# HOW TO EXPLOIT LARGE IMAGE DATA IN THE FIELDS OF TEXTURE CLASSIFICATION: A CASE STUDY WITH LOCAL BINARY PATTERNS

*Michael Gadermayr, Andreas Uhl*

Department of Computer Sciences, University of Salzburg, Austria

**ABSTRACT**

In the fields of texture classification, the sizes of images significantly vary according to the respective classification scenario. Whereas quite small image patches mostly lead to good classification accuracies, increasing the image size sometimes even has a negative effect. In this work, we focus on derivatives of Local Binary Patterns as these feature extraction methods offer a high discriminative power and efficiency on the one hand an can be effectively analyzed on the other hand. The aim is to get new insight and furthermore to explore strategies which can help to increase the classification performance. We investigate these strategies which exploit the obviously high distinctiveness of small image patches and simultaneously the redundancy available in large image patches. Finally it can be concluded that the traditionally applied strategies for texture classification should be reconsidered in case of sufficiently large image data.

## 1. INTRODUCTION

For several decades, texture classification has been a fundamental challenge in image processing. An early milestone work has been done by Haralick et al. [1], who have introduced features based on the gray level co-occurrence matrix. Furthermore texture classification based on Fourier [2] and Gabor filters [3], just to mention a few, has been done. Another work on exploiting local distributions with high impact has been proposed by Ojala et al. [4]. This method, called Local Binary Patterns (LBP), has been widely deployed, investigated and adapted to specific problem definitions [5, 6, 7, 8, 9].

Although many highly sophisticated approaches have been proposed during the last years [10, 11, 12, 13, 14], Local Binary Patterns and especially derivatives are still able to compete in many classification scenarios [15, 16]. In this work we focus on derivatives of this method because of their high discriminative power in combination with a high computational efficiency and thereby practical relevance. Furthermore, LBP turned out to be highly appropriate for analysis which is explained in Sect. 2.

During experimentation, we found out that LBP is able to generate quite good classification accuracies even in case of small image data. Whereas newer method (e.g. methods based on local pooling [10, 17]) require larger images, in case of LBP, image sizes of approximately $64 \times 64$ pixels (or even less) are sufficient to reach excellent performances. However, the available image material is often significantly larger. Unfortunately, simply increasing the image size mostly does not improve the classification accuracies significantly. In some cases even a loss of accuracy is observed.

Firstly, it is interesting to explore the reasons for this peculiar effect. From logical point of view, an increase of data would lead to a more stable probability distribution and in the worst case the achieved accuracy should remain stable. Furthermore, for practical reasons it is highly important to know how to deal with classification methods if there are larger images available. One of the approaches investigated in this paper has already been effectively applied to medical image data [15] where accuracy improvements are achieved by a split and fuse approach. However, the image data in that paper, which shows high variations within one class and even within one image, is quite different from the periodic and regular data, investigated in this work.

In this paper, effects on the classification performance are investigated by applying several different strategies in combination with varying training set sizes. This is done because the training set size turned out to be a crucial factor. In order to avoid any bias, a large image database is used which consists of high quality images of mainly periodic textures.

This paper is structured as follows. In Sect. 2 the different strategies to deal with large images are outlines. In Sect. 3 the experimental results are presented and discussed. Finally Sect. 4 concludes this paper.

## 2. METHODS

### 2.1. Feature Extraction: Local Binary Patterns

To compute the traditional LBP feature vector [18], the gray values of points surrounding a center pixel are extracted from the image. In a next step, the center pixel value is subtracted from the surrounding values and the sign function is applied to each difference value. Finally the obtained bits are concatenated and interpreted as a binary number (see Fig. 1). This first step is done for each point in the image. Then in a second

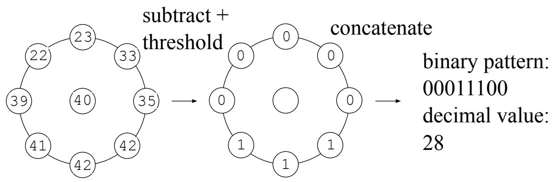



**Fig. 1**. Schematized flow of LBP computation.

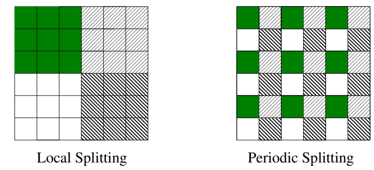



**Fig. 2**. Schema of the two splitting strategies (for $n = 4$). Similarly colored patterns are collected into one histogram.

step the histogram over all obtained binary numbers is computed. LBP derivatives are used in our experiments because of the perfect separation in step one and step two which allows two different splitting strategies as mentioned in the next subsection. We deploy two different versions of LBP. First we use a multi-resolution LBP version (MRLBP) [5], with an eight sample neighborhood and a radius (distance between center point and neighboring points) of one and two pixels. Furthermore, Local Ternary Patterns (LTP) [19] are utilized. Instead of binarization, this method generates a ternary code. An absolute difference smaller than a threshold $\Theta$ between the neighboring points leads to third (ternary) value, otherwise again the sign function is utilized to determine between zero and one. We utilize a setup with four neighbors (which turned out to be enough in case of LTP), a radius of two pixels and a threshold of five. The final feature vector thereby consists of 81 ($3^4$) bins.

### 2.2. Dealing with Large Images

In the following, we propose several strategies to deal with images which are significantly larger than a required size for LBP based feature extraction. Experiments showed that a size of $64 \times 64$ pixels is enough for quite accurate classification in case of our scenario. Even with smaller images good rates can be obtained.

First of all, we detect two different methods to reduce the number of samples which flow into one final histogram.

- **Local splitting**
  Local splitting in this context means that binary patterns in a certain local neighborhood are collected into one histogram which corresponds to the feature vector. This is done by first computing the binary patterns per pixel (step 1) and afterwords collecting the values in a rectangular neighborhood into a histogram (step 2). Separation is done by partitioning into $n \in \{1^2, 2^2, 3^2, ..., 10^2\}$ square sub-images (see Fig. 2, left). Thereby $n$ feature vectors can theoretically be generated from one single image.

- **Periodic splitting**
  Another strategy is to collect samples without considering the local neighborhood. Therefore, we apply a periodic partitioning of the binary patterns into histograms. Periodic in this case means that according to a specified step size $n \in \{1^2, 2^2, 3^2, ..., 10^2\}$, each $\sqrt{n}^{th}$ pattern in each direction is taken into one histogram (see Fig. 2, right). Thereby once again $n$ overlap-free histograms (which correspond to feature vectors) can theoretically be generated from one single image.

With local or periodic splitting we theoretically obtain a larger number of images and feature vectors. For an extensive analysis we investigate the achieved classification accuracies with the following strategies considering the training set and the evaluation set.

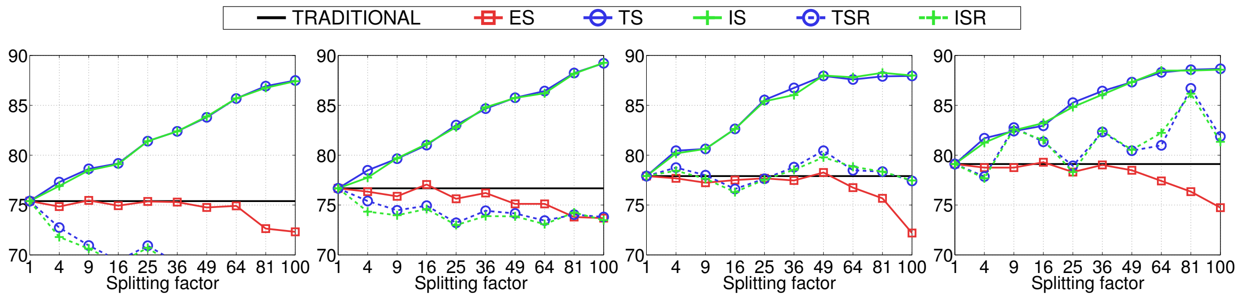
- **Image splitting (IS)**
  By local or periodic splitting of training and evaluation set images, the number of images is multiplied by the factor $n$. Thereby the training set is significantly enlarged which potentially has a positive effect. Furthermore, we obtain $n$ features and decisions for each image. To get one final decision, all of these decisions are fused by majority voting. This method has already been successfully applied in case of a medical decision support system [15].

- **Image splitting - reduced training set (ISR)**
  To separate the positive effect of the enlarged training set and the data size reduction, we furthermore investigate the accuracies achieved with a reduced training set. In this setting, the training set is artificially reduced to have the same size as in case of traditional classification without splitting.

- **Evaluation set splitting (ES)**
  Another strategy is to split only the evaluation set images. Thereby the training set size stays unchanged whereas the achieved decisions can be fused.

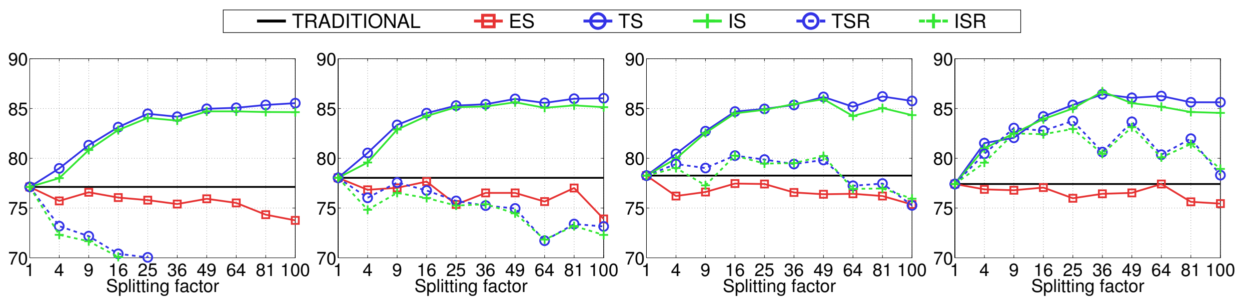
- **Training set splitting (TS)**
  If only the training set images are split (as in case of IS), we obtain an enlarged training set but only one final decision.

- **Training set splitting - reduced training set (TSR)**
  In case of this method, the training set of TS is artificially reduced as in case of the ISR method. Although this approach is supposed to be less effective than TS, it finally helps to separately consider the effect of the enlarged training set and data size reduction.




(a) Training set size 5 (b) Training set size 10 (c) Training set size 20 (d) Training set size 40

**Fig. 3**. Classification accuracies (vertical axis): LBP with local splitting.




(a) Training set size 5 (b) Training set size 10 (c) Training set size 20 (d) Training set size 40

**Fig. 4**. Classification accuracies (vertical axis): LTP with local splitting.

## 3. EXPERIMENTS

### 3.1. Setup

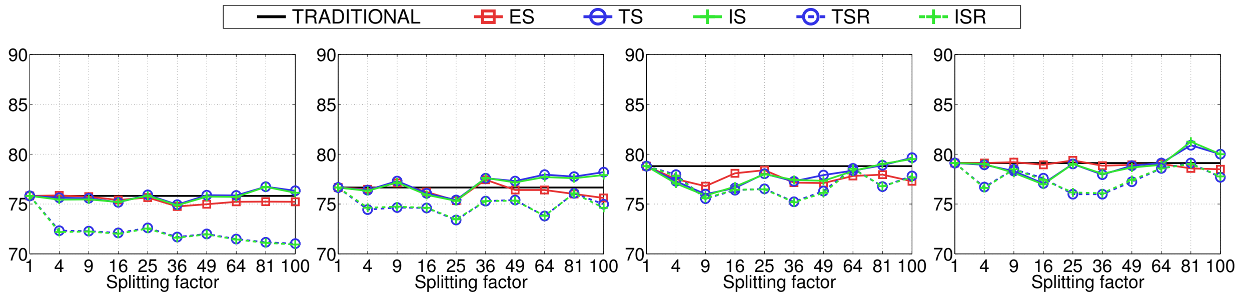
The experiments are based on the Kylberg texture database [20], consisting of 28 materials (each showing regular, periodic patterns) with 160 unique texture patches per class, captured at a single scale. Each image has a size of $576 \times 576$ pixel. The unique textures are divided in separate sets each containing 40 patches per class (i.e. the total number of images per set is 1120). We chose random subsets of one set for training and random subsets from another set for evaluation. This step is repeated 32 times to get stable average rates. We consider varying training set sizes $s \in \{5, 10, 20, 40\}$.

The feature vectors (LBP histograms) are finally $L^2$ normalized in order to be able to compare histograms of images of differing sizes. For final classification, we deploy the linear support vector classifier [21] which has been widely used in recent work on texture classification.

### 3.2. Results and Discussion

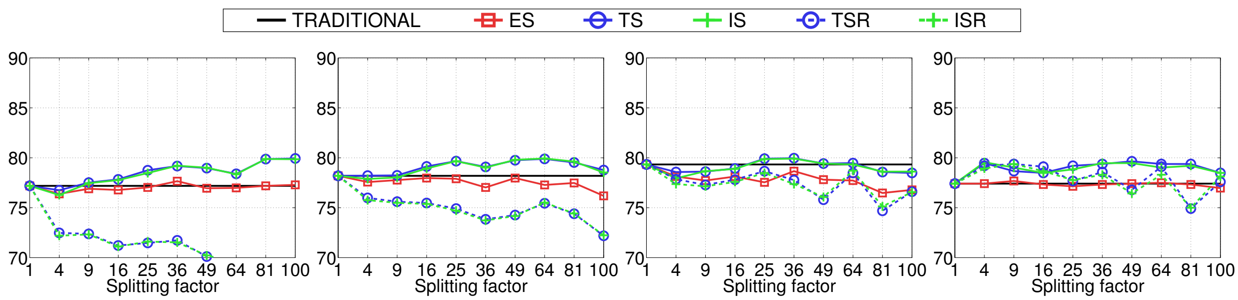
In Fig. 3 - 6, the resulting overall classification accuracies are shown for different strategies. First, we focus on local splitting (Fig. 3 and 4). One subfigure corresponds to one certain training set size reaching from 5 to 40. This value denotes the number of images per class in the training set. The values on the horizontal axis denote the splitting factor ($n$). This values reach from one (corresponds to traditional classification) to 100 which indicates a split into 10 (horizontal) times 10 (vertical) square sub-images. Considering traditional classification, we notice that the obtained accuracies (horizontal lines) can only be slightly improved by enlarging the training set (compare sub-figures (a) - (d)). Splitting the evaluation set images followed by decision level fusion (ES) obviously has a negative impact on the classification performance. Independently from the chosen setup, with increasing splitting factors, the performance decreases more or less monotonically.

Considering the methods IS and TS we see that the combination of a larger training set and the smaller images in each case leads to distinct and quite similar improvements. This is the case for IS which is based on splitting of all images (in combination with decision level fusion) and similarly for TS which is only based on split training set images. As the IS method is more complex than the TS method which is only based on split training set images, we will consequently focus on the TS technique. TS has two (potentially beneficial) effects. On the one hand the training set size is increased and on the other hand, the features are extracted from reduced image data. To find out which of these effects dominates, we furthermore look at the rates achieved with the reduced training sets (TSR). Thereby a fair comparison with traditional classification can be done considering the training set sizes. If considering a relatively small training set (see Fig. 3 (a)), we notice that a reduced training set in case of TSR and ISR has




(a) Training set size 5 (b) Training set size 10 (c) Training set size 20 (d) Training set size 40

**Fig. 5**. Classification accuracies (vertical axis): LBP with periodic splitting.




(a) Training set size 5 (b) Training set size 10 (c) Training set size 20 (d) Training set size 40

**Fig. 6**. Classification accuracies (vertical axis): LTP with periodic splitting.

a majorly negative impact. Even the rates of traditional classification cannot be reached. Interestingly, this behavior is reversed by adding images to the training set which is done in case of the sub-figures (b), (c) and (d). Having a training set that consists of a large number of items (e.g. 40 images per class as in sub-figure (d)), then the decreased image size leads to an accuracy improvement even if the training data size stays unchanged compared to traditional classification. This is highly astonishing as we supposed that an increased size of the training set images in worst case has no beneficial effect. Obviously this is still true but only when the training set is relatively small. However, as the TS method automatically leads to significantly increased training data, this approach even profits is case of relatively small training sets.

To find out why the decreased image size potentially leads to improved classification accuracies, we furthermore investigate the effects in case of periodic splitting as presented in Fig. 5 and 6. Here we see that the generally achieved improvements for both features are by far less distinct in case of large training data (see rates obtained with IS and TS). However, on opposite in combination with few training data (ISR and TSR in Fig. 3 (a) and 4 (a)), slightly better results are obtained.

We suppose that this behavior is due to the fact that the training data generated by local splitting is more idealistic. Due to the small neighborhoods that are considered, the feature vectors are extracted from images with a smaller degree of (intra-image) variation. These idealistic training data obviously requires more samples to get decision boundaries which are appropriate for an accurate generalization. Following the periodic splitting strategy, the histograms are collected from points all over the image and consequently are not collected from variation reduced data.

## 4. CONCLUSION

We have investigated diverse strategies to deal with large images in case of a varying amount of available training data. Some of these strategies have been invented only for analysis whereas others actually can be used to improve the classification accuracy. To put it into a nutshell, the reduction of the image size for feature extraction can have a significantly positive impact on the classification performance. This is especially true in case of having large training data. However, having relatively few images for training with a large size, splitting of these images should be considered in order to get a significantly enlarged (more idealistic) training set. Interestingly in case of small images the accuracy improvement with increasing training sets vanishes less rapid than in case of larger images. This is supposed to be due to lower degree of intra-image variations.

## 5. REFERENCES


[1] R. M. Haralick, Dinstein, and K. Shanmugam, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 3, pp. 610–621, November 1973.

[2] Ruzena Bajcsy, "Computer description of textured surfaces," in *Proceedings of the 3rd International Joint Conference on Artificial Intelligence (IJCAI'73)*, San Francisco, CA, USA, 1973, pp. 572–579, Morgan Kaufmann Publishers Inc.

[3] W.Y. Ma and B.S. Manjunath, "Texture features and learning similarity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'96)*, June 1996, pp. 425–430.

[4] T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," *Pattern Recognition*, vol. 32, no. 3, pp. 477–486, March 1999.

[5] T. Ojala, M. Pietikäinen, and T. Mäenpää, "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, July 2002.

[6] Xiangsheng Huang, Stan Li, and Yangsheng Wang, "Shape localization based on statistical method using extended local binary pattern," in *Proceedings of the 3rd International Conference on Image and Graphics (ICIG'04)*, Hong Kong, China, 2004, pp. 1–4.

[7] Shengcai Liao, Xiangxin Zhu, Zhen Lei, Lun Zhang, and Stan Li, "Learning multi-scale block local binary patterns for face recognition," in *Advances in Biometrics*, pp. 828–837. Springer, 2007.

[8] Xiaoyang Tan and Bill Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *Transactions on Image Processing*, vol. 19, no. 6, pp. 1635–1650, June 2010.

[9] S. Hegenbart and A. Uhl, "A scale-adaptive extension to methods based on LBP using scale-normalized laplacian of gaussian extrema in scale-space," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP'14)*, 2014, pp. 4352–4356.

[10] Manik Varma and Andrew Zisserman, "Classifying images of materials: Achieving viewpoint and illumination independence," in *Proceedings of the 7th European Conference on Computer Vision (ECCV'02)*. 2002, pp. 255–271, Springer-Verlag.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine region," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, 2005.

[12] F. Perronnin, Yan Liu, J. Sanchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'10)*, June 2010, pp. 3384–3391.

[13] L. Sifre and S. Mallat, "Rotation, scaling and deformation invariant scattering for texture discrimination," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'13)*, June 2013, pp. 1233–1240.

[14] M. Cimpoi, S. Maji, I Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'14)*, 2014, pp. 3606–3613.

[15] Michael Gadermayr, Andreas Uhl, and Andreas Vécsei, "Dealing with intra-class and intra-image variations in automatic celiac disease diagnosis," in *Proceedings of Bildverarbeitung für die Medizin 2015 (BVM'15)*, Mar. 2015, accepted.

[16] A. Vécsei, G. Amann, S. Hegenbart, M. Liedlgruber, and A. Uhl, "Automated marsh-like classification of celiac disease in children using an optimized local texture operator," *Computers in Biology and Medicine*, vol. 41, no. 6, pp. 313–325, June 2011.

[17] F. Perronnin and C. Dance, "Fisher kernels on visual vocabularies for image categorization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'07)*, June 2007, pp. 1–8.

[18] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, January 1996.

[19] Xiaoyang Tan and Bill Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," in *Analysis and Modelling of Faces and Gestures*, October 2007, vol. 4778 of *LNCS*, pp. 168–182.

[20] Gustaf Kylberg, "The kylberg texture dataset v. 1.0," External report (Blue series) 35, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, September 2011.

[21] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.