LEARNING WEIGHTED SIMILARITY MEASUREMENTS FOR UNCONSTRAINED FACE RECOGNITION

Andre Störmer and Gerhard Rigoll

Institute of Human Machine Communication
Technische Universität München
Arcisstr. 21, 80333 München

ABSTRACT

Unconstrained face recognition is the problem of deciding if an image pair is showing the same individual or not, without having class specific training material or knowing anything about the image conditions. In this paper, an approach of learning suited similarity measurements is introduced. For this the image is partitioned into several parts, to extract image region based histograms of gradients, local binary patterns and three patch local binary patterns. The similarities of respective patches are computed and it is learnt how to weight the different image regions. Finally, a fusion is applied using a Multi Layer Perceptron. Evaluations are done on the "Labeled Faces in the Wild" dataset.

Index Terms— face recognition, pair matching, image descriptors

1. INTRODUCTION

Face Recognition has been an emphasized topic in the past years. Many successfull approaches have been reported that are able to identify persons, a survey on the most prominent approaches can be found in [1].
However all of these methods are working in more or less controlled environments and assume that a gallery image of each person is available, that can be used as training material. In difference to this classical face recognition, unconstrained face recognition aims at the recognition task of pair matching. It should be decided whether two given images contain the same individual or not, without knowing anything else than that both images contain faces. In the "Faces of the Wild" dataset [2], a dataset obtained by scanning all kind of images available at the internet with a Viola and Jones [3] based face detector, a large variation in pose, lighting, expression, background, gender, clothing, hairstyle and camera is given (see Fig. 1). A strict benchmarking scheme enforces an evaluation on 10 sets with 10 separate experiments using a leave-one-out crossvalidation scheme. All of these sets are disjunct on the person level, that means, that not a single individual of the test set is known during training.
This dataset, with the given task of pair matching, leads to the development of similarity learning techniques. Recently some approaches have been proposed to measure the visual similarity of two unseen images [4][5]. Descriptor based methods have been suggested for face recognition, these could also be used in this scenario [6]. Results of different approaches on the "Labeled faces in the Wild" dataset have been published in [7][8][9].

Fig. 1. Samples of the dataset "Labeled Faces in the Wild". The upper pair shows the same individual, the lower pair shows different individuals.

2. LOCAL IMAGE DESCRIPTORS

In this approach, local image descriptors are first computed for each image. This is done, by partitioning the image into several equal sized patches. Then histograms of different features are computed and normalized that for each histogram the sum of all bins is one (see Fig. 2).

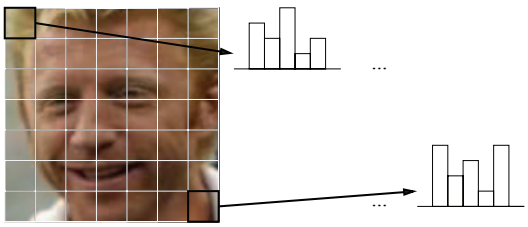


Fig. 2. The images are divided into subregions, in each of these subregions histograms of the different features are computed and stored.

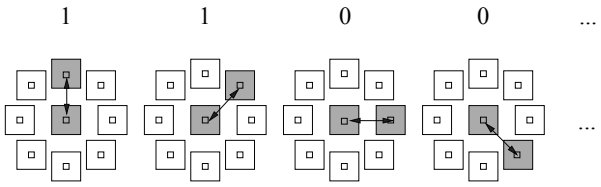


Fig. 3. Local binary pattern: clockwise comparison if the mean value of the central patch is smaller than the mean values of neighboring patches. Results are stored in a bitsequence.

2.1. Histogram of Gradients

The Histogram of Gradients(HoG) is a well known descriptor proved to be successful for tasks like human or object detection [10]. Because of the unconstrained characteristic of the data, it is also useful for this task. Now, it is described, how this feature is extracted in this approach. The partial derivatives in x and y - direction are computed by a Sobel-Operator.

$$\hat{\nabla}(I(x,y)) = \begin{pmatrix} \mathrm{S}_x(I(x,y)) \\ \mathrm{S}_y(I(x,y)) \end{pmatrix} \quad (1)$$

the gradient magnitude is then computed as:

$$|\hat{\nabla}(I(x,y)_t)| = \sqrt{(\mathrm{S}_x(I(x,y)))^2 + (\mathrm{S}_y(I(x,y)_t))^2} \quad (2)$$

and the gradient orientation can be estimated by

$$\hat{\Theta} = \arctan\left(\mathrm{S}_x(I(x,y)), \mathrm{S}_y(I(x,y))\right) \quad (3)$$

The gradient orientation is discretized to the eight possible orientations in a one pixel neighborhood. After that, different methods of sparsening the gradient field can be applied. Useful approaches are thresholding of the magnitude and a non-maximum-suppression (NMS), by checking for each pixel, if the gradient has a higher magnitude than the gradients of the two neighboring pixels in the gradient direction. To obtain the HoG, the occurrence of each of the eight orientations is counted on the gradients that remain after NMS and stored in a histogram. This is done for each image patch separately. In this paper, the magnitude is only used for the non-maximum-suppression, resulting in an eight bin histogram per image patch, each bin representing one gradient orientation.

2.2. Histogram of local binary patterns

A texture descriptor called local binary pattern (LBP) [6] has performed well for face recognition. LBPs are created by comparing the average value of a the center patch to the average values of surrounding patches. If the value of the center patch is smaller than that of the surrounding, the resulting value is set to "1" else it is set to "0" (see Fig. 3). The subsequent pattern of binary values created by clockwise comparison of all neighboring blocks is then the LBP describing local texture information. Here, eight neighboring patches are used. The resulting eight-bit-sequences are checked, if they are uniform patterns. If there are more than two "01" or "10' subpatterns in the sequence they are not uniform. Example: 00111110 is a uniform pattern, 01100101 is not. Thus, each of the 58 possible uniform patterns describes a circular feature where the average of one segment of the surrounding patches is larger then the center, the rest is not. The LBP is computed for all pixels inside a given image region. Then the histogram of LBPs is computed for this region. All non-uniform patterns are stored into one bin, each uniform pattern get a bin of its own, resulting in 59 bins in total. In this paper, two different configurations of LBP are used: A LBP with a patch size of 3×3 pixels and a radius of 3 pixels between the center pixel of the center patch and the center pixel of the surrounding patches called LBP3 and a LBP with a patch size of 5×5 pixels and a radius of 5 pixel called LBP5.

2.3. Histogram of three patch local binary pattern

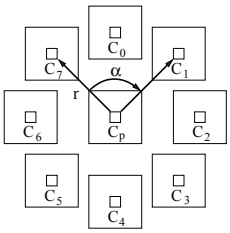
The three patch local binary pattern (TPLBP) has been introduced in [7] and is very related to the LBP. The main difference is, that three patches are used to compute a single bit value. As a first step, a distance d between two patches has to be defined. In this paper the Euclidean distance is used for simplicity. Then for each pixel the TPLBP using eight neighbors is computed as follows:

$$\mathrm{TPLBP}_{r,\alpha} = \sum_{i=0}^{7} f(d(C_i, C_p) - d(C_{(i+\alpha) \bmod 8}, C_p))2^i \quad (4)$$

with

$$f(x) = \begin{cases} 1 & \text{if} \quad x > 0 \\ 0 & \text{else} \end{cases} \quad (5)$$

Here C_p is the central patch, C_i are the patches along the circle in a clockwise numeration, r is the radius or the distance of the central pixel of the central patch to the center pixel of a neighboring patch, α is the distance between the two patches along the circle, which are compared to the central patch (see Fig. 4). For each pixel inside an image region the TPLBG is computed, then a histogram is built putting all non-uniform pattern together into one bin, each uniform patterns gets again a bin of its own.



$$
\begin{aligned}
&TPLBP_{r,2}(p) = \\
&f(d(C_0,C_p) - d(C_2,C_p))2^0 \\
&f(d(C_1,C_p) - d(C_3,C_p))2^1 \\
&f(d(C_2,C_p) - d(C_4,C_p))2^2 \\
&f(d(C_3,C_p) - d(C_5,C_p))2^3 \\
&f(d(C_4,C_p) - d(C_6,C_p))2^4 \\
&f(d(C_5,C_p) - d(C_7,C_p))2^5 \\
&f(d(C_6,C_p) - d(C_0,C_p))2^6 \\
&f(d(C_7,C_p) - d(C_1,C_p))2^7
\end{aligned}
$$

Fig. 4. Computation of the TPLBP with $\alpha = 2$

3. LEARNING SIMILARITY CLASSIFIERS

The similarity function that will be learnt is a weighted sum of similarity measures between the histograms of the respective patches of two images. The histogram of each image region in image A will be compared to the histogram at the same position in image B. This is done by computing the probabilistic symmetric χ^2-measure $d_{p\chi^2}$ or the Bhattacharyya distance d_B between the two histograms h_A and h_B for each patch.

$$d_{p\chi^2}(h_A, h_B) = 2\sum_{i=1}^{N} \frac{(h_B[i] - h_A[i])^2}{h_A[i] + h_B[i]} \tag{6}$$

$$d_B(h_A, h_B) = -\ln(\sum_{i=1}^{N} \sqrt{h_B[i]h_A[i]}) \tag{7}$$

Here N is the number of bins in the histogram, all histgrams are normed to a sum of one. For every image region both measurements are determined. To derive a final similarity measurement $D(A, B)$ for the comparison of the whole images A and B, a weighted sum of all local distances d is computed. Here d can be either $d_{p\chi^2}$-measure or Bhattacharyya distance d_B. If each image is divided into $K \times L$ subregions, $K \times L$ distances d will be weighted with $K \times L$ different weights $w_{k,l}$ and summed up:

$$D(A, B) = \sum_{k=1}^{K}\sum_{l=1}^{L} w_{k,l} d(h_{A,k,l}, h_{B,k,l}) \tag{8}$$

The learning involves now the finding of the weights $w_{k,l}$ to maximize the descriptive power of this measurement. This weighted sum can easily be modeled by a Neural Network with $K \times L$ input neurons and one output neuron, which is set to "1" if the given image pair shows the same person, and set to "-1" if not. Training is performed for each feature and distance type independently, so that for each feature type two distance functions are learnt (one for $d_{p\chi^2}$ and one for d_B). To get an impression about the descriptive power of each feature, some classification results are listed for each feature independently. To compute these results, the output of each net was thresholded at 0 to map the output to $[1, -1]$, and then compared to the ground truth.

Feature	$d = d_{p\chi^2}$ $\mu \pm S_E$.	$d = d_B$ $\mu \pm S_E$.
LBP3	0.7132 ± 0.0048	0.7036 ± 0.0051
LBP5	0.7228 ± 0.0047	0.7151 ± 0.0061
HoG	0.6824 ± 0.0047	0.6736 ± 0.0053
$\text{TPLBP}_{5,3}$	0.6981 ± 0.0046	0.6912 ± 0.0048
$\text{TPLBP}_{5,5}$	0.6872 ± 0.0050	0.6821 ± 0.0048

Table 1. Estimated mean accuracy μ and standard error of the mean S_E for $d_{p\chi^2}$ and d_B based on the different features (10-folded crossvalidation on the "pairs.txt" evaluation set).

4. FUSION OF THE DIFFERENT SIMILARITY CLASSIFIERS

After the separate similarity classifiers for each feature type are learnt, their results are merged to a final decision. All classifier outputs are fused to get the final answer if the image pair used as input shows the same faces or not. The fusion is performed by a Multi-Layer-Perceptron. The input layer is connected with the outputs of the single feature based classifiers of section 3. The number of input neurons is the number of similarity classifiers used. Because of the small number of input neurons in contrast to a much higher dimension of training material, the topology of this network can include one or more hidden layers as long as the number of connections between neurons is far smaller than the number of training examples.

In this paper, ten different similarity classifiers have been trained based on the two different distance measures for each HoG, LBP3, LBP5, $\text{TPLBP}_{5,3}$ and $\text{TPLBP}_{5,5}$. For fusion a MLP with two hidden layers is used.

5. RESULTS

In this section the result of the complete method is given and discussed. For evaluation, the benchmarking scheme of the "Labeled Faces in the Wild" database is used as suggested in the database description [2]. All experiments have been performed on the "funneled" dataset, in which image pairs are already coarsely aligned (see Fig. 1). A 10-fold cross-validation applying leave-one-out experiments with the given 10 datasets is done. Each set contains 300 image pairs showing the same individual and 300 image pairs showing different individuals. In each run, 9 datasets have been used for training the weights of both distance functions and fusion, the remaining set has been used to generate the results, that leads to 5400 image pairs for training and 600 image pairs for testing in each of the 10 experiments. An image restricted training scheme has been used, that means that context information, if a persons occurs in more than one image pair has not been used to create additional training data. Each of the originally 250×250 pixel sized images have been cropped from pixel

Feature	μ	S_E
this approach	**0.7367**	**± 0.0057**
3x3 Multi-Region Histograms[9]	0.7295	± 0.0055
Eigenfaces [12]	0.6002	± 0.0079
joint alignment [8]	0.7393	± 0.0049
MERL [4]	0.7052	± 0.0060
MERL+joint alignment [4]	0.7618	± 0.0058
Hybrid descriptor-based [7]	0.7847	± 0.0051

Table 2. Comparison to other methods: Estimated mean accuracy μ and standard error of the mean S_E of a 10-folded crossvalidation on the "pairs.txt" evaluation set.

position $(61, 61)$ to $(189, 189)$ to only use the inner part of the images, avoiding some influence of the background. The resulting cropped images have been divided into 9×9 non-overlapping patches to compute the histograms of features as described in section 2. After training of the distance function for each featuretype independently, results have been reported in table 1, the MLP for the fusion was chosen to have two hidden layers, the first with 4 neurons and the second with 2 neurons. This selection is empirical, probably other topologies will perform even better. All training processes have been done with resilient propagation (RPROP)[11] a well known variant of the backpropagation algorithm. For the proposed method, a mean accuracy μ of 73.67% has been achieved, for a comparison to other algorithms see Table 2.

6. CONCLUSION AND FUTURE PROSPECTS

Patchwise weighted similarity functions are a good way to approach the problem of unconstrained face recognition, a good accuracy in the given benchmark has been achieved. Experiments show, that HoG, LBP and TPLBG are useful features for this task, but of course additional features could further improve the results. The weights of the similarity functions can automatically be determined during a learning process on the training data, and the derived similarity functions generalize well enough to achieve good results if classifying unseen data. A fusion of similarity functions based on different features using a MLP slightly improves the overall accuracy, here further research on different fusion strategies could lead to better results. A critical ratio, the amount of training data in relation to the amount of weights to set up is an important factor during the training of neural networks. Different strategies of creating random variations of the training data could also lead to improvement. Another open, but important processing step, is an alignment step to find the best geometric transformation as a preprocessing, before the similarity classification is done. This is probably the most promising action for further enhancement of the performance. Results reported in [8] [4] have shown, that alignment greatly improves the overall accuracy.

7. ACKNOWLEGEMENTS

The work described in this paper was conducted within the EU-Collaborative Project PROMETHEUS, "Prediction and interpretation of human behavior based on probabilistic structures and heterogeneous sensors", and was funded by the European Union Division FP7-ICT Information and Communication Technologies.

8. REFERENCES

[1] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, pp. 399–458, December 2003.

[2] G.B Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," Tech. Rep., University of Massachusetts, 2007.

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

[4] E. Nowak and F. Jurie, "Learning visual similarity measures for comparing never seen objects," in *Proc. CVPR*, 2007, pp. 1–8.

[5] A. Ferencz, E. Learned-Miller, and J. Malik, "Learning hyper-features for visual identification," in *Neural Information Processing Systems*, 2004, vol. 18.

[6] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, 2006.

[7] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor based methods in the wild," in *Real-Life Images workshop, ECCV*, October 2008.

[8] G.B. Huang, V. Jain, and E.G. Learned-Miller, "Unsupervised joint alignment of complex images," in *Proc. ICCV*, 2007, pp. 1–8.

[9] C. Sanderson and B.C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference," in *Int. Conf. on Biometrics (ICB)*, 2009.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005, vol. 1, pp. 886–893.

[11] M. Riedmiller und H. Braun, "Rprop - a fast adaptive learning algorithm," in *Proc. of the Int. Symposium on Computer and Information Science VII*, 1992.

[12] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.