Assignment #1: Basic Recognition

Luka Boljević
IBB 22/23, FRI, UL
lb7093@student.uni-lj.si

I. Introduction

This assignment is about implementing Local Binary Patterns (LBP) for image recognition/classification, and comparing it to the feature extractor which just returns the pixels of the image as the feature vector. This most basic feature extractor serves as a baseline for testing LBP.

II. Related Work

The original paper describing LBP, proposed by Ojala et al. [1], is used as reference for implementing (uniform) LBP.

III. Methodology

To determine how good a feature extractor is, we will calculate the rank-1 recognition rate, or rank-1 accuracy. This means that, after converting all images from the data set into feature vectors, we will find the closest vector to a currently observed one, using some distance metric, and see if they belong to the same class. If they do, we count this as a correct classification, and incorrect otherwise. The ratio of correct classifications yields the rank-1 accuracy. We will compare the rank-1 accuracy of the basic "pixels feature extractor" with various "versions" of LBP and uniform LBP (uLBP for short).

IV. Experiments

The data set consists of 1000 samples of human ears, split into 100 classes (so, 10 samples per class). The experiment is performed in the following way. After converting all images to gray scale, and resizing to a fixed size, we run the explained method separately for the plain pixel feature extractor, LBP with various parameters, and uLBP with various parameters.

There are a lot of parameters to vary for (u)LBP: (1) radius (R), (2) pattern length (P), (3) using histograms or not, (4) using cyclic shifting or not (used to achieve rotation invariance; only applies to default LBP, uLBP is by default rotation invariant), (5) using different levels of local region overlaps. When calculating distances between feature vectors, different distance metrics could be used - we decided on **Euclidean**, **Cosine** and **Manhattan**. We could also vary the image size, but for this experiment we didn't. We will also not experiment with (5). We should also mention that, when $P \neq 8R$, i.e. when we do not take all neighbors of a pixel, we sample the neighbors randomly. The random sampling does not affect the rank-1 accuracy drastically (results for different runs differ by ± 0.01).

There are (at least) two ways we could apply histograms to the (u)LBP image (image obtained after converting it with (u)LBP). One is to just apply it to the entire image. The problem with this approach is that we lose all spatial information, which resulted in terrible performance. For pretty much every setting of different parameters, using histograms this way worsened the rank-1 accuracy, compared to the "non-histogram" approach for the same parameters. For this reason, and due to the 2 page limitation, we will not show those results here. The other way we could use histograms is by splitting the image into blocks of fixed size (say, **16x16**) and calculating the *local* histogram for each block. The final feature vector is obtained by concatenating all local histograms. The idea for this was found on the lecture slides, and we use this approach in the implementation.

V. Results and Discussion

We will now show the results of the experiment. As stated, we can vary different parameters for (u)LBP, but we fixed the image size to 128x128 and the amount of pixels we move after processing a pixel, namely 1. We also use 16x16 blocks when calculating local histograms. We will now try to show the most representative results.

A. Results

TABLE I
Rank-1 accuracies for the pixels feature extractor. There are no parameters to vary for this case, except the distance metric. We see that we get the highest rank-1 accuracy when using Cosine distance.

Euclidean	Cosine	Manhattan
0.132	0.167	0.132

The results in Table I are used as baseline. We will first show results for the usual settings for (u)LBP - $R = 1$ and $P = 8$.

The first thing we notice in Table II is how significantly lower the results are for uLBP, even when using histograms, which evidently have a big impact on the result. While local histograms improve performance, both for LBP and uLBP, cyclic shifting on the other hand, at least for this dataset, worsens the performance. Similar behaviour was present for the entire experiment, so going forward we will only show

TABLE II

RANK-1 ACCURACIES FOR LBP (FIRST 4 ROWS) AND uLBP (LAST 2 ROWS) FOR $R = 1$ AND $P = 8$. HG STANDS FOR "HISTOGRAMS", CS STANDS FOR "CYCLIC SHIFTING", SO FOR EXAMPLE "NO HG, CS" MEANS THAT FOR THIS RUN, HISTOGRAMS WERE NOT USED, WHILE CYCLIC SHIFTING WAS.

LBP / uLBP	Euclidean	Cosine	Manhattan
No HG, no CS	0.189	0.073	0.232
no HG, CS	0.018	0.012	0.018
HG, no CS	0.246	0.328	0.355
HG, CS	0.118	0.124	0.142
No HG	0.015	0.012	0.019
HG	0.110	0.124	0.134

results from runs that used histograms, but didn't use cyclic shifting.

The second thing we notice in Table II is that Manhattan distance yields a higher rank-1 accuracy than Euclidean and Cosine distance, in all of the runs, while Euclidean and Cosine are "fighting" who is better, both for LBP and uLBP. The same behaviour occurred for the entire experiment, so from now on we will only use results that Manhattan distance yields. Let us see next how our results compare against Scikit's implementation of LBP, for the case of $R = 1$, $P = 8$.

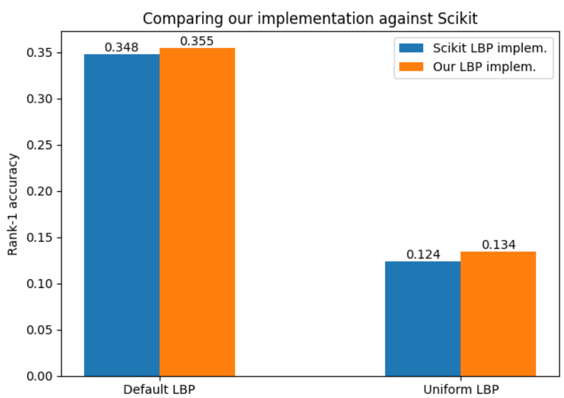

Fig. 1. A bar plot comparing the rank-1 accuracy obtained from our implementation and Scikit, for both LBP and uLBP, for $R = 1$, $P = 8$, using histograms, not using cyclic shifting, and using Manhattan distance.

Figure 1 shows that we were able to beat Scikit's implementation of (u)LBP ever so slightly. Victory. Let us now see how varying R and P will affect performance.

Table III shows that, for a fixed P, as we increase R, the accuracy goes up, for both LBP and uLBP. However, as we increase P, for a fixed R, the accuracy goes up for LBP, but drops for uLBP. It seems like these trends will continue as we increase R/P, however we have to remember that our images are size 128x128, and that we will at some point start getting worse results, as too high R/P would not capture the information around a pixel quite well.

B. Discussion

It is evident that we can beat the pixels extractor quite easily, which is to be expected, however we don't beat it in all cases. From Table II, we see that using cyclic shifting to achieve rotation invariance makes LBP perform worse than the basic pixels extractor. This could be because the images in our dataset are not rotated at all, so trying to achieve rotation invariance does not help. Similarly, uniform LBP does not give us good results. The explanation might be that, in the original LBP paper [1], uniform binary patterns were introduced because the authors noticed that they are a fundamental property of texture, while for our experiment we are not recognizing textures, but ears. On the other hand, using local histograms is a great idea, because they capture the spatial information around a pixel quite well, which is reflected in results in Tables II and III.

TABLE III

RANK-1 ACCURACIES FOR DIFFERENT R AND P, FOR RUNS USING LOCAL HISTOGRAMS, NOT USING CYCLIC SHIFTING, AND USING MANHATTAN DISTANCE. TESTS WITH $P \geq 16$ WERE INFEASIBLE FOR LBP AS RAM RUNS OUT FAST. THE RANK-1 ACCURACY FOR LBP FOR $R = 3$, $P = 12$ IS THE MAXIMUM ACHIEVED DURING THE EXPERIMENT. THE RANK-1 ACCURACY FOR uLBP FOR $R = 3$, $P = 4$ IS THE MAXIMUM ACHIEVED DURING THE EXPERIMENT.

LBP / uLBP	$P = 4$	$P = 8$	$P = 12$
$R = 1$	0.313	0.355	/
$R = 2$	0.329	0.392	0.426
$R = 3$	0.356	0.427	0.465
$R = 1$	0.233	0.134	/
$R = 2$	0.238	0.183	0.177
$R = 3$	0.268	0.267	0.223

It makes sense that increasing R and P for LBP, in Table III, increases the rank-1 accuracy, as we are simply capturing more information around a pixel. However, this, in combination with the previous explanation why uLBP does not work well for our dataset, could be the explanation why uLBP reaches its highest rank-1 accuracy for the "highest" R but "lowest" P.

VI. CONCLUSION

In this assignment, we implemented a basic pixels feature extractor, and (u)LBP, with the aim to evaluate their performance. We have varied a lot of parameters for (u)LBP, including using different distance metrics to calculate feature vector distances. We have evaluated that using Manhattan distance and local histograms, and not using cyclic shifting (for LBP) gives the highest overall results (for any R, P). We have established that, for LBP, higher values of both R and P give a higher rank-1 accuracy, while for uLBP, higher values of R and lower values of P yields better results. We have also discussed why (not) using certain parameters improves/deteriorates the performance of (u)LBP. In our future work, we tackle Assignment 2.

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.