2D Ear Classification Based on Unsupervised Clustering

Anika Pflug, Christoph Busch
Hochschule Darmstadt
{anika.pflug, christoph.busch}@cased.de

Arun Ross
Michigan State University
rossarun@msu.edu

Abstract

Ear classification refers to the process by which an input ear image is assigned to one of several pre-defined classes based on a set of features extracted from the image. In the context of large-scale ear identification, where the input probe image has to be compared against a large set of gallery images in order to locate a matching identity, classification can be used to restrict the matching process to only those images in the gallery that belong to the same class as the probe. In this work, we utilize an unsupervised clustering scheme to partition ear images into multiple classes (i.e., clusters), with each class being denoted by a prototype or a centroid. A given ear image is assigned class labels (i.e., cluster indices) that correspond to the clusters whose centroids are closest to it. We compare the classification performance of three different texture descriptors, viz. Histograms of Oriented Gradients, uniform Local Binary Patterns and Local Phase Quantization. Extensive experiments using three different ear datasets suggest that the Local Phase Quantization texture descriptor scheme along with PCA for dimensionality reduction results in a 96.89% hit rate (i.e., 3.11% pre-selection error rate) with a penetration rate of 32.08%. Further, we demonstrate that the hit rate improves to 99.01% with a penetration rate of 47.10% when a multi-cluster search strategy is employed.

1. Introduction

Classification involves assigning a class label to a subject based on features extracted from the subject's biometric data. The number of classes is usually much smaller than the number of subjects in the gallery database and each subject is typically assigned to exactly one class. Class labels can either be based on anatomical properties of the observed biometric characteristic or on inherent structural or geometric properties of the biometric sample.While classification and/or indexing techniques have been developed for fingerprints [16, 20], iris [10, 17] and face [19, 22], the possibility of classifying ear images has received limited attention in the biometrics literature. To the best of our knowledge, this is the first work on automated unsupervised classification of ear images.

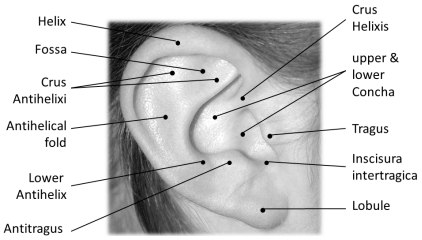


Figure 1: Morphology of the outer ear.

In this work, we explore the possibility of clustering 2D ear patterns into multiple categories based on their texture and structure. The texture is captured by the use of a texture descriptor, while local histograms capture the structure of the ear. We used *texture-based* features rather than explicit *shape-based* features because (a) extracting *shape* information from 2D ear images is a challenging problem, that often requires a highly constrained capture scenario [1], and (b) the discriminability of shape-based features is limited in low-quality 2D images.

One of the earliest work on ear classification was done by Ianerelli [11], where he classified ear images into 4 categories - round, oval, triangular, rectangular - based on a *visual* assessment of the ear. However, this classification process is difficult to automate due to the subjective nature of the assessment process. Further, as noted in [7], the number of members in each class is unevenly distributed.

In previous work, Khorsandi and Abdel-Mottaleb [13] categorized ear images into two groups - male and female - based on Gabor Filters and Sparse Representation. Motivated by their work, we aim to further explore the capability of texture descriptors for ear classification. In this regard we analyze commonly used texture descriptors, viz. Histograms of Oriented Gradients (HOG) [4], unified Local Binary Patterns (uLBP) [18] and Local Phase Quantization (LPQ) [2]. LBP and HOG have already been successfully used in the context of ear *recognition* [3, 5, 9]. LPQ has recently been used in face recognition, where it is shown to

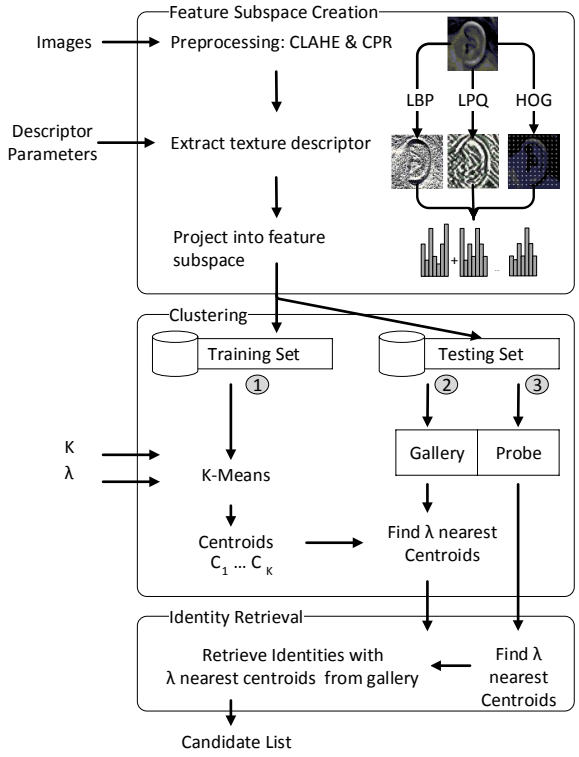
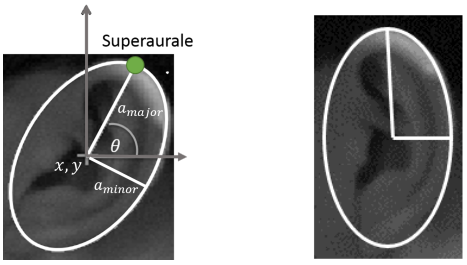


Figure 2: Illustration of the clustering scheme with preprocessing, cluster assignment and identity retrieval. The input parameters for each stage are shown on the left. The output is a list of possible gallery candidates to search.

be more robust to blur than LBP [2]. For an elaborate survey of features used for ear recognition, we refer the reader to [1].

In this work, we use unsupervised clustering in conjunction with texture-based local histograms to discover classes of ear patterns. Instead of using pre-defined labels such as triangular, oval, etc. [11], we deduce clusters based on the distribution of texture features in a high-dimensional space. Although this approach may not result in classes that can be trivially interpreted by a human, it allows us to circumvent ambiguities in class label assignment and results in classes with more evenly distributed numbers of members. The extraction of shape features can be complex and time consuming. Our goal is to use simple features that can be generated quickly and that do not bear the risk of error in the feature extraction process.

The primary contributions of this work are (a) an analysis of the clustering tendencies of feature spaces corresponding to 3 different texture descriptors; (b) a detailed experimental evaluation demonstrating the benefits of the proposed clustering approach for ear classification; and (c) a method for fusing the outputs of multiple classification schemes.



(a) Fitted Ellipse (b) Normalized Ear

Figure 3: Illustration of the CPR-based geometrical normalization of ear images. We fit an ellipse that encloses the ear, and rotate the whole image such that the major axis of the ellipse is vertical.

2. Clustering 2D Ear Patterns

The proposed approach has two distinct phases: the training phase and the classification phase. The *training phase* has two stages: (a) feature subspace creation, where texture-based feature vectors are extracted from a set of training images in order to define a feature subspace; and (b) cluster generation, where unsupervised k-means clustering is used to discover clusters in this feature space, with each cluster being denoted by its centroid. It must be noted that the subjects whose images were used in the training phase are not used in the testing/classification phase.

The *classification phase* has two stages: (a) gallery classification, where each gallery image in the database is projected onto the feature subspace created in the training phase and assigned a class label (i.e., a cluster index) based on the minimum distance criteria; and (b) identity retrieval, where the class label of an input probe image is first computed and a matching identity is obtained from the database by restricting the search to only those gallery images associated with this class label. Below we describe each phase in more detail. An overview of the complete system, including feature space creation, cluster assignment for gallery images and identity retrieval for probe images is given in Fig. 2.

All training and test ear images are pre-processed, in order to remove the influence of rotation, scale and illumination. We first adjust the contrast by using CLAHE [24]. Subsequently, we crop and geometrically normalize the images. This is done by applying cascaded pose regression (CPR) [6]. Using CPR, we train a classifier that fits an ellipse to the ear region, such that the ear is fully enclosed by the ellipse (see Fig 3). We then normalize rotate the cropped ear image, such that the major axis of the ellipse is vertical.

Finally, all images are resized to 100×100 pixels in order to compensate for different resolutions and to facilitate the extraction of local histograms in the subsequent step. We

experiment with three different texture descriptors, namely LBP, LPQ and HOG.

3. Training Phase

3.1. Feature Subspace Creation

Uniform Local Binary Pattern (uLBP): uLBP [18] encodes local texture information on a pixel level by comparing the grey level values of a pixel to the grey level values in its neighbourhood. The size of the neighbourhood is defined by a radius around the pixel g_j, which is at least 1 (for a neighbourhood having 8 pixels). Every pixel q_i within the radius that has a larger grey level value than the center pixel is assigned the binary value 1, whereas all pixels with a smaller grey level value are assigned the binary value 0.

The binary values of the neighborhood pixels are concatenated to form a binary string corresponding to the center pixel. Only those binary strings which have at most two bitwise transitions from 0 to 1 (or vice-versa) are considered - there are 58 such strings. Each binary string is then mapped to a value between 0 and 58 (the first 58 bins correspond to the uniform binary strings, and the 59-th bin corresponds to the rest). The uLBP-based ear descriptor is computed by sliding a window of a predefined size and overlap (step size in pixels) in the horizontal and vertical direction over the LBP image. From each sub window a local histogram with 59 bins is extracted.

The final descriptor is the concatenation of each local histogram. For a window size of 20×20 pixels and an overlap of 10 pixels, this results in a feature vector of dimension 3776.

Local Phase Quantization (LPQ): The concept behind LPQ [2] is to transform the image into the fourier domain and to only use the phase information in the subsequent steps. Given that a blurred image can be viewed as a convolution of the image and a centrally symmetric point spread function, the phase of a transformed image becomes invariant to blur. For each pixel in the image, we compute the phase within a predefined local radius and quantize the phase by observing the sign of both the real and the imaginary part of the local phase. Similarly to uLBP, the quantized neighbourhood of each pixel is reported as an 8-bit binary string.

Given an image, the LPQ value is computed for every pixel. Next, local histograms with 256 bins are computed within a sliding window. We move this window, with a certain overlap between two neighbouring windows, in the horizontal and vertical directions over the image and concatenate the resulting local histograms. For a 20×20 window size and an overlap of 10 pixels, this results in a 16,384 dimensional feature vector.

Histogram of Oriented Gradients (HOG): Computation of the HOG [4] descriptor involves five steps: gradient computation, orientation binning, histogram computation, histogram normalization and concatenation of local histograms. The algorithm starts with computing the local gradient by convolving a 3×3 region (HOG cells) with two one-dimensional filters (-101) and $(-101)^T$. The local orientation associated with the center of each HOG cell is the weighted sum all filter responses within the cell. The local orientation is quantized into a bin value in the [0, 2π] interval. Subsequently, the image is divided into blocks of equal size and a local histogram of quantized orientations is computed for each block. This histogram is normalized with the L2-norm. Finally, all local histograms are concatenated to form the HOG descriptor for the image. The HOG descriptor for a block size of 8×8 pixels and 9 orientation bins has 5184 dimensions.

Subspace projection: Once the texture descriptors are computed, they are projected onto a lower subspace using a projection matrix (one for each descriptor). The projection matrix is computed using PCA on the training set. The optimal number of dimensions for the target feature subspace is estimated using Maximum Likelihood Estimation (MLE) [15]. Depending on the choice of training data and the texture descriptor used, the resulting feature space has at least 18 and in some cases up to 150 dimensions.

3.2. Cluster Generation

Once the feature subspace corresponding to a texture descriptor is derived, the next step is to cluster the training data in this subspace (see step ① in Fig. 2). The K-means algorithm [1] is used to accomplish this. The input to the K-means algorithm is the projected feature vectors from the training data. The output consists of K cluster centroids, $\{C_1 \dots C_K\}$.

4. Testing Phase

4.1. Gallery Classification

In step ②, we divide the test set into two distinct parts, the gallery and the probe set. The subjects in the test set are different from the ones in the training set. The gallery set contains exactly one image for each identity. The probe set may contain any number of images for each identity. The images in the gallery and probe sets do not overlap. We use the feature extraction and projection matrix that were computed in the training stage to project the gallery images into the feature space. Let I_g be a gallery image and F_g be the projected feature vector (corresponding to one of the texture descriptors). Then, the distances between F_g

[1] We also evaluated Hierarchical Clustering and Gaussian Mixture Models, but neither of them returned satisfactory results. Hierarchical Clustering does not converge well and produces inconsistent solutions, whereas GMM returns one large cluster that covers nearly all of the identities and $K-1$ small clusters that contain outliers.

and the cluster centroids is computed as $d_i = ||F_g - C_i||$, $i = 1 \ldots K$. These distances are sorted in ascending order and the gallery image (identity) is labelled with the cluster indices corresponding to the $\lambda < K$ smallest distances.

4.2. Probe Identity Retrieval

In the retrieval step (3), the given probe image, I_p, is projected into the feature subspace (corresponding to a texture descriptor), resulting in a feature vector F_p. The distance between F_p and the K centroids is next computed, and the probe is assigned the cluster indices corresponding to the λ smallest distances. Thus the search is confined to the gallery images (i.e., identities) in λ target clusters. Note that $\lambda = 1$ denotes a single-cluster search, while $\lambda > 1$ denotes a multi-cluster search. The output of the retrieval process is the list of gallery identities, $L_{gallery}$, corresponding to the λ target clusters.

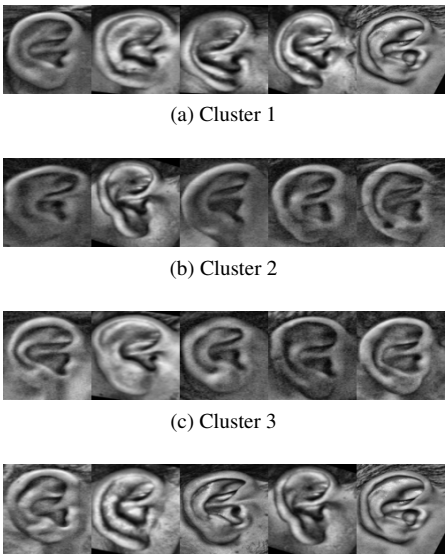
It is also possible to generate two different feature subspaces (e.g., corresponding to two different texture descriptors) and generate clusters independently in these individual subspaces. Let there be two sets of clusters corresponding to two subspaces S^1 and S^2 with centroids $\{C_1^1 ... C_n^1\}$ and $\{C_1^2 ... C_m^2\}$. The classification process will now result in two sets of cluster indices, one corresponding to S^1 and the other corresponding to S^2. Thus the output of the retrieval process will be two sets of gallery identities, $L^1_{gallery}$ and $L^2_{gallery}$. Subsequently, we can combine the two lists of identities using simple set operations such as union and intersection. The final list of gallery identities to be matched will be $(L^1_{gallery} \cup L^2_{gallery})$ or $(L^1_{gallery} \cap L^2_{gallery})$, respectively.

5. Experimental Analysis

The classification performance is defined in terms of a trade off between the pre-selection error and the penetration rate as defined in [12]. The pre-selection error rate (PSE) computes the probability that an image I_p from the probe set is not assigned to the same cluster as the corresponding identity I_g in the gallery set [2]. The penetration rate (PEN) is defined as the average fraction of the gallery database that has to be searched based on the list of retrieved gallery identities. The ultimate goal in classification is to reduce both the penetration rate as well as the pre-selection error rate. In an ideal clustering and classification scheme, the pre-selection error rate would be zero and the penetration rate would be $1/n$ where n is the number of images in the gallery set.

Let K be the total number of clusters in a feature subspace. Further, let n be the number of images in the gallery set and m the total number of images in the probe set. In our experiments, each identity in the gallery has exactly one image. Let $\xi_{p_i} \subset \{C_1 ... C_K\}$ be the cluster labels of I_{p_i}, the i-th probe, and L_{p_i} be the corresponding list of gallery images (identities) retrieved from the database. Moreover let $\xi_{g_i} \subset \{C_1 ... C_K\}$ be the cluster labels of the corresponding gallery image I_{g_i} with the same identity as I_{p_i}. Note that $|\xi_{p_i}| = |\xi_{g_i}| = \lambda$.

[2]The pre-selection error rate is (1 - hit rate)



(a) Cluster 1

(b) Cluster 2

(c) Cluster 3

(d) Cluster 4

Figure 4: Example images in each cluster for LPQ-3-20-15 with K=4. The images show the closest ears to each cluster centroid in ascending order (from left to right)

$$Hit(p_i) = \begin{cases} 1, & \text{if} \quad \xi_{p_i} \subseteq \xi_{g_i} \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

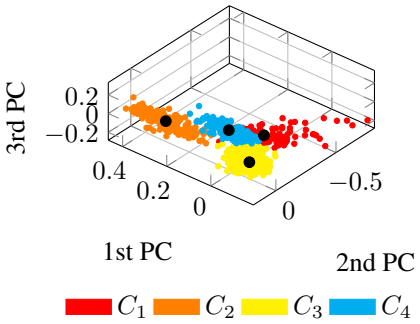
$$PSE = 1 - \frac{1}{m} \sum_{i=1}^{m} Hit(p_i) \tag{2}$$

Accordingly, the penetration rate can be written as

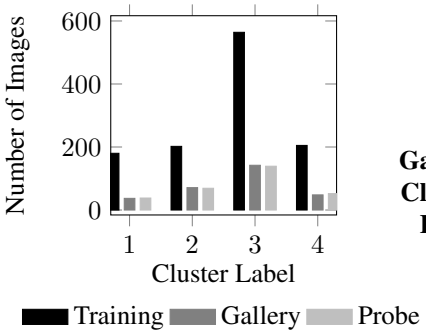
$$PEN = \frac{1}{m} \sum_{i=i}^{m} \frac{|L_{p_i}|}{n}. \tag{3}$$

6. Evaluation and Results

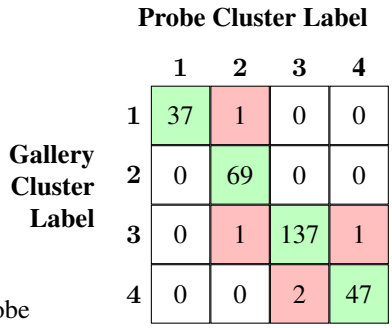
All results in this section are based on a heterogeneous dataset that has been composed of images from the UND-J2 (1800 images from 415 subjects) [23], AMI (700 images from 100 subjects) [8] and IITK (494 images from 125 subjects) [14] databases. The dataset used in our classification experiments consists of 2432 images from 555 subjects: 363 subjects from UND-J2, 67 subjects from AMI and 125 subjects from IITK. There are at least two samples per subject. (Images of 52 subjects from UND J2 and 33 subjects from AMI were used to train the CPR model for



(a) First three principal components (PC) of the feature space showing 4 different clusters (colors) and their respective centroids (black)

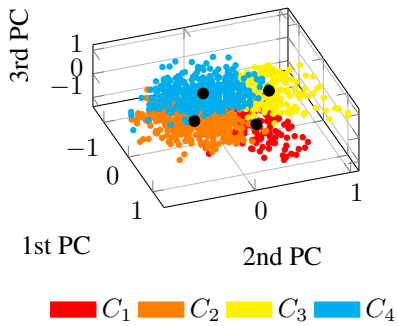


(b) Number of images per cluster in the training set, gallery set and probe set

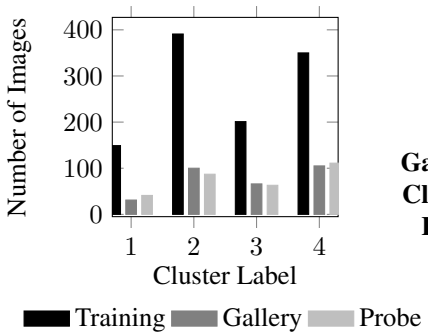


(c) Confusion matrix

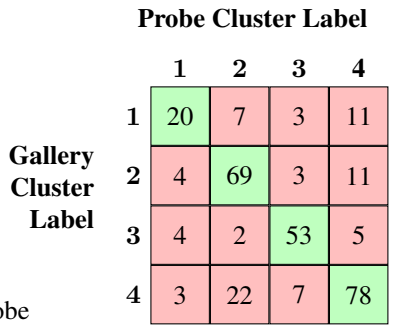
Figure 5: Cluster analysis for LPQ-3-20-15 with $K = 4$



(a) First three principal components (PC) of the feature space showing 4 different clusters (colors) and their respective centroids (black)



(b) Number of images per cluster in the training set, gallery set and probe set



(c) Confusion matrix

Figure 6: Cluster analysis for HOG-8-9 with $K = 4$

ear normalization, and were not used in subsequent experiments).

Training is accomplished by randomly selecting 253 subjects; all images of the 253 subjects are used in the training phase. On average, the training set contains approximately 1100 images. The remaining 302 subjects are used for testing. For each test subject, 1 image is added to the gallery database while the remaining images are used as probes. All experiments were conducted with 10-fold cross-validation.

In order to generate the clusters for a specific feature subspace, the K-means algorithm is used. The input consists of the projected feature vectors for a set of training images and the output is a set of cluster centroids. Since the output relies on the initialization process, the K-means technique is run 1000 times with a different set of initial centroids each time. From these 1000 solutions, we pick the one with the smallest sum of distances between all feature vectors and their respective cluster centroids. An analysis of the best solution using the silhouette measure [21] indicated that small values of K result in more coherent clusters than large values of K.

Given that we have a solution with a fixed number of centroids, we evaluated the performance of the proposed classification and retrieval scheme in three different steps. In the first, step we focus on the pre-selection error when the search process is confined to a single cluster for each probe. In the second experiment, we allow the search to expand to multiple clusters corresponding to the nearest centroids. Finally we evaluate the classification performance when candidate lists corresponding to multiple feature subspaces are combined [3].

6.1. Single Cluster Search

In the first experiment, we compare the classification performance due to feature subspaces generated from

[3] We found that fusing the identity lists of more than two cluster spaces does not improve the performance.

Table 1: $PSE//PEN$ (in %) for different configurations of the three texture descriptors for a single-cluster search. The performance is reported for different values of K. The best tradeoff between PSE and PEN for each configuration is denoted in bold. The best performance was achieved with LPQ-3-20-15.

	Number of Clusters (K)						
Algorithm	K=2	K=3	K=4	K=5	K=6	K=10	K=20
LPQ-3-20-0	1.37 // 65.46	7.6 // 35.37	**7.06 // 30.72**	9.13 // 29.30	14.42 // 25.66	30.00 // 14.25	37.83 // 11.38
LPQ-3-30-10	1.97 // 63.94	4.71 // 37.90	**5.10 // 32.49**	12.57 // 25.94	18.82 // 20.37	31.50 // 14.03	44.77 // 8.11
LPQ-3-20-15	0.77 // 65.81	4.75 // 38.65	**3.11 // 32.08**	3.77 // 32.64	8.68 // 27.34	30.43 // 13.4	32.52 // 12.84
LPQ-3-12-7	1.07 // 65.52	4.28 // 42.76	**5.09 // 32.94**	6.53 // 31.13	10.99 // 27.85	27.01 // 16.22	33.59 // 12.86
LPQ-5-20-15	0.61 // 57.25	7.66 // 35.34	**5.9 // 31.70**	10.23 // 27.85	17.53 // 21.69	27.21 // 14.31	36.01 // 10.02
LPQ-10-20-10	**4.16 // 53.06**	9.66 // 35.10	12.46 // 28.06	18.85 // 20.47	23.85 // 17.60	33.02 // 12.07	44.78 // 7.06
HOG-8-9	**6.46 // 49.88**	16.87 // 37.81	19.75 // 29.55	25.33 // 21.41	28.52 // 18.73	33.63 // 11.35	48.8 // 6.23
HOG-16-32	**11.14 // 50.08**	24.64 // 36.05	26.61 // 29.09	31.58 // 21.44	34.89 // 18.47	43.27 // 11.96	42.27 // 6.45
uLBP-1-20-0	5.60 // 50.69	**7.96 // 35.88**	9.12 // 30.24	19.25 // 21.70	21.44 // 19.13	33.17 // 12.30	46.44 // 6.67
uLBP-1-20-10	5.07 // 50.09	**6.64 // 35.54**	9.24 // 29.54	16.52 // 21.81	20.56 // 17.98	31.94 // 12.20	43.81 // 6.92
uLBP-1-20-15	4.61 // 50.85	**5.15 // 36.00**	5.13 // 20.83	16.43 // 18.83	17.75 // 31.32	31.17 // 12.63	42.85 // 7.71
uLBP-2-20-10	**5.64 // 50.04**	8.61 // 36.18	10.17 // 20.33	21.6 // 17.16	25.03 // 28.92	33.90 // 11.93	48.43 // 6.76

uLBP, LPQ and HOG. Each texture descriptor was tested with different parameter sets and with different window sizes. However, we found that many of the configurations result in similar performance. In Table 1, we report the pre-selection error (denoted as PSE) and the penetration rate (denoted as PEN) of specific configurations. The configurations for LPQ and uLBP-techniques are defined as follows: `<algorithm> - <radius> - <windowSize> - <overlap>`. The configuration for HOG is defined as `<algorithm> - <block size> - <number of bins>`.

For all texture descriptors in Table 1, we see that the PSE declines monotonically with an increasing number of clusters. As expected, the penetration rate decreases with an increasing number of clusters. This implies that there is no optimal number of clusters that can be automatically determined for each of the feature subspaces.

When comparing the performance of LBP, LPQ and HOG feature spaces, we observe that LPQ with a small radius and large overlap between the local windows has the best classification performance (also see Fig. 4). In our experiments, HOG yields the largest pre-selection error rates. The performance of uLBP lies between HOG and LPQ when the number of clusters is smaller than 6. For $K \geq 10$ the classification performance of HOG and uLBP becomes similar. When performing single cluster search with LPQ, solutions with K=4 appear to be a good choice, whereas for uLBP, $K \leq 3$ appears to be good. The HOG descriptor does not seem to lend itself to clustering since the pre-selection error rate is larger than 5% for K=2.

The penetration rate for HOG and uLBP is roughly $1/K$, whereas the penetration rate for LPQ tends to be larger than $1/K$. We can conclude from this, that the points in all examined feature subspaces are not uniformly distributed and that the number of identities per cluster is different. This is further illustrated in Fig. 5, where an example solution for LPQ-3-20-15 is shown. Fig. 6 shows an example solution in the HOG feature space, where the preselection error is significantly larger than for LPQ. As shown in Fig. 5c, clusters 3 and 4 mainly contribute to the overall preselection error, because these two clusters are located next to each other in Fig. 5a. [4] As shown in Fig. 5b the number of identities per cluster varies across the clusters. These variations can partly be explained by the fact that the input images come from three different datasets that contain a different number of subjects. C_2 mainly contains images from IITK, whereas C_1 contains many images from AMI. Fig. 4 shows examples of the five closest ear images to each centroid for LPQ-3-20-15. Images that originate from a particular database are overrepresented in some of the clusters; however, each cluster contains images from all of the three original databases. This implies that the classification not only reflects the identity of a person, but also contains information about skin tone (IITK contains ear images from Asian Indians, while AMI and UND-J2 mainly contain images from Caucasians). This is confirmed by evaluations using only a single database, where the capture settings and the skin tone of most subjects are the same. The performance of these individual databases is lower than that of the combined dataset. On UND-J2, for instance, we obtain a penetration rate of 81.45% for a pre-selection error rate of 1.11%.

Cross-database evaluations show that texture descriptors contain information that captures the demographic at-

[4] The reader has to exercise caution when interpreting these figures. These are projected features - the original dimensionality is 73

Table 2: PSE and PEN (in %) for different configurations of the three texture descriptors when multi-cluster search is used. Here K = 10. The best tradeoff between PSE and PEN for each configuration is denoted in bold. The best results were achieved with LPQ-3-20-15.

	Number of clusters searched (λ) for K = 10.						
Algorithm	**λ=2**	**λ=3**	**λ=4**	**λ=5**	**λ=6**	**λ=7**	**λ=8**
LPQ-3-20-0	10.58 // 28.01	2.77 // 40.99	**1.30 // 53.48**	0.27 // 65.93	0.10 // 74.98	0.00 // 84.84	0.00 // 92.76
LPQ-3-20-10	8.77 // 27.90	2.00 // 40.80	**0.83 // 53.56**	0.27 // 66.55	0.03 // 75.24	0.00 // 81.10	0.00 // 89.94
LPQ-3-20-15	6.20 // 32.74	**0.99 // 47.10**	0.40 // 61.69	0.00 // 73.58	0.00 // 81.68	0.00 // 90.59	0.00 // 97.30
LPQ-3-12-7	8.07 // 31.41	**1.70 // 46.08**	0.601 // 59.99	0.30 // 72.49	0.20 // 81.06	0.03 // 89.73	0.03 // 96.77
LPQ-5-20-15	8.30 // 28.12	1.47 // 41.65	**0.53 // 53.17**	0.23 // 64.48	0.03 // 75.64	0.00 // 87.01	0.00 // 96.23
LPQ-10-20-10	14.45 // 23.08	6.89 // 34.23	3.41 // 45.67	1.97 // 56.52	**0.57 // 67.31**	0.20 // 78.28	0.03 // 88.77
HOG-8-9	17.43 // 22.34	8.83 // 33.58	4.27 // 44.31	2.17 // 54.71	**1.30 // 65.11**	0.77 // 75.13	0.40 // 85.11
uLBP-1-20-10	11.32 // 23.62	4.30 // 34.66	2.40 // 44.42	**1.73 // 53.93**	0.97 // 63.21	0.37 // 72.48	0.20 // 82.77

tributes of the subjects and the acquisition settings. The cluster centroids obtained from one database do not properly reflect those from the other databases. We plan on incorporating additional features related to the shape of the ear to mitigate this concern.

6.2. Multi Cluster Search

In the second experiment, we explore the impact of multi cluster search. Based on the probe feature vector, the clusters corresponding to λ nearest centroids are searched. This potentially decreases the pre-selection error, but will also increase the penetration rate. The results for this experiment are summarized in Table 2. Here, the best configurations from Table 1 corresponding to K = 10 were used. We found that, for solutions with a larger K, the number of clusters does not influence the rate in which the penetration rate converges. This means that a graph depicting the ratio between pre-selection error rate and penetration rate will have the same shape, regardless of K (see Fig. 7). For higher values of K, we have more possibilities to select λ in a way that meets the requirements of a particular application.

As the results show, multi cluster search quickly reduces the pre-selection error at the cost of increased penetration rate. Due to the fact that the number of identities in each cluster varies, the penetration rate increases much faster than $1/\lambda$ with $\lambda < K$. However, searching through the closest two clusters significantly improves the performance by keeping the penetration rate below 50% while reaching a pre-selection error that is as small as 0.1% for LPQ-3-20-15. For other LPQ-based configurations with a radius of 3, the pre-selection error falls below 1% when four clusters are included in the search process at the cost of a higher penetration rate between 53.5 and 60%. All the other configurations result in searching at least 75% of the gallery images in order to obtain a pre-selection error below 1%.

Upon evaluating the identification performance, we obtain a rank-1 recognition rate of 93.06% when searching through three neighboring clusters using LPQ-3-20-15. As opposed to an exhaustive search, where we would obtain a similar performance, we only have to compare against 47.10% of the images, on an average, in the database.

6.3. Feature Space Fusion

As pointed out in Section 4.2, the identity lists corresponding to multiple feature subspaces can be combined to facilitate the retrieval process. Fusion is carried out by either using the union or the intersection of these two lists. Additionally, the previously mentioned multi-cluster search can be used in each of these feature spaces. Our results are based on different numbers of clusters searched (λ) and K = 10.

As expected, the penetration rate when using the union operator on the identity lists is much higher than when using the intersection operator. The intersection operator results in a pre-selection error rate of 1.98% at a penetration rate of 55.53% when a single cluster search strategy is used in each subspace. Searching through 5 clusters only slightly improves the performance and yields a pre-selection error rate of 1.65% and a penetration rate of 66.69%. Using the union operator results in a pre-selection error rate of 0.99% at a penetration rate of 47.65% when searching though 3 clusters in each feature space. This implies that the classification performance was not necessarily improved when fusing two identity lists.

7. Summary and Future Work

Using a single cluster search strategy the best results were obtained using LPQ with a radius of 3 and a 20×20 window size with 15 pixels overlap (pre-selection error rate was 3.11% with a penetration rate of 31.7%). A multi cluster search strategy further reduces the pre-selection error to

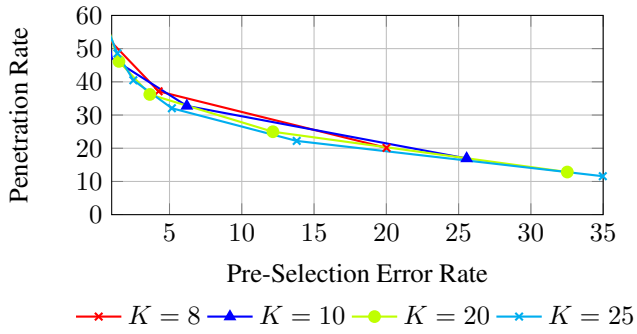


Figure 7: Impact of K on the convergence rate of the pre-selection error rate and penetration rate trade-off with LPQ-3-20-15. The number of clusters to be searched increases from right to left.

0.99% with a penetration rate of 47.1%. In summary, we have the following observations.

- Unsupervised classification of 2D ear images using texture descriptors is possible.
- Solutions with four clusters are a good choice for single cluster search when using the LPQ texture descriptor.
- A multi cluster search strategy further improves the classification performance.
- Fusion of candidate lists corresponding to two different feature subspaces using the union or intersection operator does not improve the classification performance.

This work can be extended in many ways: (a) We will investigate if all three texture descriptors cluster the subjects similarly. (b) In this work, only left ear images were used. We plan to investigate if the left and right ears of subjects are clustered similarly. (c) We will study the classification error when the quality of the input image is degraded. (d) We plan on incorporating shape features to improve classification accuracy, especially in homogeneous datasets.

Acknowledgment

This applied research effort was supported by the Federal Bureau of Investigation Biometric Center of Excellence. Thanks also to the German BMBF for the support of the GES-3D project.

References

[1] A. Abaza, A. Ross, C. Hebert, M. A. F. Harrison, and M. S. Nixon. A survey on ear biometrics. *ACM Comput. Surv.*, 45(2):22:1–22:35, March 2013.

[2] T. Ahonen, E. Rahtu, V. Ojansivu, and J. Heikkila. Recognition of blurred faces using local phase quantization. In *International Conference on Pattern Recognition*, 2008.

[3] N. Boodoo-Jahangeer and S. Baichoo. LBP-based ear recognition. In *Bioinformatics and Bioengineering*, 2013.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, volume 1, pages 886–893 vol. 1, June 2005.

[5] N. Damer and B. Fuhrer. Ear recognition using multi-scale histogram of oriented gradients. In *Intelligent Information Hiding and Multimedia Signal Processing*, 2012.

[6] P. Dollar, P. Welinder, and P. Perona. Cascaded pose regression. In *Computer Vision and Pattern Recognition*, 2010.

[7] R. Gadde. Iris indexing and ear classification. Master's thesis, West Virginia University, 2012.

[8] E. Gonzalez, L. Alvarez, and L. Mazorra. Normalization and feature extraction on ear images. In *International Carnahan Conference on Security Technology*, 2012.

[9] Y. Guo and Z. Xu. Ear recognition using a new local matching approach. In *International Conference on Image Processing*, 2008.

[10] F. Hao, J. Daugman, and P. Zielinski. A fast search algorithm for a large fuzzy database. *IEEE Trans. Information Forensics and Security,*, 3:203–212, 2008.

[11] A. V. Iannarelli. *Ear identification*. Paramont Publishing Company, 1989.

[12] ISO/IEC TC JTC1 SC37 Biometrics. *ISO/IEC 19795-1:2006. Information Technology – Biometric Performance Testing and Reporting – Part 1: Principles and Framework*. International Organization for Standardization and International Electrotechnical Committee, Mar. 2006.

[13] R. Khorsandi and M. Abdel-Mottaleb. Gender classification using 2-D ear images and sparse representation. In *Workshop on Applications of Computer Vision,*, 2013.

[14] A. Kumar and C. Wu. Automated human identification using ear imaging. *International Conference on Pattern Recognition*, 45(3):956–968, March 2012.

[15] E. Levina and P. J. Bickel. Maximum likelihood estimation of intrinsic dimension. In *Neural Information Processing Systems*, 2004.

[16] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition*. Springer-Verlag, 1st edition, 2009.

[17] R. Mukherjee and A. Ross. Indexing iris images. In *International Conference on Pattern Recognition*, 2008.

[18] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:971–987, 2002.

[19] F. Perronnin and J.-L. Dugelay. Clustering face images with application to image retrieval in large databases. In *Proc. SPIE Conf. Biometric Technology for Human Identification II*, volume 5779, pages 256–264, 2005.

[20] A. Ross and R. Mukherjee. Augmenting ridge curves with minutiae triplets for fingerprint indexing. In *SPIE Biometrics*, 2007.

[21] P. J. Rouseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:5365, 1987.

[22] T. Vikram, K. Chidananda Gowda, D. Guru, and S. Urs. Face indexing and retrieval by spatial similarity. In *Image and Signal Processing*, 2008.

[23] P. Yan and K. Bowyer. Biometric recognition using 3D ear shape. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 29:1297 – 1308, August 2007.

[24] K. Zuiderveld. *Graphics Gems IV*, chapter Contrast Limited Adaptive Histogram Equalization, pages 474–485. Academic Press, 1994.