VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

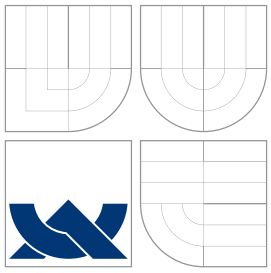
GENDER RECOGNITION FROM FACE IMAGES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

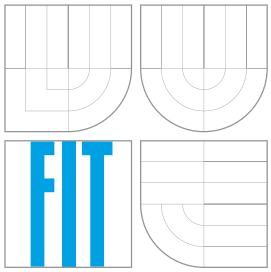
AUTOR PRÁCE
AUTHOR

MARIAN KALUŻA

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

ROZPOZNÁNÍ POHLAVÍ ČLOVĚKA NA FOTOGRAFII

GENDER RECOGNITION FROM FACE IMAGES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — MARIAN KALUŻA
AUTHOR

VEDOUCÍ PRÁCE — Ing. ADAM HEROUT, Ph.D.
SUPERVISOR

BRNO 2010

Abstrakt

Tato práce popisuje postup s využitím víceměřítkových fotografií pro rozpoznávání pohlaví podle obličeje. Koncept je založen na algoritmech jako "Histogram of Oriented Gradients" nebo "Local Binary Patterns". Experimenty ukázaly, že úspěšnost rozpoznání pohlaví se dá zvýšit nejenom s využitím více příznaků aplikovaných na jedno měřítko obrázku, ale také s využitím pouze jednoho příznaku aplikovaného na více rozlišení. Popsaný postup dosáhl více než 95% úspěšnosti rozpoznání u obou zvolených kolekcí obrázků.

Abstract

This paper presents a multiresolution approach for gender recognition based on Histogram of Oriented Gradients and Local Binary Patterns. The experiment showed that gender recognition accuracy can be improved not only by aquiring different features on the same image resolution but even by gathering just a single feature at different image scales. The presented approach is quite competitive with above 95% accuracy in both evaluated datasets.

Klíčová slova

Rozpoznání pohlaví, Histogram of Oriented Gradients, Local Binary Patterns, Včesná fúze, Pozdní fúze, Lineární regrese

Keywords

Gender recognition, Histogram of oriented gradients, Local binary patterns, Serial fusion, Parallel fusion, Linear regression

Citace

Marian Kałuża: Gender Recognition from Face Images, bakalářská práce, Brno, FIT VUT v Brně, 2010

Gender Recognition from Face Images

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Adama Herouta, Ph.D.

.....
Marian Kałuża
May 19, 2010

Poděkování

Rád bych poděkoval svému vedoucímu práce panu Adamu Heroutovi, který mě vedl správnou cestou, konzultoval se mnou problémy a pomohl mi celkově k vypracování této bakalářské práce.

Contents

Chapter 1

Introduction

Attention to methods in a gender recognition can be justified by many applications: improving search engines relevance accuracy, demographic data collection, gender-adaptive interface (software behavior changes due to the user's gender), adjustable advertisements with respect to the viewer's gender, etc..

The aim of this paper is to test the idea, proposed by Alexandre [1], that by fusing obtained decisions from just a single feature on different image resolutions can have comparable recognition accuracy like classical aproach, i.e. fusing different features on a single scale.

A few methods for feature fusion were tested to obtain the final result from feature decisions. Two naive methods were used: serial strategy where the final feature vector was concatenated from all feature vectors and then fed to the classifier, parallel strategy where the final result was achieved by a majority voting at feature decisions; and one more advanced method which use a linear logistic regression. Results from all methods for fusing features are compared later in section 5.

The experiments were performed on two publicly available datasets: FERET and UND. Quite much attention was recently given to FERET dataset, so results can be easily comparable with other works. UND dataset is quite new but according to Alexandre [1] this dataset can be used for an objective evaluation of the gender recognition accuracy.

The rest of the paper is organized as follows: the method principle and the features are described in chapter 2, datasets used and an image normalization in chapter 3. Results from classifier's evaluated separately in chapter 4, and feature fusion of all features in chapter 5. Section 6 concludes the paper.

Chapter 2

Gender Recognition by Multiscale Approach

This section presents some basic concepts, the methods used in gender recognition and features extraction.

2.1 Previous work

An objective comparison between different gender recognition approaches was difficult until recently because much of the published work was evaluated on non-replicable datasets and with various conditions used.

Mäkinen and Raisamo [7] made an effort towards improving the standards in the field by making available the details of one of the datasets used – FERET dataset. This allowed Mayo and Zhang [8] and Alexandre [1] to compare against the same dataset so it is also done in this paper.

Gutta et al. proposed [6] one of the first gender recognition solutions on a large dataset. They used an ensemble of RBF networks and C4.5 decision trees on FERET images which were manually segmented and normalized to 64x72 pixels. However, on set of 3006 images of 1009 subjects they obtained 96% accuracy, the same person could appear in both training and testing set so the classifier could learn to recognize faces instead of gender.

Moghaddam and Yang [10] also used the FERET database. They used an automatic face-processing system to normalize images, what compensated for translations and slight rotations. Evaluation used a 5-fold cross validation scheme. The presented accuracy was 96.62%, but the experiments were done using several images from the same subjects thus having the same problem as the work of Gutta et al. [6].

Mäkinnen and Raissamo [7] tested how the face alignment influenced the accuracy of gender classification. The best results (87.1%) were obtained using no automatic alignment with using 36x36 pixel size images.

Mayo and Zhang [8] showed that augmenting the training set can improve the generalization. They pointed that by introducing a rotated and translated version of the original images, the gender recognition accuracy could be increased: they improved the results of Mäkinen and Raisamo [7] on FERET dataset from 87.1% to 92.50%.

An AdaBoost based approach was 50 times faster than using SVM on the FERET images in Bajula and Rowley work [2]. The best accuracy reported was 94.40% on 20x20 images, with normalization similar to the one used by Moghaddam and Yand [10].

Alexandre [1] tested a multiscale approach on FERET and UND datasets. The best result obtained on the FERET dataset was quite high – 99.07% but the testing set used was slightly unbalanced, what can cause an irrelevant improvement or decrease recognition accuracy as shown later in chapter 3 in table 3.3.

2.2 Image recognition

Task of the recognition is to classify objects to the categories – in this case: a man or a woman.

Process of recognition is done in several steps: First, an image is given to the *feature*, what producess a *feature vector* from the input image. The *feature vector* is a vector of observations (measurments) and represents specific characteristics of the input image by numerical values. Second, a *learning machine* compute the objective function, which the best separate the recognition categories based on the training set pictures. Then, *features* are applied on the images from the testing set and obtained *feature vectors* are fed into the *classifier*, what decides to which category each input image belongs. Finally, the evaluation of the recognition accuracy is performed.

Features are described in sections 2.4 and 2.5, classifier in section 2.6.

2.3 Multiscale approach

A classical approach for image classification fuses several decisions obtained from different features at a single scale. This paper uses a multiscale approach for gender recognition. The main idea is to extract features on different image resolutions and fuse the obtained decisions.

According to that idea, the result obtained from fusing decisions from a single feature applied on several image resolutions should be at least comparable with the classical approach where extraction was done on several features at single scale.

To verify that idea there has to be used at least two features and two image resolutions in the experiment to compare these approaches. This paper uses shape and texture features and image resolutions: 256x384, 128x192, 64x96. Figure 2.3 shows the proposed approach.

The following section describes the features used in this paper.

2.4 Shape features

The shape feature used is a histogram of edge directions, similar to a histogram of oriented gradients used by Dalal and Triggs [4]. The main differences are that the input images are grayscale and no histogram normalization is performed.

Process is now described: First vertical and horizontal edge maps are found using the following masks: $[-1, 0, 1]$ and $[-1, 0, 1]^T$. Dalal and Triggs tested also different masks like:

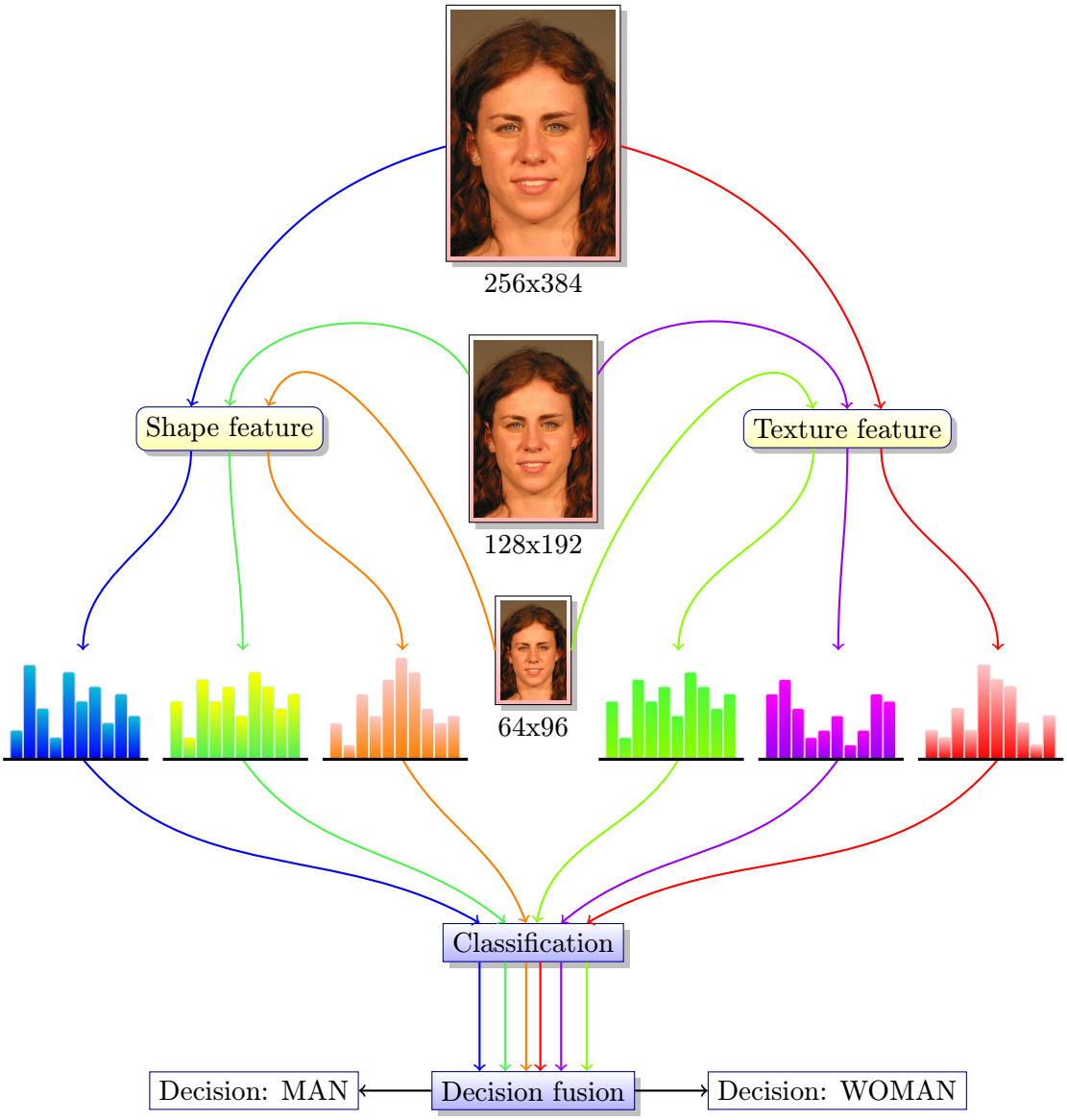


Figure 2.1: The proposed multiscale feature fusion approach. Each feature is applied on each image size and the obtained histograms are classified and fused.

various 1–D point derivatives (uncentered $[-1, 1]$, centered $[-1, 0, 1]$ and cubic-corrected $[-1, -8, 0, 8, 1]$) as well as 3x3 Sobel masks and 2x2 diagonal ones $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ (the most compact centred 2-D derivative masks). Simple 1-D $[-1, 0, 1]$ masks at $\sigma = 0$ work the best so they are used also in this paper.

For each pixel, the edge map direction is found using:

$$\theta = tan^{-1}\left(\frac{v}{h}\right) \tag{2.1}$$

and the weighted vote (in this case an intensity) is obtained with:

$$m = \sqrt{v^2 + h^2} \tag{2.2}$$

where v and h represent the vertical and horizontal edge values at a pixel, respectively.

The weighted vote from each pixel is a function of vertical and horizontal edge maps representing here the magnitude of a given pixel. Dalal and Triggs [4] tested other variations of weighted vote functions like: gradient magnitude, magnitude itself, its square, its square root, or a clipped form of magnitude representing soft presence/absence of an edge at the pixel. Using magnitude itself gives the best results.

For each pixel a weighted vote obtained from (2.2) is accumulated into orientation bins over a region called window. It means in that case that each pixel adds its edge magnitude m to the bin that corresponds to its edge direction θ. The orientation bins are discretized to 18 degrees intervals, such that the histogram contains 20 bins to cover the full range of 360 degrees. Picture 2.2 shows an example of a histogram in polar coordintates representing 360^o.

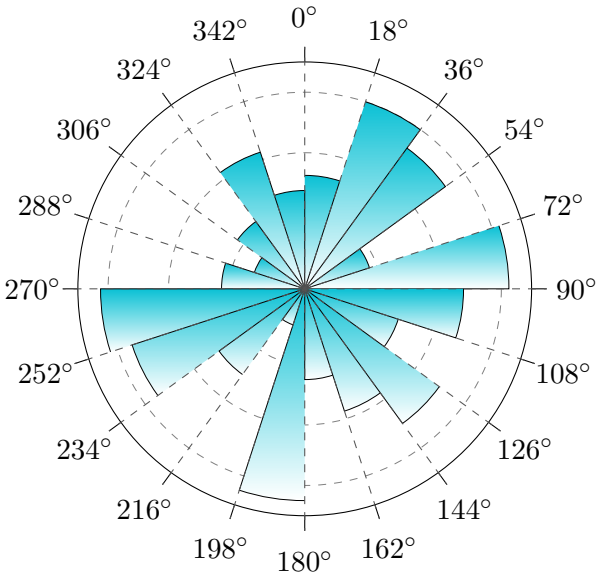


Figure 2.2: Example of a histogram of 20 bins in polar coordinates representing 360^o. A face image is then represented by a vector of 20 numbers.

An image is then represented by a vector of $20n$ real values by concatenating n individual histograms, where n represents the number of windows used on the whole image.

2.5 Texture features

Texture features have to deal with real world textures which have often variations in orientation, uneven illumination or great within-class variability. Good example of a texture feature which is not much affected by these complications can be Local Binary Patterns as proposed by Ojala et al. [11].

Let the definition of a texture T will be in a local neighborhood of a monochrome texture image as the joint distribution of the gray levels of P image pixels:

$$T = t(g_c, g_0 - g_c, g_1 - g_c, \ldots, g_{P-1} - g_c) \tag{2.3}$$

where g_c corresponds to the gray value of the center pixel of the local neighborhood and $g_i\,(i = 0, \ldots, P-1)$ correspond to the gray values of the pixels on a circle of radius $R\,(R > 0)$ that form a circulalry symmetric neighbor set. Figure 2.3 illustrates the circular neighbor sets for various (P,R). The gray values of neighbors which do not fall exactly in the center of pixels are estimated by interpolation.

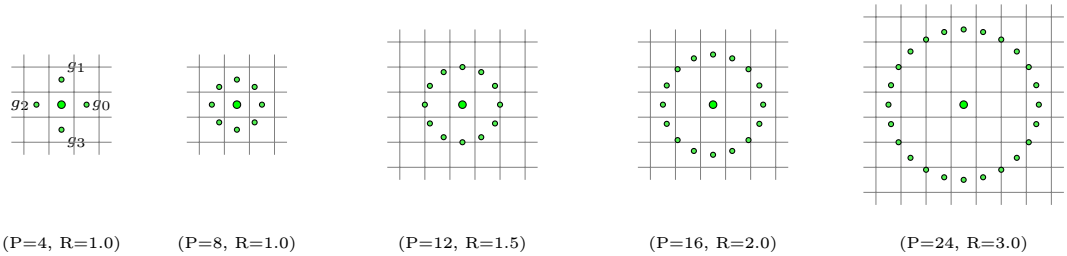


Figure 2.3: Circularly symmetric neighbor sets for different (P,R)

This descriptor is invariant to changes in mean luminance; hence, the joint difference distribution is invariant against gray-scale shifts. For constant regions, the differences are zero in all directions. On an edge, the descriptor records the highest difference in the gradient direction and zero values along the edge and, for a spot, the differences are high in all directions.

To achieve invariance with respect to the scaling of the gray scale by considering just the signs of the differences instead of their exact values:

$$T = t(s(g0 - g_c), s(g_1 - g_c), \ldots, s(g_{P-1} - g_c)) \tag{2.4}$$

where

$$s(x) = \begin{cases} 1 & \text{for x} \geq 0 \\ 0 & \text{for x} < 0. \end{cases} \tag{2.5}$$

A unique value can be obtained from the histogram of binary values from 2.6 by multiplying each term by a binomial coefficient. A unique $LBP_{P,R}$ number characterizes the spatial structure of the local image texture:

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^P \tag{2.6}$$

The $LBP_{P,R}$ operator is by definition invariant against any monotonic transformation of the gray scale, i.e., as long as the order of the gray values in the image stays the same, the output of the $LBP_{P,R}$ operator remains constant.

The $LBP_{P,R}$ operator produces 2^P different output values, corresponding to the 2^P different binary patterns that can be formed by the P pixels in the neighbor set.

Ojala et al. [11] introduced a notion of uniform patterns: an $LBP_{P,R}$ binary code is called "uniform" if it contains at most two transitions from 0 to 1 or 1 to 0, considering the code circularly.

Ojala et al. also present a version of descriptor that is invariant to rotation of the patterns, labeled "riu2" as opposed to the "u2" variant just described. Face images are automatically normalized so experiments use standart "u2" descriptor.

The number of uniform patterns when considering $LBP^{u2}_{8,1}$ is 60 (58 patterns with two transitions and 2 with no transition) and thus, each $LBP^{u2}_{8,1}$ produces an histogram with 60 bins. Picture 2.4 shows an example of $LBP_{8,1}$ histogram:

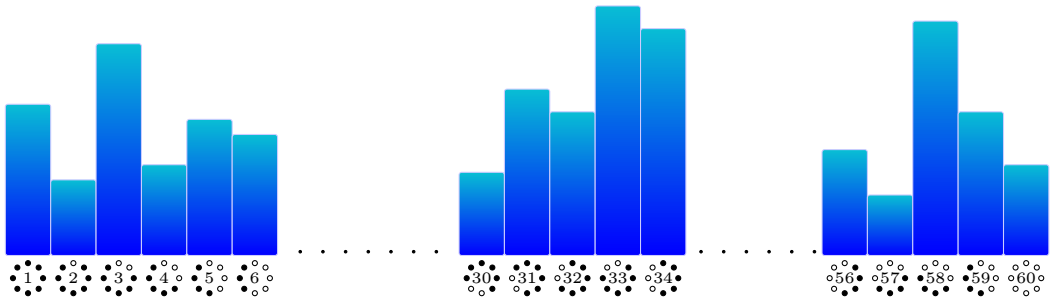


Figure 2.4: An example of $LBP_{8,1}$ histogram. Bins 1 and 60 have no transitions, the others have two.

2.6 Classifier

There were used a support vector machine (SVM) as learning machine of proven usefulness in similar classification works [1].

Support vector machine computes from a set of training examples, each marked as belonging to one of two categories, a model that predicts whether a new example falls into one category or the other.

Fixed parameter C = 100 (the trade off between the training error and the margin) was used because the aim of this paper is to focus attention on the features rather than on the classifier and its variable optimization.

The implementation used, LIBSVM [3], is freely available.

2.7 Testing features

For the learning purposes with SVM learning machine an easy classification experiment was performed. Testing the accuracy of features was done on recognizing ellipses from rectangles. These elements can be easily generated and are easily recognized due their edge angles.

For testing purposes the generator for statictically relevant amount of images was created. This generator can create any number of simple elements like ellipses or rectangles. The generator is a part of appendix A.3.

Examples of elements created by the generator are shown in table 2.1.

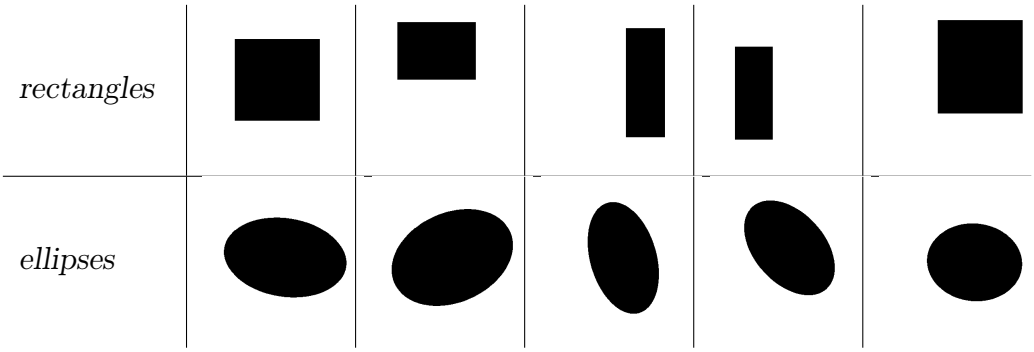


Table 2.1: Examples of generated elements

The experiment was done on a statistically relevant number of images. Machine learning was done on the training set containing 500 ellipses and rectangles. Accuracy evaluation of the shape feature was tested on the set containing 1000 rectangles and ellipses.

Table 2.2 shows result of this experiment. Column 2 shows the number of correctly recognized shape item from testing set of size 2000. Column 3 shows accuracy in percentage.

Window size	**Recognized**	**Accuracy**
400	2000	100.00
200	2000	100.00
100	1950	99.50
50	1975	98.75
20	1935	96.75

Table 2.2: Serial strategy fusion accuracy in %

Chapter 3

Datasets

The FERET dataset defined by Mäkinnen and Raisamo [7] was used as one of the two evaluated datasets. Several works used the FERET dataset so this allowed to compare the performance of proposed method with other works [6], [10] or [8].

The experiments were performed also on the second dataset: Collection B from UND dataset by Flynn et al. [5]. This dataset has total 487 images. No person has two images in the dataset: this prevents the classifiers to recognize persons instead of gender.

Face segmentation and normalization was obtained with the OpenCV 2.0. Each file in both datasets has its *.gnd* file where are included eye, mouth and nose coordinates. Face normalization is a process where each image was rotated, scaled and translated that way that all images have eyes at the same position. Mouth and nose was not necessarily at the same position due to variations of the human face. Hair is not included in the image because it can distort the classifier from recognizing gender. Only face should be important to the classifier. Process of a face normalization shows picture 3.1.

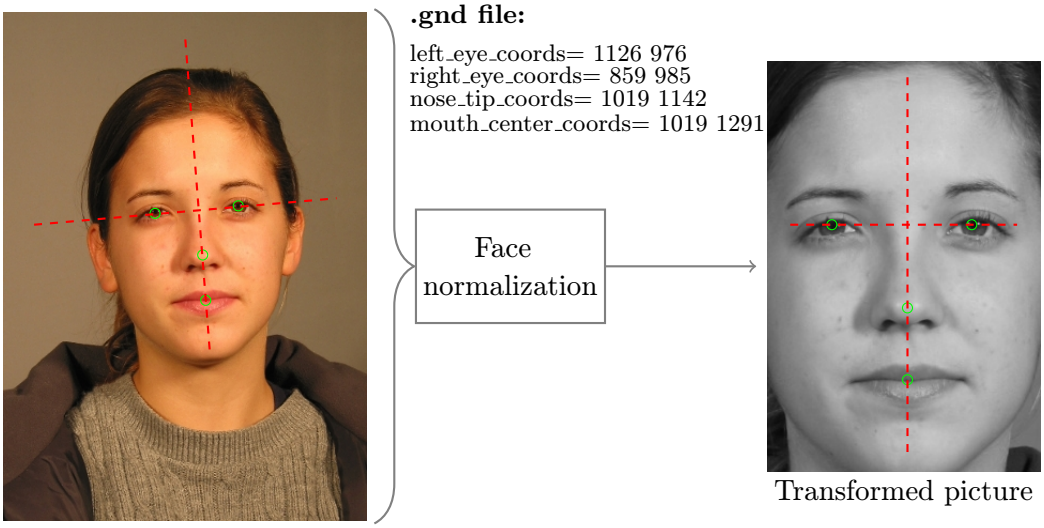


Figure 3.1: Face normalization process.

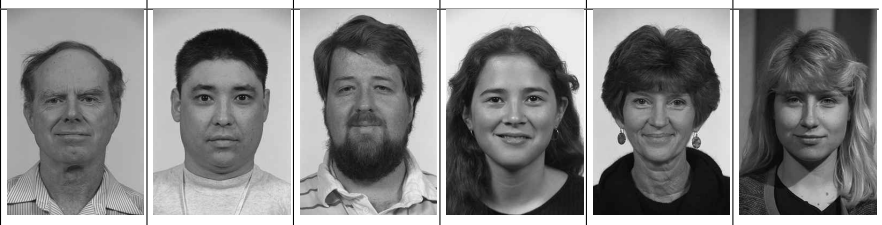
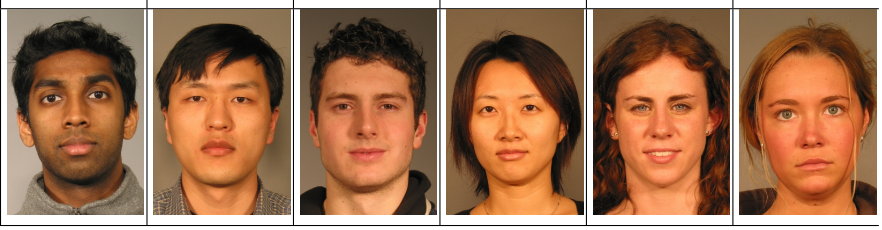
FERET train set	
FERET test set	
UND train set	
UND test set	

Table 3.1: Sample images from both datasets.

These images, as the ones in the above mentioned FERET set, contain persons with different face aspects, such as Asian, Causcasian and African. A few persons wear some elements that do not relate to the face itself like glasses, hats or caps. Each person has several images in the dataset so the one without glasses and caps was chosen. If no image of any person without glasses or caps are found person was not included in the experiment.

	Total number	**Train**		**Test**	
		Male	Female	Male	Female
FERET	734	292	264	89	89
UND	477	211	100	83	83

Table 3.2: Number of images used in both datasets.

For objective comparison it is important to have the test set equally balanced, i.e, the number of men and women images have to be equal. In some cases when a classifier favorized men or women and test set are not equally balanced the recognition accuracy can be unobjectively interpreted.

Men	Women	Recognized men	Recognized women	Accuracy
59	89	53	71	83.78
64	89	57	71	83.66
69	89	62	71	84.17
74	89	67	71	84.66
79	89	72	71	85.11
84	89	77	71	85.54
89	89	82	71	85.95
89	84	82	67	86.12
89	79	82	63	86.30
89	74	82	61	87.73
89	69	82	57	87.97
89	64	82	54	88.88
89	59	82	50	89.19

Table 3.3: Serial strategy fusion accuracy in %

Table 3.3 shows an example of obtained decision from shape feature applied on 128x192 image size with 32x32 window size on FERET. Classifier in this case favorized men. The table shows how accuracy is changing on an unbalanced test set. In equally balanced test set the gender recognition accuracy is 85.95%. This accuracy can have almost 4% improvement, because test set contains 30 women image picture less than men images (last row).

Chapter 4

Classifiers Evaluated Separately

4.1 Sliding windows

If the feature is applied on the whole image, it will lose plenty of information by summing all together into one histogram. The original image was divided into small windows which cover the full image size.

Features were evaluated sequentially on each window in the image so information summing was done in a limited way. The final feature vector for a machine learning is obtained by concatenating feature vectors from all windows in the image and the order of sliding windows is kept the same so the position in the final histogram indicates the position in the original image. Concatenation of the final vector is shown in the image 4.1.

A special variant of overlapping windows was used. In this case the windows have 50% overlap both vertically and horizontally. This special variant was marked with "ov" in tables 4.1 and 4.2.

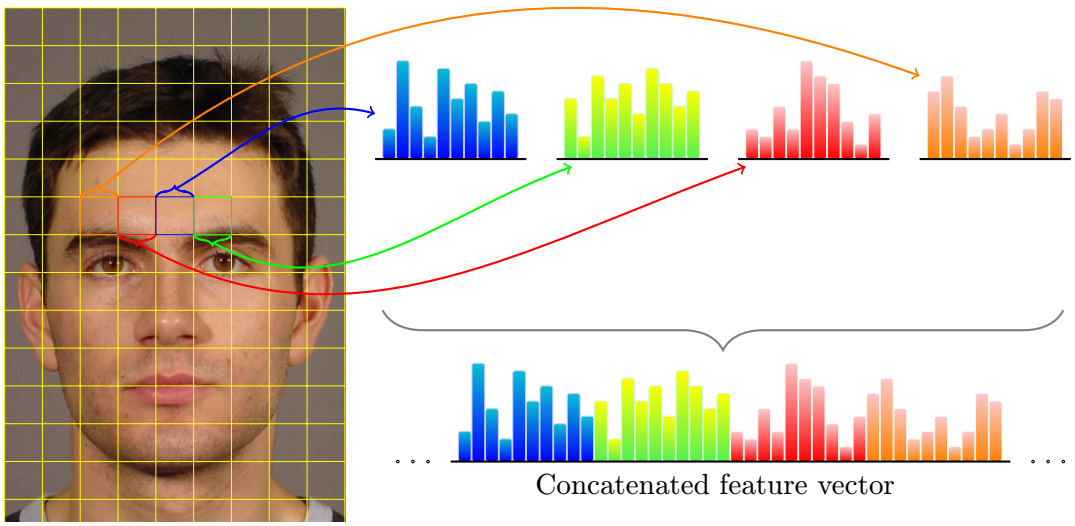


Figure 4.1: Final feature vector is concatenated from all windows.

4.2 Shape features

The first set of results evaluates the gender recognition performance using the shape features. The results obtained are presented in table 4.1.

The best results were obtained for FERET dataset using images of 64x96 and window size of 8x8. The results obtained from images of 256x384 have quite poorer performance but show higher level of stability: 89.88% – 93.25% instead of 73.59% – 94.94% in the case of 64x96 image size.

For UND dataset the best result was obtained with using 128x192 image size and 8x8 window size. Image size 64x96 show also lower result stability than other two image results: 75.30% – 91.57%.

The results showed no differences between overlapped and non-overlapped windows. The accuracy of recognition is always the same for both window versions for both of the datasets. Image pixels in the shape feature are evaluated separately (no neighborhood used) so the information evaluated from all pixels does not change in cases of different window positions.

Image size	Window size	FERET	UND
64x96	4x4	94.94	91.57
	4x4ov	94.94	91.57
	8x8	95.50	89.16
	8x8ov	95.50	89.16
	16x16	88.20	88.55
	16x16ov	88.20	88.55
	32x32	73.59	75.30
	32x32ov	73.59	75.30
128x192	4x4	92.69	88.55
	4x4ov	92.69	88.55
	8x8	93.82	93.98
	8x8ov	93.82	93.98
	16x16	87.64	92.77
	16x16ov	87.64	92.77
	32x32	85.95	88.55
	32x32ov	85.95	88.55
256x384	4x4	89.88	87.35
	4x4ov	89.88	87.35
	8x8	92.69	89.16
	8x8ov	92.69	89.16
	16x16	93.25	92.17
	16x16ov	93.25	92.17
	32x32	92.69	92.77
	32x32ov	92.69	92.77

Table 4.1: Shape feature accuracy in %

4.3 Texture features

The results of the LBP features evaluation are in table 4.2.

Globally, the texture feature gives quite better results than shape feature. Unfortunately, this feature has much less stability. In some cases, the learning machine was not able to classify at all – for example all images was classified as a man. Texture features requires 60 bins for a window instead of 20 in the case of shape features. It means that the texture feature requires much more system resources and time to compute.

The best result obtained for FERET dataset was 96.62% in several cases. The best result 96.62% was obtained by using 256x384 image size and 16x16 window size with overlap. In the case of the UND images, the best result 96.99% was obtained in two cases.

In this case the results showed a difference between overlapped and non-overlapped window versions. In some cases even more than a slight difference: 76.40% to 91.57% in case of 4x4 window on 64x96 image in FERET dataset. Almost always the overlapped window version was better than non-overlapped.

Image size	Window size	FERET	UND
64x96	4x4	76.40	-
	4x4ov	91.57	86.75
	8x8	91.57	93.98
	8x8ov	95.50	96.99
	16x16	94.38	93.37
	16x16ov	95.50	96.39
	32x32	87.08	89.16
	32x32ov	92.70	92.77
128x192	4x4	89.32	81.93
	4x4ov	-	90.36
	8x8	94.94	95.18
	8x8ov	95.50	94.58
	16x16	94.94	93.98
	16x16ov	95.50	95.78
	32x32	92.69	92.17
	32x32ov	92.69	95.78
256x384	4x4	-	93.37
	4x4ov	-	93.98
	8x8	92.13	93.37
	8x8ov	93.98	91.56
	16x16	95.50	95.78
	16x16ov	96.62	93.98
	32x32	93.82	96.99
	32x32ov	94.94	93.98

Table 4.2: Texture feature accuracy in %

Chapter 5

Feature fusion

This section presents the results of fusion across image sizes and feature types. Three methods for fusion of separate feature decisions are tested: two naive methods serial and parallel strategy and one more sophisticated method as linear logistic regression. Results of each method are presented later in this section.

5.1 Serial strategy

Serial strategy performs the fusion before machine learning. Feature vectors obtained from each feature evaluation are concatenated into "final fused feature vector". Process of machine learning is done then on the whole concatenated vector.

$$F_{final} = [\, F_1 \,|\, F_2 \,|\, \ldots \,|\, F_n \,] \tag{5.1}$$

where F_{final} is final concatenated feature vector and $F_i\,(i = 0, ..., N)$ corresponds to the feature vector obtained from each of N evaluated features.

Partial results of this fusion method are shown in table 5.1:

Image size	**Shape feature**	**Texture feature**
64x96	92.69	94.94
128x192	93.82	95.50
256x384	93.25	-

Table 5.1: Serial strategy fusion accuracy in %

Serial fusion strategy shows rather stability among all feature decisions rather than accuracy improvement. Fused accuracy is only slightly better than an average of all feature decisions. Nevertheless, concatenating the feature vectors into the final vector is very complex and the obtained feature vector is too long for a practical usage.

Final feature vector is too long even for concatenation texture feature vectors of FERET images on image size 256x384. Concatenating longer final vectors like fusing two features is not possible because of lack of the system memory.

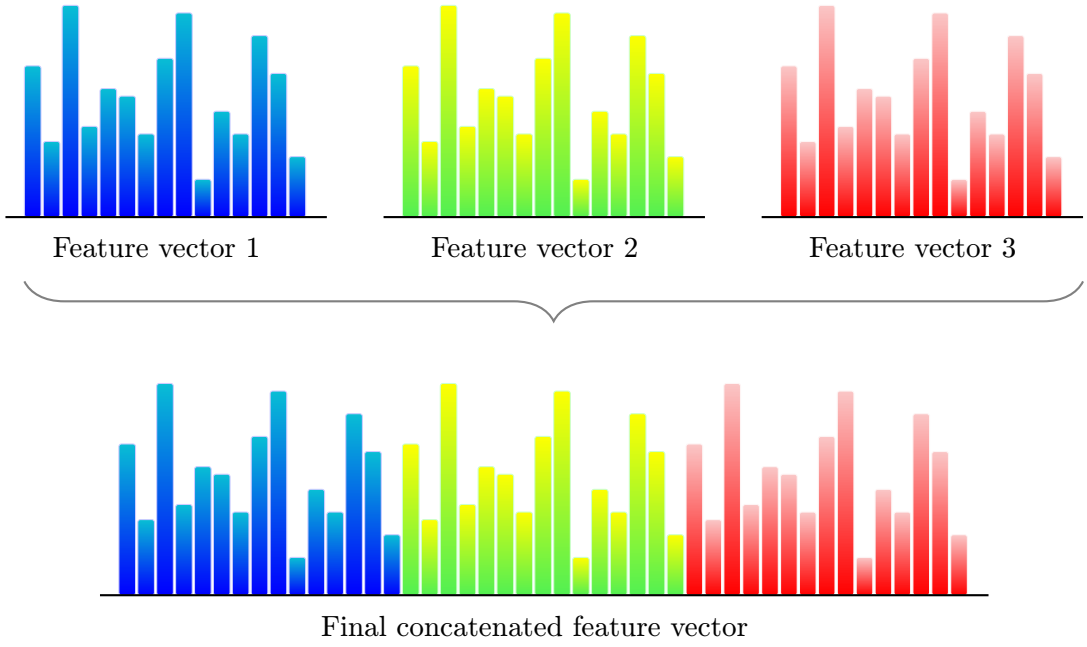


Figure 5.1: Graphical representation of serial strategy fusion.

Serial fusion strategy requires relatively much resources and has only slight accuracy improvement.

5.2 Parallel strategy

Parallel strategy fuses the feature decisions instead of feature vectors. Each of these decisions can be obtained from classifiers trained with different features or with the same features. In any case, the classifiers should make their errors mostly on different patterns.

Fusion is done from the decisions of different classifiers trained either with different types of features at a single scale or with the same feature captured at different scales.

A majority rule is used for the decision process. Each decision is a integer label (1 for man, 0 for woman). Then, majority vote decision is given by:

$$d_m = \begin{cases} man & \text{for} \quad \sum_{i=0}^{m} d_i > m/2, \\ woman & \text{for} \quad \sum_{i=0}^{m} d_i < m/2 \end{cases} \tag{5.2}$$

where d_i is a decision of each of m classifiers. When m is even and there is a tie, decision is made randomly.

The results of parallel strategy are shown in table 5.2.

First all decisions were fused from features on each window size for each database. The results are in the first three lines of table 5.2. The fused results do not improve the best result from tables 4.1 and 4.2 but are among the best.

Then all decisions from each feature were fused at each window resolution. The results are in the fourth and fifth line of table 5.2. Shape feature shows a marginal improvement

Fusion	FERET	UND
64x96	95.50	94.58
128x192	94.38	95.18
256x384	94.38	93.97
Shape	95.50	93.37
Texture	92.13	95.78
All	**96.06**	**95.78**

Table 5.2: Parallel strategy fusion accuracy in %

or no improvement: for FERET dataset no change – 95.50% and 93.98% versus 93.37% for UND dataset. For texture feature, the fusion yielded no improvement or slight underperfomance: 95.50% versus 92.13% for FERET dataset and no change – 95.78% for UND dataset.

When comparing the results obtained by fusing the decisions from two features at a given image size (lines 1, 2 and 3) with the results obtained by fusing decisions from different image sizes for each feature type (line 4 and 5) it can be seen that these aproaches are quite competitive. This points to the fact that the information from different scales, even if just from a single feature, can be as much important as different features at a single scale.

Thirdly, the final results were obtained by fusing all decisions from all independent sets and this approach obtained the best results for both datasets (last line in table 5.2).

In the case of the FERET dataset, the best obtained results by using majority voting was 96.06% what gives 7 wrongly classified images out of 172 images in the test set.

In the case of the UND dataset, using all features with majority voting yields also 7 classification errors but out of 166 test images, which amounts to 95.78% accuracy.

5.3 Linear logistic regression

Parallel fusion strategy described in 5.2 considers just best selected class from each of the classifier. Linear logistic regression instead, considers all values for different classes.

The logistic regression model uses the predictor variables (categorical or continuous) to predict the probability of each class. It means that the feature output is a probability of belonging to each of the classes.

For this purpose, the training set is split into a part used for training the individual classifiers and a part used for training the fusion by linear logistic regression [9].

For calculating the probability of each class from the function value instead of a single decision SVM had to be modified. There is an altered version of program `svm-predict` on dvd as appendix A.7.

This approach should give better variability on feature decisions than the best selected class. Table 5.3 shows the results of this fusion strategy.

Fusion	**FERET**	**UND**
64x96	94.38	90.96
128x192	94.94	89.16
256x384	94.38	90.96
Shape	87.08	87.35
Texture	88.76	93.37
All	93.26	89.16

Table 5.3: Linear regression fusion accuracy in %.

Interesting is the fact that fusion of all features is not the best (for FERET dataset: the 128x192 fusion 94.94% is better than the fusion of all 93.26%, and for UND dataset: the Texture fusion 93.37% is better than the fusion of all 89.16%). Nevertheless, the final accuracy is worse than in the parallel fusion strategy presented in table 5.3. Based on that experiment simply using just binary decisions is better than linear logistic regression.

Chapter 6

Conclusions

This paper presented the way of gender recognition from face images that shows how fusion aproach based on features from different scales can improve gender recognition accuracy.

The evaluation used two features: "Histogram of Oriented Gradients" and "Local Binary Patterns", three different image sizes, four window sizes with and without overlap and on two different datasets.

The proposed approach requires previous geometric face normalization. Images are transformed so eyes are always in the same position.

This paper showed that this approach have comparable gender recognition accuracy with other works: For the FERET dataset: 96.06% against Mayo and Zhang [8] published result 92.58%, and Alexandre [1] 99.06%. In the case of the UND dataset: 95.78% against Alexandre's [1] 91.19%.

Based on performed experiments conclusion is that fusing information from different scales, even if just from a single feature, can have comparable result improvement as a classical approach, i.e fusing from more features on a single scale.

Future work includes to deal with greater noise in the images like uneven luminosity or items not related to the face directly like: glasses, hair or caps. Testing the usefullness of the proposed aproach includes its application to other face datasets.

Finally, althought the current approach has proven usefullness in gender recognition on frontal face images, human face can be out of frontal angle or only part of the face would be visible so including automatic geometric normalization and a parts based model with a greater degree of local spatial invariance would help to improve the detection results in more general situations.

Bibliography

[1] Alexandre, L. A. Gender recognition: A multiscale decision fusion approach. *Pattern Recognition Letters*. 2010, In Press, Corrected Proof. pp. –. Available at: <http://www.sciencedirect.com/science/article/B6V15-4YFTKVV-2/2/625c16867459dce7790e63a24f428f10>. ISSN 0167-8655.

[2] Baluja, S., Henry and Rowley, A. Boosting sex identification performance. *In: AAAI*. 2005. pp. 1508–1513.

[3] Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines. 2001. Available at: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.

[4] Dalal, N. and Triggs, B. Histograms of Oriented Gradients for Human Detection. *In: Computer Vision & Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. 2005, Vol. 1. pp. 886–893. Available at: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360>.

[5] Flynn, P. J., Bowyer, K. W. and Phillips, P. J. Assessment of time dependency in face recognition: an initial study. In *AVBPA'03: Proceedings of the 4th international conference on Audio- and video-based biometric person authentication*. Berlin, Heidelberg: Springer-Verlag, 2003. pp. 44–51. ISBN 3-540-40302-7.

[6] Gutta, S., Wechsler, H. and Phillips, P. J. Gender and Ethnic Classification of Face Images. In *FG '98: Proceedings of the 3rd. International Conference on Face & Gesture Recognition*. Washington, DC, USA: IEEE Computer Society, 1998. pp. 194. ISBN 0-8186-8344-9.

[7] Mäkinen, E. and Raisamo, R. An experimental comparison of gender classification methods. *Pattern Recogn. Lett.* 2008, Vol. 29, n. 10. pp. 1544–1556. ISSN 0167-8655.

[8] Mayo, M. and Zhang, E. Improving face gender classification by adding deliberately misaligned faces to the training data. In. 2008. pp. 1–5.

[9] Minka, T. P. *A comparison of numerical optimizers for logistic regression*. 2003. Available at: <http://research.microsoft.com/minka/papers/logreg/>.

[10] Moghaddam, B. and Yang, M.-H. Learning Gender with Support Faces. *IEEE Trans. Pattern Anal. Mach. Intell.* 2002, Vol. 24, n. 5. pp. 707–711. ISSN 0162-8828.

[11] Ojala, T., Pietikäinen, M. and Mäenpää, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2002, Vol. 24, n. 7. pp. 971–987.

Appendix A

Obsah DVD

A.1 Dataset images

Face images used in the experiments for both dataset: *feret-images* and *und-images*. Folders are organized as follows:

- **men** Original men pictures from the dataset.
- **women** Original women pictures from the dataset.
- **train_men** Normalized men pictures for training set.
- **train_women** Normalized women pictures for training set.
- **test_men** Normalized men pictures for testing set.
- **test_women** Normalized women pictures for testing set.

A.2 Feature decisions

Decisions from all used features.

A.3 face-generateelements

Generate any number of simple elements like an ellipse or a rectangle.

A.4 face-rotate

Proceed translation and slight rotations so all face images have eyes in the same position.

A.5 face-fusion

Fuse feature vectors into one final feature vector. Used in serial strategy fusion.

A.6 face-train

Scripts for automating the learning process.

A.7 svm-predict

Alternated program svm-predict for returning the function value instead of selected class.

Appendix B

Manuals

B.1 face-generateelements

Generate any number of simple elements like an ellipse or a rectangle.

Program arguments:

--width N	specifies the width of the output images
--height N	specifies the height of the output images
--circle or -C	generate circles
--ellipse or -E	generate ellipses
--rectangele or -R	generate rectangles
--output or -o	output prefix for generated elements
N	number of generated images

example:

`./face-generateelements -E -n 100 --width 200 --height 200 /media/data/ellipses`
Generate 100 ellipses of size 200x200 into folder /media/data/ellipses.

B.2 face-rotate

Proceed translation and slight rotations so all face images have eyes in the same position.

Program usage: `./face-rotate gnd-file input-file output-file` where:

- **gnd-file** File containing the eye, mouth and nose coordinates
- **input-file** Image input file
- **output-file** Normalized image file

B.3 face-fusion

Fuse feature vectors into one final feature vector.

Program usage: `./face-fusion fused-file number-of-lines file1 file2 ...` where:

- **fused-file** Output fused file containing the final feature vector
- **number-of-lines** Number of items in the set
- **file1** File containing the feature vector