International Journal of Research in Advent Technology (E-ISSN: 2321-9637) Special Issue
1st International Conference on Advent Trends in Engineering, Science and Technology
"ICATEST 2015", 08 March 2015

Snake Texture Classification Using BRINT

Keerthana A.V[#1], Ashwin M[*2]

(M.E Scholar, Department of Computer Science and Engineering, Adhiyamaan College of Engineering,India.)
(Associate Professor , Department ofComputer Science and Engineering, Adhiyamaan College of Engineering, India.)

Abstract-Binary Rotation Invariant and Noise Tolerant (BRINT) has been proposed which is a very fast, compact and also more accurate while illumination variations, noise and rotation changes. Here feature extraction has been done with local binary pattern (LBP) approach where computed local binary descriptor. These descriptors points are sampled in a circular neighborhood and provided single-scale LBP histograms. In this approach there is not needed dictionary constructor because of using different datasets. This proposed algorithm not only provide accurate solution and also performed significantly and consistently better in presence of noise due to its high distinctiveness and robustness. This method used for different level of noise such as Gaussian, salt and pepper, and speckle noise. It is highly discriminative, has low computational complexity, is highly robust to noise and rotation, and allows for compactly encoding a number of scales and arbitrarily large circular neighborhoods. These approach overcome the limitation such as the instability of the uniform patterns, the lack of noise robustness, the inability to encode a large number of different local neighborhoods, an incapability to cope with large local neighborhoods, and high dimensionality. Texture classification process involves two phases: the learning phase and the recognition phase. In the learning phase, the target is to build a model for the texture content of each texture class present in the training data, which generally comprises of images with known class labels. The texture content of the training images is captured with the chosen texture analysis method, which yields a set of textural features for each image. In our process snake texture has been taken and processed.

Keywords: Binary Rotation Invariant and Noise Tolerant, local binary pattern, Local ternary patterns and Texture classification.

1. INTRODUCTION

Texture analysis is important in many applications of computer image analysis for classification or segmentation of images based on local spatial variations of intensity or color. A successful classification or segmentation requires an efficient description of image texture. Important applications include industrial and biomedical surface inspection, for example for defects and disease, ground classification and segmentation of satellite or aerial imagery, segmentation of textured regions in document analysis, and content-based access to image databases. However, despite many potential areas of application for texture analysis in industry there is only a limited number of successful[1] examples. A major problem is that textures in the real world are often not uniform, due to changes in orientation, scale or other visual appearance. In addition, the degree of computational complexity of many of the proposed texture measures is very high.

Texture classification process involves two phases: the learning phase and the recognition phase. In the learning phase, the target is to build a model for the texture content of each texture class present in the training data, which generally comprises of images with known class labels. The texture content of the training images is captured with the chosen texture analysis method, which yields a set of textural features for each image[2]. These features, which can be scalar numbers or discrete histograms or empirical distributions, characterize given textural properties of the images, such as spatial structure, contrast, roughness, orientation, etc. In the recognition phase the texture content of the unknown sample is first described with the same texture analysis method. Then the textural features of the sample are compared to those of the training images with a classification algorithm, and the sample is assigned to the category with the best match. Optionally, if the best match is not sufficiently good according to some predefined criteria, the unknown sample can be rejected instead.

Statistical methods analyze the spatial distribution of gray values, by computing local features at each point in the image, and deriving a set of statistics from the distributions of the local features. Depending on the number of pixels defining the local feature statistical methods can be further classified into first-order (one pixel), second-order[1] (two pixels) and higher-order (three or more pixels) statistics[2]. The basic difference is that first-order statistics estimate properties (e.g. average and variance) of individual pixel values, ignoring the spatial interaction between image pixels, whereas second- and higher-order statistics estimate properties of two or more pixel values occurring at specific locations relative to each other.

Geometrical methods consider texture to be composed of texture primitives, attempting to describe the primitives and the rules governing their spatial organization. Which is defined generalized cooccurrence matrices,[1] which describe second-order statistics of edges. Model-based methods hypothesize the underlying texture process, constructing a parametric generative model, which could have created the observed intensity distribution. The intensity function is considered to be a combination of a function representing the known structural information on the image surface and an[2] additive random noise sequence. Pixel-based models view an image as a collection of pixels, whereas region-based models regard an image as a set of sub patterns placed according to given rules. An example of region-based models are random mosaic models, which tessellate the image into regions and assign gray levels to the regions according to a specified probability density function. The facet model is a pixel-based model, which assumes no spatial interaction between neighboring pixels, and the observed intensity function is assumed to be the sum of a deterministic polynomial and additive noise.

When choosing a texture analysis algorithm, a number of aspects [3]should be considered:

1. Illumination (gray scale) invariance; how sensitive the algorithm is to changes in gray scale. This is particularly important for example in industrial machine vision, where lighting conditions may be unstable.
2. Spatial scale invariance; can the algorithm cope, if the spatial scale of unknown samples to be classified is different from that of training data.
3. Rotation invariance; does the algorithm cope, if the rotation of the images changes with respect to the viewpoint.
4. Projection invariance (3-D texture analysis); in addition to invariance with respect to spatial scale and rotation the algorithm may have to cope with changes in tilt and slant angles.
5. Robustness with respect to noise; how well the algorithm tolerates noise in the input images.
6. Robustness with respect to parameters; the algorithm may have several built-in parameters; is it difficult to find the right values for them, and does a given set of values apply for a large range of textures.
7. Computational complexity; many algorithms are so computationally intensive that they cannot be considered for applications with high throughput requirements, e.g. real-time visual inspection and retrieval of large databases
8. Generativity; does the algorithm facilitate texture synthesis, i.e. regenerating the texture that was captured using the algorithm.
9. Window/sample size; how large sample the algorithm requires to be able to produce a useful description of the texture content.Section 2 describes about existing methodology, Section 3describes about proposed methods and section 4 is conclusion.

2. EXISTING SYSTEM

In this proposed approach BRINT_S Descriptor has been introduced which is a same scheme of sampling method of original LBP approach, it sample pixels around a central pixel, also restricted the number of points sampled to be a multiple of eight. It's grouping equal binary representations under rotations, assigning[1] code numbers to the resulting groups. This approach performed well such as LBP rotation invariance but it does not imply the performance when computational cost much high compared with traditional LBP descriptors.

Next BRINT_M Descriptor has been introduced to overcome the drawback of the BRINT_S Descriptor. This new method has been worked as CLBP_CSM feature extraction method and also like LBP (uniform rotation invariance) where used single features. Compared with CLBP methods this BRINT_M outperformed. BRINT_M Descriptor is achieved by combining BRINT_S and performance of the CLBP_CSM feature. Finally Multi Resolution BRINT has been proposed which performed efficient and [1] extracted from a single resolution with a circularly symmetric neighbor set. Goal of our approach is to cope with a large number of different scales and created operators for different spatial resolutions. This method operation by concatenating binary histograms from multiple resolutions into a single histogram. BRINT descriptor is noise robust, in contrast to the noise sensitivity of the traditional LBP and its many variants. Since BRINT_CSM, the joint histogram of BRINT_C, BRINT_S and BRINT_M, has a very high dimensionality of $36 * 36 * 2 = 2592$, in order to reduce the number of bins needed. This modern approach achieved the Nearest Neighbor Classifier (NNC) applied to the normalized BRINT histogram feature[2] vectors. The proposed BRINT gives the highest performance at high SNR. The proposed BRINT descriptor is noise robust, in contrast to the noise sensitivity.

Advantage

- It is applicable for discriminative and robust combination for multi resolution analysis.
- This proposed method more robustness in performance for all three Outex databases under noisy conditions.
- Our proposed method more reliable compared with state-of-the-art texture classification methods on all three Outex test suites.
- The robustness of the proposed approach to image rotation and noise has been validated.
- The proposed approach to produce consistently good classification results on all of the datasets, most significantly outperforming in high noise conditions.

A. BRINT_CSM

Proposition and analysis of a weighted combination scheme of the proposed descriptor for images and the feature for grayscale images in gender recognition using different sources of RGB-D data.

BRINT operator performs by thresholding the differences of the center value and the neighborhood in the 3x3 grid surrounding one pixel. [1] The resulting values are then considered as an 8-bit binary number represented for that pixel. The histogram of these binary numbers in the whole image can be used as a descriptor for the image.

Another remarkable improvement of BRINT is the so called uniform pattern BRINT codes are not uniformly distributed, some codes appear much more frequently than the others. These frequent codes have at most two transitions from 0 to 1 or vice versa when the pattern is traversed circularly, and are called uniform[4] patterns. When computing the histogram, every uniform pattern is labeled with one distinguished value while all the non-uniform patterns are group into one category.

It is a powerful approach to analyze and discriminate textures. However, it just considers the sign of differences and ignores the difference values, [5]which can be an important source of information. By just keeping the sign of the differences, two different textures could be misclassified as the same by BRINT.

B. Local ternary patterns

Local ternary patterns (LTP) are an extension of Local binary patterns (LBP).[1] Unlike LBP, it does not threshold the pixels into 0 and 1, rather it uses a threshold constant to threshold pixels into three values. Considering k as the threshold constant, c as the value of the center pixel, a neighboring pixel p, the result of threshold is:

$$\begin{cases} 1, & \text{if } p>c+k \\ 0, & \text{if } p>c-k \text{ and } p<c+k \\ -1 & \text{if } p<c-k \\ \end{cases}$$

In this way, each thresholded pixel has one of the three [1]values. Neighboring pixels are combined after thresholding into a ternary pattern. Computing a histogram of these ternary values will result in a large range, so the ternary[6] pattern is split into two binary patterns. Histograms are concatenated to generate a descriptor double the size of LBP.

C. Local binary patterns

Local binary patterns (LBP**)** is a type of feature used for classification in computer vision. LBP is the particular case of the Texture Spectrum model proposed. LBP was[1] first described and it has since been found to be a powerful feature for texture classification; it has further been determined that when LBP is combined with [8]the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

The Process of LBP are,

- Divide the examined window into cells (e.g. 16x16 pixels for each cell).
- For each pixel in a cell, compare the pixel to each of its 8 neighbors (on its left-top, left-middle, left-bottom, right-top, etc.). Follow the pixels along a circle, i.e. clockwise or counter-clockwise.
- Where the center pixel's value is greater than the neighbor's value, write "1". Otherwise, write "0". This gives an 8-digit binary number (which is usually converted to decimal for convenience).
- Compute the histogram, over the cell, of the frequency of each "number" occurring (i.e., each combination of which pixels are smaller and which are greater than the center).
- Optionally normalize the histogram.
- Concatenate (normalized) histograms of all cells. This gives the feature vector for the window.

D. *Completed modeling of Local Binary Pattern*

Completed modeling of Local Binary Pattern (CLBP**)** which is composed by the center gray level, sign components and magnitude components. To improve rotation invariance, considerably lower dimensionality we implemented Completed Local Binary Patterns (CLBP) which is consists of three LBP descriptors. [1] Which include information on the center pixel, signed differences, and magnitudes of differences, respectively, with the variants tested to improve the discriminative power of the original LBP operator. The three CLBP proposed CLBP_C, CLBP_S and CLBP_M. This histogram effectively has a description of the Texture on three different levels of locality: the CLBP labels for the histogram contain information about the patterns on a pixel-level, the labels are summed over a small region to produce information on a regional level and the regional histograms are concatenated to build a global description of the Texture. It should be noted that when using the histogram based methods the regions do not need to be rectangular. Both do they need to be of the same size nor shape, and they do not necessarily have to cover the whole image. It is also possible to have partially overlapping regions.

The LBP methodology has led to significant progress in texture analysis. It is widely used all over the world both in research and [9] applications. Due to its discriminative power and computational simplicity, the method has been very successful in many such computer vision problems which were not earlier even regarded as texture problems, such as face analysis and motion analysis.

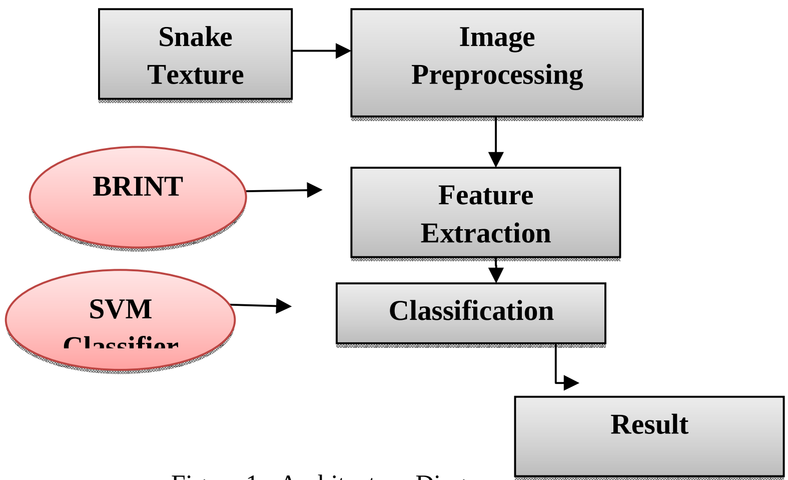

Figure 1: Architecture Diagram

Preprocessing

In pre-processing we are applying Gaussian filtering to our input image. Gaussian filtering is often used to remove the noise from the image. Here we used wiener function to our input image. **Gaussian filter** is windowed filter of linear class, [1] by its nature is weighted mean. Named after famous scientist Carl Gauss because weights in the filter calculated according to Gaussian distribution.

The Gaussian Smoothing Operator performs a weighted average of surrounding pixels based on the Gaussian distribution. It is used to remove Gaussian noise and is a realistic model of defocused lens. Sigma defines the amount of [10] blurring. The radius slider is used to control how large the template is. Large values for sigma will only give large blurring for larger template sizes. Noise can be added using the sliders.

Feature Extraction

Feature selection is the process of selecting a subset of relevant features for use in model construction. The central assumption when using a feature [12] selection technique is that the data contains many redundant or irrelevant features. Redundant features are those which provide no more information than the currently selected features, and irrelevant features provide no useful information in any context. Feature selection techniques are a subset of the more general field of feature extraction. Feature extraction[1] creates new features from functions of the original features, whereas feature selection returns a subset of the features. Feature selection techniques are often used in domains where there are many features and comparatively few samples (or data points). The archetypal case is the use of feature selection in analyzing DNA microarrays, where there are many thousands of features, and a few tens to hundreds of samples. Feature selection techniques provide three main benefits when constructing predictive models:Improved model interpretability, shorter training times, enhanced generalization by reducing overfitting.

BRINT has to reduce Dimensionality [1] Reduction and Rotation Invariance, Discriminative Power and Noise Robustness. BRINT_CSM, the joint histogram of BRINT_C, BRINT_S and BRINT_M, has a very high dimensionality of $36 * 36 * 2 = 2592$, in order to reduce the number of bins needed. The BRINT operator assigned a label to every pixel of a gray level image. The label mapping to a pixel is affected by the relationship between this pixel and its eight neighbors of the pixel. If we set the gray level image is I, and Z0 is one pixel in this image.

SVM CLASSIFIER

SVM maps input vectors to a higher dimensional vector space where an optimal hyper plane is constructed. Among the many hyper planes available, there is only one hyper plane that maximizes the distance between itself and the nearest data vectors of each category. This hyper plane which maximizes the margin is called [1] the optimal separating hyper plane and the margin is defined as the sum of distances of the hyper plane to the closest training vectors of each category.

(i).Data setup: our dataset contains three classes, each N samples. The data is 2D plot original data for visual inspection

(ii).SVM with linear kernel (-t 0). We want to find the best parameter value Cusing 2-fold cross validation (meaning use 1/2 data to train, the other 1/2 to test).

(iii).After finding the best parameter value for C, we train the entire data

again using this parameter value

(iv). plot support vectors

(v). plot decision area

Expression for hyper plane

$w.x+b = 0$

x – Set of training vectors

w – vectors perpendicular to the separating hyper plane

b – offset parameter which allows the increase of the margin

Kernel function is used when decision function is not a linear function of the data and the data will be mapped from the input space through a nonlinear transformation rather than fitting non-linear curves to the vector space to separate the data

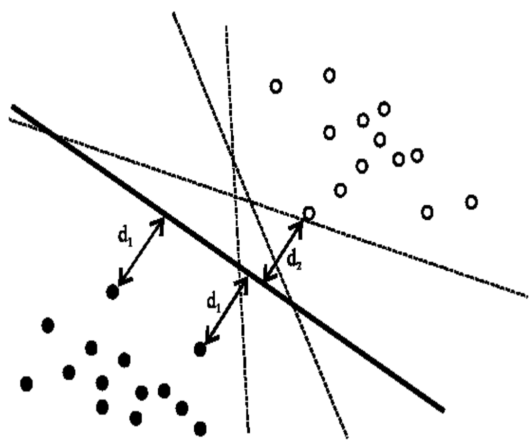

Figure 2: Margin is d1+d2

With an optimal kernel function implemented in SVM model, the classification task is able to scale high dimensional data relatively well, tradeoff between classifier complexity and classification error can be controlled explicitly.

Classification by n-class SVM

This defines a grouping of all the classes in two disjoint groups of classes. This grouping is then used to train a SVM classifier in the root node of the decision tree, using the samples of the first group as positive examples and the samples of the second group as negative examples. The classes from the first clustering group are being assigned to the first (left) subtree, while the classes of the second clustering group are being assigned to the (right) second subtree. The process continues recursively until there is only one class per group which defines a leaf in the decision tree.

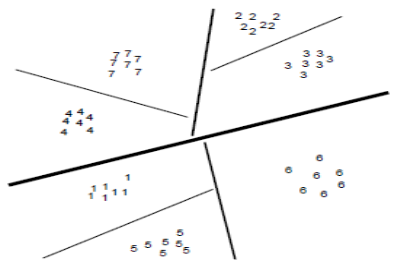

Figure 3:Decision tree

3. PROPOSED SYSTEM

In proposed method going to implement detection and classification of texture by computing feature extraction. There are three feature extraction [1] such as BRINT_CSM, Statistical Pattern and Gray level Co-Occurrence features will be extracted. Finally RVM classifier can be used to classify the texture based on the extracted features.

A. RVM Algorithm

A relevance vector machine (RVM) is a machine learning technique that uses Bayesian inference to obtain parsimonious solutions for regression and probabilistic classification.[1] The RVM has an identical functional form to the support vector machine, but provides probabilistic classification.

A relevance vector machine [Tipping 2001] provides a regression method in a Bayesian framework. It can be also adapted to perform classification tasks. Like Support Vector Machines (SVM) it learns a sparse representation of input basis functions. In its original form it only has a single dimensional output. This is a drawback in some regression tasks with multi-dimension outputs (e.g. human body pose estimation), since we have to use a separate relevance vector machine for each output dimension and will lead to separate sets of basis functions being selected for each output dimension, reducing the sparsity. To avoid this, we propose an extension which enables a single relevance vector machine to handle multiple output dimensions. We also extend the fast bottom-up basis function selection algorithm [Tipping 2003] to the multivariate output case.

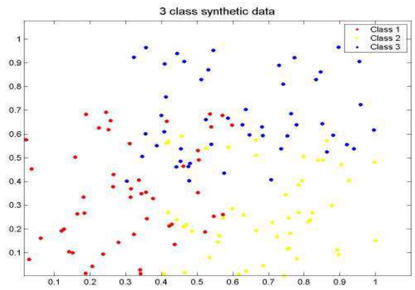

Figure 2: 3 class synthetic data in RVM classifier

It is developed based on the probabilistic Bayesian learning framework. The RVM process is an iterative one and involves repeatedly re-estimating and until a stopping condition is met. Our hyper parameter values and which result from the procedure are those that maximize marginal likelihood. Hence are those used when making a new estimate of a target value t for a new input x0.RVM is a probabilistic non-linear model with a prior distribution on the weights that enforces sparse solutions. It is reported that RVM can yield nearly identical performance to, if not better than, that of SVM while using far fewer relevance vectors than the number of support vectors for SVM in several benchmark studies. Compared with SVM, it is not necessary for RVM to tune any regularization parameter during the training phase, neither for kernel function to satisfy Mercer's condition. Furthermore, the predictions are probabilistic. For regression problems, the RVM makes predictions based on the function.

4. CONCLUSION

The proposed BRINT descriptor is noise robust, in contrast to the noise sensitivity of the traditional LBP and its many variants. **T**he proposed idea can be generalized and integrated with existing LBP variants, such as conventional LBP, rotation invariant patterns, rotation invariant uniform patterns, CLBP and Local Ternary Patterns (LTP) to derive new image features for texture classification. Highly effective multi-resolution descriptor for rotation invariant texture classification. The proposed approach firmly puts rotation invariant binary patterns back on the map, after they were shown to be very ineffective.

REFERENCE

[1] L. Liu, B. Yang, P. Fieguth, Z. Yang, and Y. Wei, "BRINT: A Binary Rotation Invariant and Noise Tolerant Texture Classification," in Proc. IEEE Transactions on Image Processing vol. 23, NO. 7, JULY 2014 3071

[2] X. Xie and M. Mirmehdi, "A galaxy of texture features," in *Handbook of Texture Analysis*, M. Mirmehdi, X. Xie, J. Suri, Eds. London, U.K.:Imperial College Press, 2008, pp. 375–406.

[3] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolutiongray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[4] L. Liu and P. Fieguth, "Texture classification from random features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 574–586, Mar. 2012.

[5] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *Int. J. Comput. Vis.*, vol. 73, no. 2, pp. 213–238, 2007.

[6] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images," *Int. J. Comput.Vis.*, vol. 62, nos. 1–2, pp. 61–81, 2005.

[7] M. Varma and A. Zisserman, "A statistical approach to material classification using image patches," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2032–2047, Nov. 2009.

[8] M. Crosier and L. D. Griffin, "Using basic image features for texture classification," *Int. J. Comput.Vis.*, vol. 88, no. 3, pp. 447–460, 2010.

[9] B. S. Manjunathi and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18,no. 8, pp. 837–842, Aug. 1996.

[10] L. Liu, B. Yang, P. Fieguth, Z. Yang, and Y. Wei, "BRINT: A binary rotation invariant and noise tolerant texture descriptor," in *Proc. IEEEICIP*, Sep. 2013, pp. 255–259.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1265–1278, Aug. 2005.

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