DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-389

Sudoku Texture Classification

Graham Finlayson, Seth Nixon. Department of Computing Sciences, University of East Anglia, Norwich, United Kingdom.

Abstract

One of the most successful features for texture recognition is the Local Binary Pattern. The LBP is the 8 digit binary number created by comparing the value of a central pixel with its 8 neighbours where 1s and 0s are assigned when respectively the central pixel is larger or smaller than its neighbour. This pattern is bit shifted circularly to its maximum value to obtain rotational invariance. Comparing histograms of LBPs provides leading texture recognition.

In our research, we rank the center pixel with all its 8 neighbours. Each pixel is substituted by a 3x3 grid where the numbers one through nine appear once and correspond to the rank of the underlying pixel values (of the local 3x3 neighbourhood) i.e. an input image is transformed to look like a Sudoku grid. Then, we read out the ranks clockwise starting with the right-most rank and appending the central pixel to the end, we then rotate to the maximum value (so achieving rotational invariance). Each 9 digit number is non-linearly mapped to the interval [0,1] so that the overall dataset histogram has a uniform distribution. By comparing the histograms of our Sudoku rank features, we observe a significant increase in recognition performance for the Outex and Curet benchmark datasets.

Introduction

The topic of texture classification hinges on the assumption that it can be quantified and/or understood using computational methods. It has been an intensive area of research since the 1960s [1]. One of the most significant early contributions was the Co-occurrence matrix proposed in [2]. The authors generated features which, given a spatial relationship, describe how many times varying grey-level intensities occur simultaneously. These features were, by necessity, very fast to compute and showed good performance on images of sandstone and aerial photography. Another classical method is the Markov Random Field (MRF). In [3] the authors model real world textures using MRFs. They show that their method can reproduce small-scale "micro-textures" well, but not larger scale "macro-textures". A long standing technique is the Gabor Filter. These are linear filters which extract orientation and frequency information and do so in a similar way to the human visual system [4]. They are commonly used in the field of texture. These include texture segmentation [5] and texture-based defect detection [6]. A more recent method is the Dual Tree Complex Wavelet Transform. This uses multiple filter-banks to extract orientation and magnitude information from the images [7]. More specifically it uses two discrete wavelet transforms in tandem to form a decomposition of the image with bands of orientation and magnitude information. While applicable to a wide range of vision applications they have shown good performance in texture classification [8, 9]. The Local Binary Pattern (LBP) is another recent method based on the idea that pixel comparisons in local neighborhoods across an image can form an index which can be used for classification [10]. Good performance has been shown in a number of areas including texture classification, face recognition [11] and medical imaging [12]. There has also been a lot of research into extending this initial idea. These methods attempt to add to the feature, or change how patterns are formed or selected to improve the baseline performance. Examples include Dominant LBP [13] and Centre Symmetric LBP [14, 1].

A significant problem in the area of texture is that no standard description of it exists and no single method works best in all conditions. Methods have been proposed which attempt to organise how we classify texture. A major contribution to this was [15], where the authors propose that texture classification can be split into four categories: statistical properties, mathematical models, geometric methods and signal processing methods. This work was then furthered by [16] where they define an entire taxonomy of texture which also considers colour. One particular description of a set of texture classification methods is the Histograms of Equivalent Patterns (HEP) [17]. This defines a framework which encompasses many benchmark methods such as LBP and Gray Level Co-occurrence Matrices . It makes the distinction that all methods which are instances of HEP partition the feature space based on image patches by applying a pre-defined function on the intensities of that patch. Our method is most comparable to LBP. We apply a function to a local 3x3 neighborhood to form a pattern and then separate these patterns to form a histogram. This places our method within the HEP framework.

Background

Our work is based on LBP. An LBP is a binary string which describes a neighborhood of pixels. It is formed by comparing a central pixel with its neighbors, if the neighbor is greater than the centre it is assigned 1, if less than it is assigned 0.

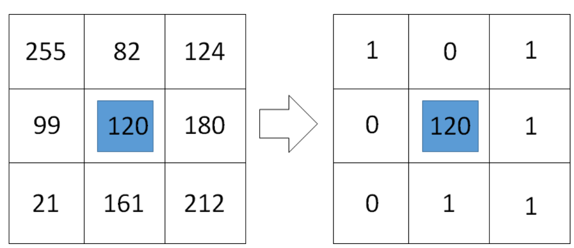

Figure 1. *LBP transformation of a local neighborhood.*

This number is then read clockwise starting from an arbitrary point and rotated to its maximum value. For reference the LBP in Figure 1 would be 11110010, or 242 in decimal. This can be expressed as

$$LBP(N_p) = \sum_{i=1}^{P} (p_i > p_c)^{2(i-1)}$$

where N is the neighbourhood, P is the number of pixels and p_c is the central pixel. Once this process has been applied to every pixel in an image a histogram of all the resultant integers forms the feature vector. This representation has two main advantages: firstly, it is invariant to any monotonic changes in the grey-scale and secondly, it is rotationally invariant. It is also very computationally efficient; the only calculations needed are inequality tests and for an eight pixel neighbourhood there are only 36 rotationally invariant 8-bit LBPs. A further addition to the LBP was the concept of uniformity. It was found that up to 95% of patterns in an image contain two or less 0 to 1 transitions. These LBPs are denoted uniform and are used to form the histogram with the remaining patterns grouped into one bin. This significantly compresses the feature length such that each histogram now has $p+2$ bins, where p is the number of points in the neighbourhood.

LBP is a benchmark method and has proven performance on many datasets. Its main drawback is the fact that it assumes the magnitude of the difference between two pixels is unimportant. An extreme case is a neighbourhood with a central pixel value of 0 and two outer pixels with values 1 and 255. Both are assigned a 1 despite the fact that one represents near black and the other white. We believe this difference is important. Our representation, while it does not encode absolute difference, does describe how a local area is organised in terms of magnitude while retaining the benefits of LBP as described above [10].

The rest of this paper is organised as follows. In the following Section we propose the method for forming our Sudoku patterns and the subsequent feature vector. We then go on to discuss classification. Finally we will discuss our results and then move on to a conclusion.

Method

We form a pattern in a similar way to LBP however instead of using just the centre pixel for comparisons we compare all pixels in a neighbourhood simultaneously to form a rank-ordering.

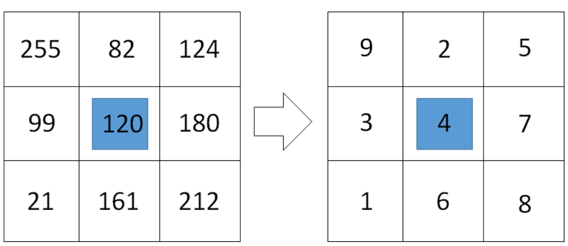

Figure 2. *Transformation of pixel intensities to rank values.*

We read this clockwise as an integer starting at the right-most pixel and appending the central pixel's rank to the end. This can be expressed as

$$S(P) = \sum_{i=1}^{n} RANK(p_i) * 10^i$$

where the $RANK$ operator assigns the rank to pixel p in neighborhood P. This number is then "bit" shifted with circular wrap around to its maximum value to achieve rotational invariance. The pattern in Figure 2 would be 786139254 and then shifted to 925478613. This process is applied to every pixel in an image to form our "Sudoku" image. Figure 3 shows transforms of three images to their Sudoku Counterparts, note how in the first image the lines are represented by lighter pixels (patterns with a higher value) and the areas in between tend to be darker.

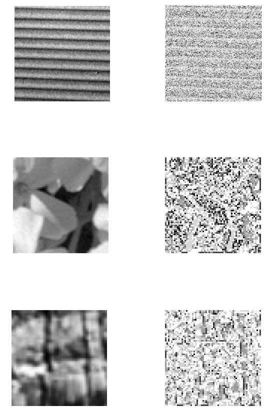

Figure 3. *Transformation of raw images to their Sudoku counterparts.*

A key feature of LBP and also our Sudoku representation is that it is invariant to all typical photometric changes that can occur when the capture conditions change (e.g. scaling, offsets or any non-linear increasing functions applied to the image). However, the Sudoku rank also has the advantage that it is a full rank order and is not based on the binary relation of a central pixel and its neighbours. As all our Sudoku patterns begin with a 9 there are only 8 numbers in each pattern which can change. As each pattern contains each number from 1 to 9 only once this means the number of rotationally invariant Sudoku patterns is $8! = 40320$. We plausibly have more information because the number of rotationally invariant LBPs is 36. Our hypothesis is that the Sudoku rank which compares all pixels with each other will capture yet more salient information and so support yet more accurate texture recognition.

Histogram Formation

Because we choose to begin the Sudoku rank with 9 all our initial ranks are large integers. In order to remove any bias in the representation–from how we make the nine digit rank number we map the calculated Sudoku rank to the interval [0,1] so that the probability of each mapped value is uniform. Specifically we would like the raw histogram of Sudoku ranks to be uniform and this can be achieved by histogram equalization [18]. A visualization of histogram equalization is that it defines an increasing function $f()$ when applied to our input data-the Sudoku ranks-have the property that the histogram of the data is uniform. In our work we apply $f()$, calculated for the whole dataset, on a per image basis. The histogram of mapped Sudoku ranks for a given image will not be uniform and is used as an index for texture recognition. See Figure 4 where the points on the x axis correspond to the upper and lower bounds for each histogram bin.

We have observed that the number of bins in each histogram can have a significant effect on the performance of our system, and that the optimal number of histogram bins varies between datasets. In Figure 5 we show a graph of percentage accuracy

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-389

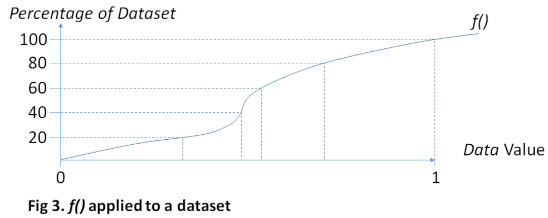

Figure 4. *Function $f()$ applied to a dataset.*

vs number of histogram bins for the Outex 13 dataset. As dimensionality increases so does average performance, however it is worth noting that there are certain very specific partitions of the feature space which provide either significant performance benefits or losses. This is interesting and will be a topic of future research. To decide on the number of histogram bins for our final result we simply choose the value with the highest percentage accuracy.

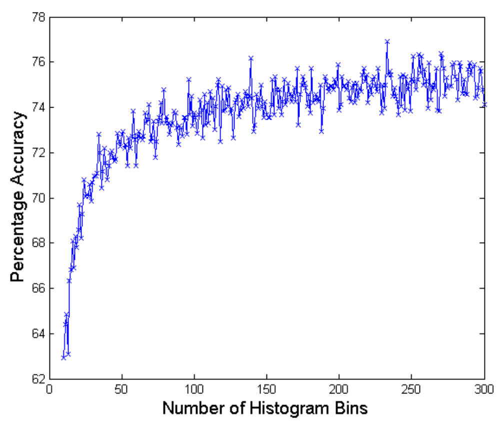

Figure 5. *Classification accuracy against number of histogram bins on the Outex_TC_00013 dataset.*

Uniformity

LBP also considers the concept of “uniformity”. This concerns the number of transitions between 0 and 1 that a pattern contains. Consider the LBPs 11111110 and 00000100. Both these patterns have “uniformity” 2 (two transitions from 0 to 1). Through statistical analysis of pattern occurrence it has been shown that in $LBP_{8,1}$ the patterns with uniformity 2 or less typically correspond to 95 percent of the patterns in the image [10]. In the LBP method we typically histogram LBPs that have uniformity of 2 or less and group all the remaining higher uniformity pixels into a single histogram bin. In our Sudoku method we apply the LBP “uniformity of 2” method to our Sudoku ranks. We do this by only considering patterns which vary above or below the central pixels rank more than twice. All other patterns are placed in a bin appended to the end of the histogram.

Classification

To classify we separate our data into a training and testing set. To do this we take every second histogram from each class and use this as our training set, the remainder we use for testing. We classify our histograms using a K Nearest Neighbor (KNN) classifier [19]. We calculate the distance between a test sample and every training sample. We then take the top K results and assign the test sample with the modal class within those K results. In our experiments we use K = 7 (empirically this gave the best result across all three datasets). The distance measure we use is the Kullback-Leibler divergence

$$KL(S,M) = \sum_{i=1}^{b} S_b log M_b$$

where S is the test sample, M is the training sample and B is the number of bins[20].

Data

We test our method on three benchmark datasets:

- Outex - A large dataset comprising multiple test suites of images with varying image conditions. We test on Outex_TC_00000 which is a grey-scale suite with 24 texture classes under constant illumination, Outex_TC_00010 which is a grey-scale suite with 24 texture classes with varying rotations and Outex_TC_00013 which is a colour suite with 68 texture classes under constant illumination. All images are 128x128 pixels.
- Curet - A dataset of 61 classes of colour images under varying illumination conditions. The images are 200x200 pixels.
- Vistex - A grey-scale dataset of 167 texture classes. Each class contains 64 patches each at 64x64 pixels.

Figure 7 shows examples of the images we use in our experiments. The colour datasets are converted to grey-scale in pre-processing. We also up-scaled the images in the Vistex dataset to twice their original size after observing a significant performance boost when doing so. Figure 6 is a graph of performance against image scale for the Vistex dataset. For these results we used 191 histogram bins.

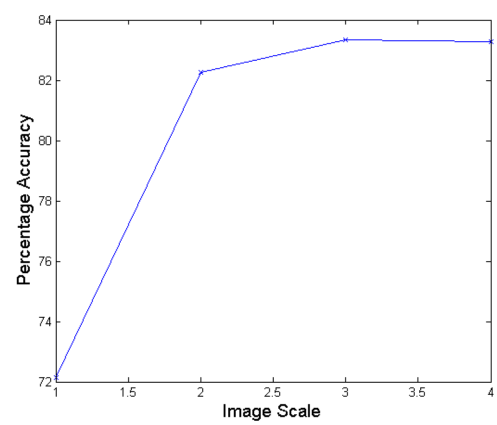

Figure 6. *Percentage accuracy against image scale for the Vistex dataset.*

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-389

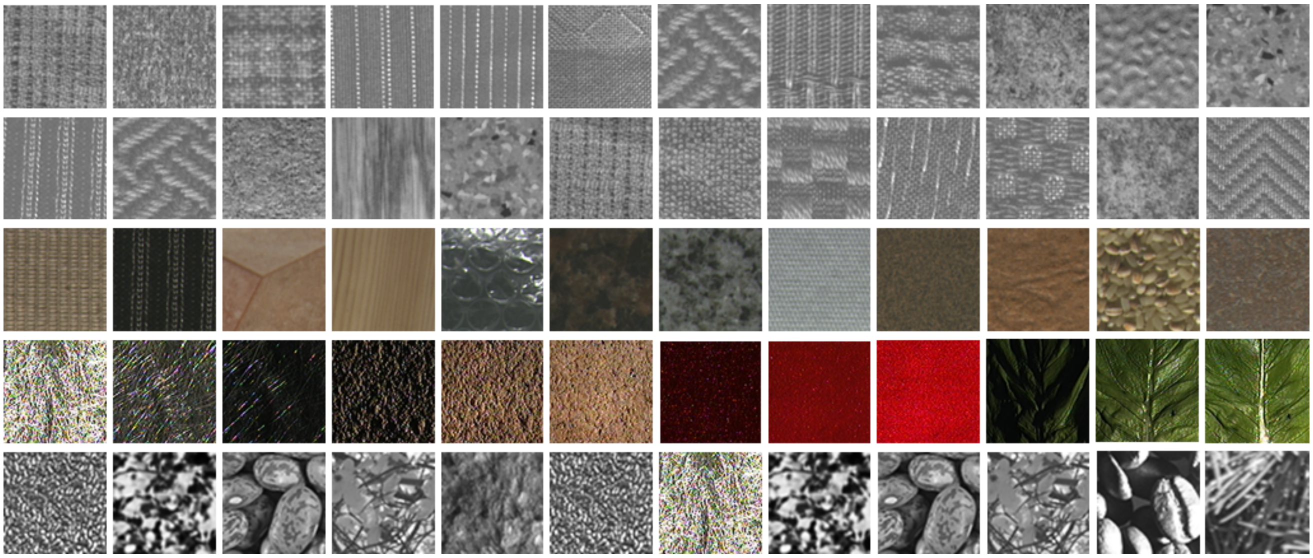

Figure 7. *Examples of images from our five experimental datasets. Row 1 is Outex_TC_00000, row 2 Outex_TC_00010, row 3 Outex_TC_00013, row 4 Curet and row 5 Vistex.*

Results

Table 1 shows our results. We compare our Sudoku method with an 8 bit uniform rotationally invariant LBP and achieve higher performance for all of our datasets.

Results of our experiments (percentage accuracy).

	Outex00	Outex10	Outex13	Curet	Vistex
Sudoku	**98.9%**	**88.9%**	**73.8%**	**77.7%**	**83.4%**
LBP	92.5%	86.1%	71.4%	76.7%	76.1%

Conclusion

In conclusion we have presented a novel feature based on the benchmark method Local Binary Patterns. Our feature considers the relationships between all pixels in a local neighborhood instead of just the central pixel. We have achieved good performance on a number of benchmark datasets outperforming LBP in every case.

References

[1] Sheryl Brahnam, Lakhmi C. Jain, Loris Nanni and Alessandra Lumini. Local binary patterns: new variants and applications. Springer Berlin Heidelberg, pg.2, 2014,

[2] Robert M. Haralick, K. Shanmugam and Its'hak Dinstein. Textural features for image classification. IEEE Transactions on Systems, Man and Cybernetics, (6), pgs.610-621. (1973).

[3] George R. Cross and Anil K. Jain. Markov random field texture models. IEEE Transactions on Pattern Analysis and Machine Intelligence, (1), pgs.25-39. (1983).

[4] Dennis Gabor, Theory of communication. Journal of the Institute of Electrical Engineers, 93, pgs429457. (1946).

[5] Anil K. Jain and farshid Farrokhnia. Unsupervised texture segmentation using Gabor Filters. IEEE International Conference on Systems, Man and Cybernetics, pgs14-19, IEEE. (1990)

[6] Ajay Kumar and Grantham K. H. Pang, Defect detection in textured materials using Gabor filters. IEEE Transactions on Industry Applications, 38(2), pgs.425-440. (2002).

[7] Ivan W. Selesnick, Richard G. Baraniuk and Nick G. Kingsbury, The dual-tree complex wavelet transform. IEEE Signal Processing Magazine, 22(6), pp.123-151. (2005).

[8] Serkan Hatipoglu, Sanjit K. Mitra, and Nick Kingsbury Texture classification using dual-tree complex wavelet transform. Seventh International Conference on Image Processing And Its Applications, (Conf. Publ. No. 465) (Vol. 1, pgs. 344-347). IET. (1999).

[9] Turgay Celik, and Tardi Tjahjadi, Multiscale texture classification using dual-tree complex wavelet transform. Pattern Recognition Letters, 30(3), pgs.331-339. (2009).

[10] Timo Ojala, Matti Pietikinen and Topi Menp, Multiresolution grayscale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), pgs.971-987. (2002).

[11] Timo Ahonen, Abdenour Hadid, and Matti Pietikinen, Face description with local binary patterns: Application to face recognition. IEEE Transactions On Pattern Analysis and Machine Intelligence, 28(12), pgs.2037-2041. (2006).

[12] Fahran Riaz, Ali Hassan, Muhammad Younis Javed and Miguel Tavares Coimbra, Detecting melanoma in dermoscopy images using scale adaptive local binary patterns. 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pgs. 6758-6761). IEEE. (2014).

[13] S. Liao, Max W. K. Law, and Albert C.S. Chung, Dominant local binary patterns for texture classification. IEEE Transactions on, mage Processing, 18(5), pgs.1107-1118. (2009).

[14] Marko Heikkil, Matti Pietikinen, and Cordelia Schmid, Description of interest regions with center-symmetric local binary patterns. Computer Vision, Graphics and Image Processing (pgs. 58-69). Springer Berlin Heidelberg. 2006.

[15] Tuceryan, Mihran and Anil K. Jane, Texture analysis. The handbook of pattern recognition and computer vision, World Scientific Publishing, 1998 pg.207-248.

[16] Francesco Bianconi, Richard Harvey, Paul Southam and Antonio

DOI: 10.2352/ISSN.2470-1173.2016.14.IPMVA-389

Fernndez. Theoretical and experimental comparison of different approaches for color texture classification. Journal of Electronic Imaging, 20(4), 043006-043006. (2011)
[17] Antonio Fernndez, Marcos X. lvarez, and Francesco Bianconi. Texture description through histograms of equivalent patterns. Journal of mathematical imaging and vision, 45(1), 76-102. (2013)
[18] Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B. Zimmerman, Karel Zuiderveld, Adaptive histogram equalization and its variations. Computer vision, graphics, and image processing, 39(3), pgs.355-368. (1987).
[19] Padraig Cunningham and Sarah J. Delany, k-Nearest neighbour classifiers. Multiple Classifier Systems, pp.1-17. (2007).
[20] S. Kullback and R.A. Leibler, On information and sufficiency. The annals of mathematical statistics, pgs.79-86. (1951).

Author Biography

Seth Nixon is a PhD student in the School of Computing Sciences at the University of East Anglia. He also conducted his undergraduate studies there and obtained a BSc in Computer Science. His work is primarily focused in the areas of colour and texture analysis.

Graham Finlayson is a professor in the School of Computing Sciences at the University of East Anglia where he leads the world renowned Colour Lab. His interests span the development of physics-based image processing algorithms, to their implementation in embedded devices and, ultimately, to their commercialisation. Colour Lab technology ships in 100s of millions of devices. Professor Finlayson serves as a fellow of the Institute for Engineering Technology, the Royal Photographic Society and the Society for Imaging Science and Technology.