 for the original problem, the face recognition performance is not as strong as it was evaluated in section 3.1. This is due to the fact that the face detection error is added to the classification. Besides, the face recognition algorithm is introduced to a new set of images which is different from the AT & T face database used for training.

References

[1] V. BELLE. Detection and recognition of human faces using random forests for a mobile robot. Master's thesis, RWTH Aachen University, Aachen, 2008.

[2] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pages I–511–I–518, Kauai, HI, December 2001.

[3] B. Wu, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Proc. of IEEE Conf. on Automatic Face and Gesture Recognition*, pages 79–84, Seoul, Korea, 2004.

[4] M.J. Jones. Face recognition: Where we are and where to go from here. *IEEJ Trans. on Electronic, Information and Systems*, 129(5):770–777, January 2009.

[5] Baback Moghaddam and Alex Pentland. Beyond euclidean eigenspaces: Bayesian matching for visual recognition, 1998.

[6] K. Okada, J. Steffens, T. Maurer, Hai Hong, E. Elagin, H. Neven, and C. von der Malsburg. The bochum/usc face recognition system and how it fared in the feret phase iii test, face recognition: From theory to applications. *Springer-Verlag*, pages 186–205, 1998.

[7] M. J. Jones and P. Viola. Face recognition using boosted local features. Technical report, TR 2003-25, MERL, April 2003.

[8] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Journal of Machine Learning*, 37:297–336, December 1999.

[9] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, August 1997.

[10] T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer series in statistics. Springer, 2008.

[11] C.P. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conf. on Computer Vision*, pages 555–562, Bombay, India, January 1998.

[12] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.

[13] http://gim.unmc.edu/dxtests/Seriesof.htm.