# Assignment #1: Basic Recognition

Mila Marinković
*IBB 22/23, FRI, UL*
*mm9136@student.uni-lj.si*

## I. Introduction

Within the biometric community, there is an active study into the identification of people through ear images. Similar to other biometrics like the face, iris, and fingerprints, the ear includes a variety of distinctive characteristics that make it possible to identify a person.

In this project, our task was to use the well-known algorithm Local Binary Patterns (LBP) for ear recognition and evaluate its performance. We compared this with both the OpenCV implementation and a basic pixel-by-pixel comparison of images in order to see how the LBP's implementation performed. In other words, our task was to compute how well our algorithm will classify data.

The remaining part of the work will be focusing on the following: Section 2 represents some related prior work. Section 3 focuses on the methodology, while Section 5 describes the experiments which were previously carried out, followed by Section 4 where obtained results and a discussion of results are presented. Finally, section 6 contains the conclusions.

## II. Related Work

Local Binary Patterns (LBP) are a texture descriptor made popular by the work of Ojala et al [1]. They paper demonstrates a multiresolution method for classifying grayscale and rotation-invariant textures based on neighborhood binary patterns and the nonparametric distinction between sample and prototype distributions. The approach is founded on the understanding that some local binary patterns, referred to as "uniform," are essential components of local image texture and that the histogram of their occurrence is a pretty effective texture feature.

Some other useful papers are RLBP: Robust Local Binary Pattern [2], A Completed Modeling of Local Binary Pattern Operator for Texture Classification [3] etc.

## III. Methodology

In this section, we explain the used methods for conducting this experiment. We did that in 3 stages:

1) Resized and converted input images
2) Call feature extractor for each:
    a) Pixel-by-pixel transforms the 2D image into a 1D vector
    b) LBP was done with different adjustments: different radii, with and without histograms. It selects the neighborhood (of the selected size) for each pixel in the image. Then it compares every neighbor pixel with the central pixel. If the neighbor of the central pixel is smaller than the central pixel, it set its value to 0, otherwise to 1. We then turn any ordering of these numbers into one string of numbers, which we turn into a decimal number. The central pixel value is replaced by that calculated decimal value. If we do not use the histogram, then all images are converted into 1D vectors. Otherwise, for each neighborhood, we check how many times some of the possible 256 numbers occurred.
    c) OpenCV
3) Calculate the rank-1 recognition rate by comparing all the calculated feature vectors for each type of feature extractor: we compare the distances between computed vectors and for each of them, we find the one which is the least distant and check if they belong to the same class. Predictions are added together across all vectors, and the result is the percentage of rank-1 recognition rate.

## IV. Experiments

The experiment was carried out on a given dataset that contains data from 100 subjects, and 10 cropped ear images per subject (1000 images in total). Pictures were taken in different positions, different angles, and different light settings.

Initially, each image was resized and converted to greyscale, then the feature extractor was called. For each feature extractor, the experiment was conducted for different image sizes (128x128 and 64x64). LBP was implemented on different image sizes using different radii (1 and 2) and the rank-1 recognition rate was calculated with and without histograms. In order to check our LBP implementation results, the Rank-1 recognition rate was calculated with the implementation from OpenCV as well, using different methods such as default, uniform, and ror.

Furthermore, the distances between 2 feature vectors for each of the above-mentioned feature extractors were calculated with different distance measures (Euclid and Cosine).

## V. Results and Discussion

After performing the experiment, we obtained the following results.

### *A. Results*

Table I represent rank-1 recognition rate calculated with Pixel-by-pixel feature extractor on 2 different sizes with 2 different distance measures.

| PBP | Euclidean distance | Cosine distance |
|---|---|---|
| 128x128 | 0.132 | 0.167 |
| 64x64 | 0.132 | 0.168 |

TABLE I: Rank-1 recognition rate: Pixel by Pixel

In Table II and Table III is shown the rank-1 recognition rate computed with LBP implementation and OpenCV's implementation.

| LBP | | | Range 1 (3x3) | Range 2 (5x5) |
|---|---|---|---|---|
| size 128x128 | without histogram | Euclidean | 0.208 | 0.027 |
| | | Cosine | 0.229 | 0.023 |
| | with histogram | Euclidean | 0.248 | 0.035 |
| | | Cosine | 0.228 | 0.024 |
| size 64x64 | without histogram | Euclidean | 0.277 | 0.048 |
| | | Cosine | 0.283 | 0.4 |
| | with histogram | Euclidean | 0.306 | 0.073 |
| | | Cosine | 0.262 | 0.041 |

TABLE II: Rank-1 recognition rate: LBP

| openCV | | | Range 1 (3x3) defaut | Range 2 (5x5) default | Range 1 (3x3) uniform | Range 2 (5x5) uniform |
|---|---|---|---|---|---|---|
| size 128x128 | without histogram | Euclidean | 0.223 | 0.277 | 0.014 | 0.014 |
| | | Cosine | 0.144 | 0.187 | 0.012 | 0.013 |
| | with histogram | Euclidean | 0.261 | 0.277 | 0.016 | 0.016 |
| | | Cosine | 0.182 | 0.16 | 0.048 | 0.051 |
| size 64x64 | without histogram | Euclidean | 0.243 | 0.31 | 0.033 | 0.033 |
| | | Cosine | 0.166 | 0.232 | 0.046 | 0.046 |
| | with histogram | Euclidean | 0.293 | 0.31 | 0.039 | 0.039 |
| | | Cosine | 0.215 | 0.203 | 0.116 | 0.116 |

TABLE III: Rank-1 recognition rate: OpenCV

### *B. Discussion*

From Table II, which represents a rank-1 recognition rate computed with LBP, we can say that the smaller the image size is, the results are better. However, the image size should not be too small, because in that way we can lose some image information. Furthermore, LBP using histogram and using euclidean distance measures for feature vector comparison seems to give the best results.

All in all, LBP results are very weak but compared to the results obtained with OpenCV, we can say that they are pretty much the same and that they are better than the results obtained with Pixel-by-pixel.

## VI. Conclusion

The use of ears as a biometric method of identification has many benefits. It is a very universal modality that may be acquired with ease from pictures of humans. However, cropped photos of ears may not be high-quality and maybe that is the case why we got this small rank-1 recognition rate. We can conclude that using LBP and openCV did not prove to be the best method and for future work, we can suggest doing an ear recognition tasks with a deep convolution neural network (CNN) to extract features from images.

## References


[1] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[2] J. Chen, V. Kellokumpu, G. Zhao, and M. Pietikäinen, "Rlbp: Robust local binary pattern." 2013.

[3] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 19, pp. 1657–63, 03 2010.