

# Texture Classification using Random Forest Classifier based on Local Vector Quantization Pattern

**Anumol Antony**

*PG Student*

*Department of Computer Science & Engineering  
Federal Institute of Science and Technology*

**Dr. Arun Kumar M N**

*Associate Professor*

*Department of Computer Science & Engineering  
Federal Institute of Science and Technology*

## Abstract

The Nearest Neighbor (NN) Classifier is the fundamental and simplest algorithm used for classification. It has successfully been used in classification and pattern recognition applications, such as texture recognition. However, it has a disadvantage that it has lower predictive accuracy. In order to overcome these disadvantages, we propose a new classification method using random forest classification based on local vector quantization pattern (LVQP). In LVQP, different kinds of texture images are chosen to train a local pattern codebook, where each different structural pattern is described by a unique codeword index. Contrarily to the original LBP and its many variants, LVQP does not quantize each neighborhood pixel separately to 0/1, but aims at quantizing the whole difference vector between the central pixel and its neighborhood pixels. Since LVQP deals with the structural pattern as a whole, it has a high discriminability and is less sensitive to noise.

**Keywords: Nearest Neighbor Classifier, Pattern Recognition, Texture Recognition, Random Forest, Local Vector Quantization Pattern**

## I. INTRODUCTION

Texture classification is a fundamental problem in computer vision with a wide variety of applications. The goal of texture classification is to assign an unknown sample image to one of a set of known texture classes. Texture classification is one of the four problem domains in the field of texture analysis. The other three are texture segmentation (partitioning of an image into regions which have homogeneous properties with respect to texture; supervised texture segmentation with a priori knowledge of textures to be separated simplifies to texture classification), texture synthesis (the goal is to build a model of image texture, which can then be used for generating the texture) and shape from texture (a 2D image is considered to be a projection of a 3D scene and apparent texture distortions in the 2D image are used to estimate surface orientations in the 3D scene).

Two fundamental issues in texture classification are how to characterize textures using derived features and how to define a robust distance/similarity measure between textures. Because images of the same underlying texture can vary significantly, textural features must be invariant to (large) image variations and at the same time sensitive to intrinsic spatial structures that define textures.

Texture is an important feature of objects in an image. The perception of texture is believed to play an important role in the human visual system for recognition and interpretation. There has been a great interest in the development of texture based pattern recognition methods in many different areas, especially in the areas of industrial automation, remote sensing and medical diagnosis. Texture classification passes through the difficult step of texture representation or description. What is seen as a relatively easy task to the human observer becomes a difficult challenge when the analysis is made by a computational algorithm.

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. 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.

In the past decades, a lot of algorithms focusing on extracting features from local patterns in texture classification have been proposed. Designing robust algorithms for rotation, spatial-resolution and illumination changes is still a challenge in texture classification.

In the proposed algorithm, each difference vector finds its best match codeword in the codebook, where codeword index is defined as the LVQP code of the central pixel in a local region. After the LVQP code of each pixel of the image is generated, the

algorithm proceeds by generating a histogram of the LVQP code as the classification feature to represent the whole texture image. The LVQP algorithm achieves higher classification accuracy and is more robust to noise.

In order to achieve higher predictive accuracy, random forest classifier is used. The Random Forests algorithm is one of the best among classification algorithms – able to classify large amounts of data with accuracy. Random Forests are an ensemble learning method (also thought of as a form of nearest neighbor predictor) for classification and regression that construct a number of decision trees at training time and outputting the class that is the mode of the classes output by individual trees.

Random Forest is a versatile machine learning method capable of performing both regression and classification tasks. It also undertakes dimensional reduction methods, treats missing values, outlier values and other essential steps of data exploration. It is a type of ensemble learning method, where a group of weak models combine to form a powerful model.

## II. RELATED WORKS

In [1], Ojala et al. proposed an efficient method, namely Local Binary Pattern (LBP), for rotation invariant texture classification based on uniform local binary patterns and nonparametric discrimination of sample and prototype distributions. Uniform patterns were recognized to be a fundamental property of texture as they provide a vast majority of local texture patterns in examined textures, corresponding to texture microstructures such as edges. By estimating the distributions of these microstructures, we combined structural and statistical texture analysis. It was proposed to encode the pixel-wise information in the texture images. LBP is a simple yet efficient operator to describe local image pattern, and it has achieved impressive classification results on representative texture databases. The algorithm of LBP contains two main steps, i.e., thresholding step and encoding step. In the thresholding step, the values of neighbour pixels are turned to binary values (0 or 1) by comparing them with the central pixel. Obviously, the local binary grayscale difference information is extracted in the thresholding step. In the encoding step, the binary numbers are encoded to characterize a structural pattern, and then the code is transformed into decimal number. Aiming at achieving rotation invariance, Ojala proposed rotation invariant uniform LBP, in which only rotation invariant uniform local binary patterns were selected. It was believed that LBP is an excellent measure of the spatial structure of local image texture since it can effectively detect micro-structures (e.g., edges, lines, spots) information.

In [2] Zhibin Pan proposes a unified projection method for fast search of vector quantization. The method basically consists of five steps. They are extract interest point descriptors, construct visual word dictionary, build spatial histograms, create intersection kernels and finally train an SVM. Two key ideas are used. They are Pyramid Match Kernel and Spatial Matching Scheme. Using the pyramid matching, an approximate correspondence between two sets of vectors  $X$  and  $Y$  is found in a  $d$ -dimensional feature space. Informally, pyramid matching works by placing a sequence of increasingly coarser grids over the feature space and taking a weighted sum of the number of matches that occur at each level of resolution. At any fixed resolution, two points are said to match if they fall into the same cell of the grid; matches found at finer resolutions are weighted more highly than matches found at coarser resolutions. In Spatial Matching Scheme quantize all feature vectors into  $M$  discrete types, and make the simplifying assumption that only features of the same type can be matched to one another. Each channel  $m$  gives us two sets of two-dimensional vectors,  $X_m$  and  $Y_m$ , representing the coordinates of features of type  $m$  found in the respective images. The final kernel is then the sum of the separate channel kernels. This method achieves high accuracy on a large database.

In [3] Liao et al proposed a novel approach to extract image features for texture classification. The approach uses Dominant LBP (DLBP), which operates by discarding some patterns with fewer occurrences in the histogram and by treating them as noise during the classification process. The proposed features are robust to image rotation, less sensitive to histogram equalization and noise. It comprises of two sets of features: dominant local binary patterns (DLBP) in a texture image and the supplementary features extracted by using the circularly symmetric Gabor filter responses. The dominant local binary pattern method makes use of the most frequently occurred patterns to capture descriptive textural information, while the Gabor-based features aim at supplying additional global textural information to the DLBP features. The proposed method achieves the highest classification accuracy in various texture databases and image conditions.

In [4] Manik Varma proposed a statistical approach to texture classification from single images without any knowledge of illumination or viewing conditions. A single texture can appear vastly different with changes in illumination and viewing direction. The statistical approach is developed where textures are modelled by the joint probability distribution of filter responses. This distribution is represented by the frequency histogram of filter response cluster centres (textons) and choosing the closest model from the histogram set. Recognition proceeds from single, uncalibrated images and the novelty here is that rotationally invariant filters are used and the filter response space is low dimensional.

In [5] X. Tan proposed a method for making recognition more reliable under uncontrolled lighting conditions is one of the most important challenges for practical face recognition systems. We tackle this by combining the strengths of robust illumination normalization, local texture-based face representations, distance transform based matching, kernel based feature extraction and multiple feature fusion. The main contributions are as follows: 1) a simple, efficient image preprocessing chain whose practical recognition performance is comparable to or better than current (often much more complex) illumination normalization methods; 2) a rich descriptor for local texture called LTP that generalizes LBP while fragmenting less under noise in uniform regions; 3) a distance transform based similarity metric that captures the local structure and geometric variations of LBP/LTP face images better than the simple grids of histograms that are currently used; and 4) a heterogeneous feature fusion-based recognition framework that combines two popular feature sets Gabor wavelets and LBP with robust illumination

normalization and a kernelized discriminative feature extraction method. The combination of these enhancements gives the state-of-the-art performance on three well-known large-scale face datasets that contain widely varying lighting conditions.

In [6] Guo et al proposed the completed LBP (CLBP) for texture classification. It is developed by combining the conventional LBP with the measures of local intensity difference and central gray level. The CLBP descriptor aims to preserve additional distinguishing information by using three components: the sign descriptor, the magnitude descriptor and the central pixel descriptor. A local region is represented by its center pixel and a local difference sign-magnitude transform (LDSMT). The center pixels represent the image gray level and they are converted into a binary code, namely CLBP-Center (CLBP C), by global thresholding. LDSMT decomposes the image local differences into two complementary components: the signs and the magnitudes, and two operators, namely CLBP-Sign (CLBP S) and CLBP-Magnitude (CLBP M), are proposed to code them. The traditional LBP is equivalent to the CLBP S part of CLBP, and we show that CLBP S preserves more information of the local structure than CLBP M, which explains why the simple LBP operator can extract the texture features reasonably well. By combining CLBP S, CLBP M, and CLBP C features into joint or hybrid distributions, significant improvement can be made for rotation invariant texture classification.

In [7] Jie Chen, proposed a simple, very powerful and robust local descriptor, called the Weber Local Descriptor (WLD). It is based on the fact that human perception of a pattern depends not only on the change of a stimulus (such as sound, lighting) but also on the original intensity of the stimulus. Specifically, WLD consists of two components: differential excitation and orientation. The differential excitation component is a function of the ratio between two terms: one is the relative intensity differences of a current pixel against its neighbors; the other is the intensity of the current pixel. The orientation component is the gradient orientation of the current pixel. For a given image, we use the two components to construct a concatenated WLD histogram. WLD is robust to noise appearing in a given image. WLD is robust to changes in illumination.

In [8] Z.B. Pan proposed an improved fast training method in the Hadamard domain to reduce computational complexity of the Linde-Buzo-Gray (LBG) algorithm. This method uses an optimally-ordered Hadamard transform kernel through relocating energy distribution in k-dimensional space. It then apply a three-step elimination criterion to speed-up the training process. The advantage of this method is that it reduces search space and arithmetical operations and rforms better when generating a larger codebook.

In [9] Yang Zhao proposed a new local descriptor that discards the structural information from LBP operator, which is named as Local Binary Count (LBC). The proposed LBC can extract the local binary grayscale difference information, and totally abandon the local binary structural information. The most discriminative information of local texture for rotation invariant texture classification is not the micro-structures information but the local binary grayscale difference information. The proposed CLBC achieve comparable accurate classification rates. The completed LBC (CLBC) is also proposed to enhance the performance of texture classification. It allows slight computational savings in the process of training and classification.

In [10] Zhen Lei proposed a method to learn a discriminant face descriptor (DFD) in a data-driven way. The idea is to learn the most discriminant local features that minimize the difference of the features between images of the same person and maximize that between images from different people. It enhances the discriminative ability of face representation in three aspects. First, the discriminant image filters are learned. Second, the optimal neighborhood sampling strategy is soft determined. Third, the dominant patterns are statistically constructed. Discriminative learning is incorporated to extract effective and robust features. The proposed method is applied to the heterogeneous (cross-modality) face recognition problem. Coupled image filters are learned to reduce the gap between features of heterogeneous face images to improve the performance of this challenging problem.

In [11] Zhibin Pan proposed texture classification using local pattern based on vector quantization. Local binary pattern (LBP) is a simple and effective descriptor for texture classification. But it has two main disadvantages: 1) different structural patterns sometimes have the same binary code and 2) it is sensitive to noise. In order to overcome these disadvantages, we propose a new local descriptor named local vector quantization pattern (LVQP). In the proposed algorithm, each difference vector finds its best match codeword in the codebook, where codeword index is defined as the LVQP code of the central pixel in a local region. After the LVQP code of each pixel of the image is generated, the algorithm proceeds by generating a histogram of the LVQP code as the classification feature to represent the whole texture image. The LVQP algorithm achieves higher classification accuracy and is more robust to noise.

### III. METHODOLOGY

In the proposed algorithm, each difference vector finds its best match codeword in the codebook, where codeword index is defined as the LVQP code of the central pixel in a local region. After the LVQP code of each pixel of the image is generated, the algorithm proceeds by generating a histogram of the LVQP code as the classification feature to represent the whole texture image.

#### A. Difference Vector of Local Pattern

The difference vector of a central pixel and its neighbors for a local pattern is defined as  $(g_0 - g_c, g_1 - g_c, \dots, g_{p-1} - g_c)^T$ . The difference vector preserves plenty of discriminative information over the local pattern, and whether the information can be extracted effectively or not has a great influence on classification accuracy.

LBP and LTP convert the elements of the difference vector to binary/ternary format respectively, but ignore inter-elements relationships, as shown in Fig.3.1. The essence of this kind of process is a scalar quantization process for the elements in the difference vector.

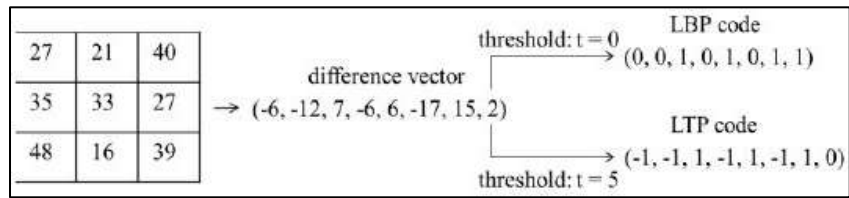


Fig. 1: Quantization for LBP and LTP difference vector.

The scalar quantization process causes loss of information in the difference vector. In fact, in case of low noise or in a at image area, similar patterns can also have quite different LBP features. To tackle these shortcomings, the Vector Quantization (VQ) method is applied to the difference vector, since it is more robust against noise or at image areas.

**B. Vector Quantization**

Vector Quantization is a high-efficiency data compression algorithm. In the VQ method, initially a codeword table, usually called codebook, is trained. Codebook training is performed through Linde-Buzo-Gray(LBG) algorithm. LBG algorithm is the most widely used training algorithm. In LBG algorithm, during the initial codebook design phase the random selection strategy is used. Here S codewords are randomly selected from the training vectors. Then, the final codebook Y is generated iteratively. The LBG algorithm is given as follows:

- Step 1: Generate an initial codebook Y0 of size S. Set the iteration counter i = 0 and the initial average distortion D<sub>0</sub> = 1. Set the maximum iteration counter as I and the distortion threshold as ε.
- Step 2: For each training vector x, find its best match codeword with the least distortion in the current codebook Yi.
- Step 3: Assign the training vectors into S cells and update the centroid of each cell to obtain a new codebook Yi+1
- Step 4: Calculate the current average distortion Di for all training vectors at the i<sup>th</sup> iteration.
- Step 5: If (Di-1 - Di)/Di ≤ ε or i = I, set the ultimate codebook Y = Yi + 1 and the LBG algorithm is completed. Otherwise, let i = i+1, return to Step 2.

The codebook consists of S L-dimensional code words yi = (yi0, yi1, ..., yi(L-1))<sup>T</sup> (i = 0, 1, . . . , S - 1), where subscript "i" represents the index of the codeword yi . VQ calculates the Euclidean distance between each codeword and the input vector x = (x0, x1, ..., xL-1)<sup>T</sup> as follows

$$ED(x, y_i) = \sum_{j=0}^{L-1} (x_j - y_{ij})^2 \tag{1}$$

and finds the best match codeword y<sub>bm</sub> in the codebook for the input vector x as shown in Eq.(2)

$$ED(x, y_{bm}) = \min\{ED(x, y_i), i = 0, 1, \dots, S - 1\} \tag{2}$$

**C. Local Vector Quantization Pattern**

In LVQP, first the images without rotation is chosen as training images, calculating all the difference vectors for each central pixel with its neighbors. Then, it proceed in training a local pattern codebook with all the difference vectors through LBG. Codebook size is predefined. Throughout the paper, it denote the LVQP with codebook size S as LVQP(S).

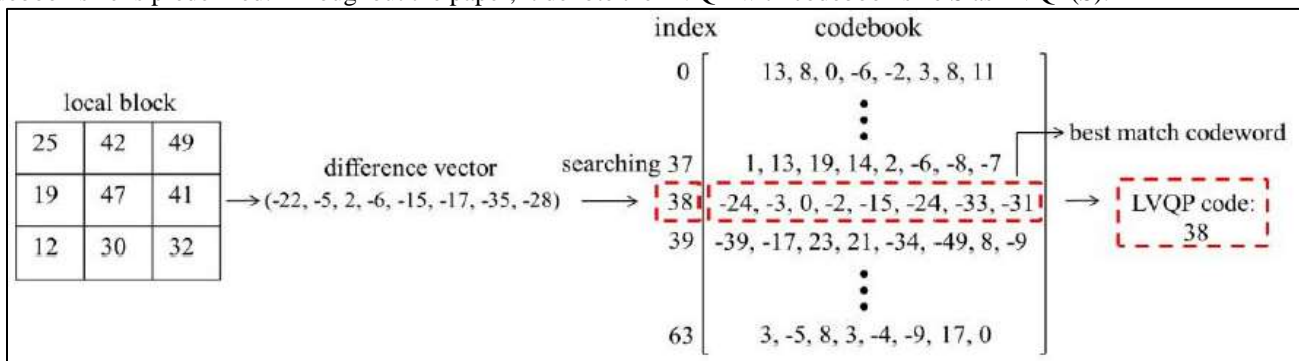


Fig. 2: LVQP code generation for a difference vector through Vector Quantization.

With LVQP, the difference vectors of each pixel of the input image are generated and assign the index of the best match codeword as LVQP code, as shown in Fig.2.

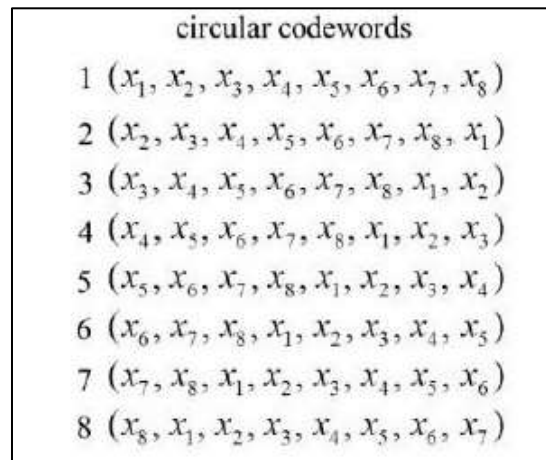


Fig. 3: Eight circular codewords for searching (R=1, P=8).

To achieve rotation invariance, each codeword is circularly calculated P times during the codebook search, as visually described in Fig.3.

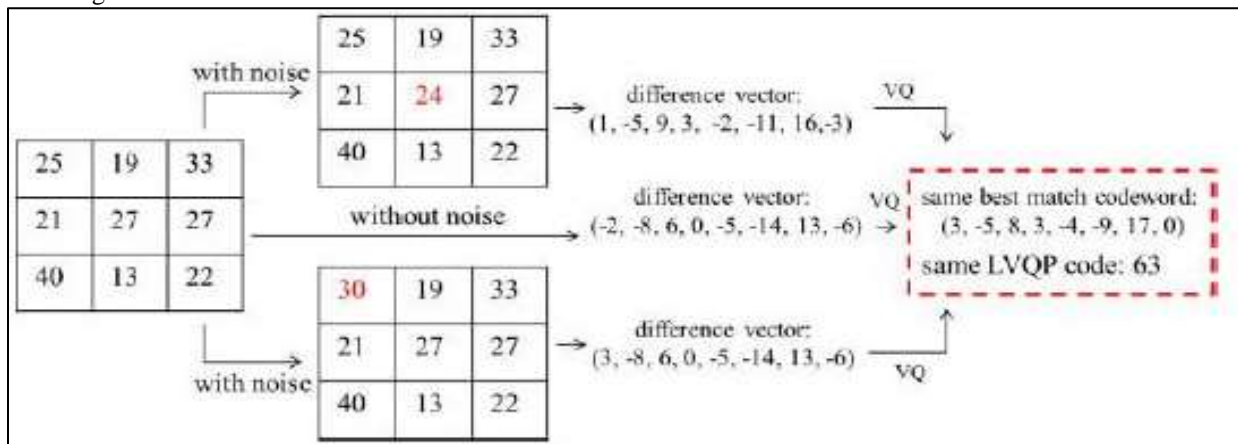


Fig. 4: LVQP finds the same code even pixels corrupted by noise.

In LVQP, difference vectors can be easily affected by noise polluted pixels. However, the minimum Euclidean distance approach used here for finding the best match codeword overcomes the noise effects and lead to the same LVQP code, as shown in Fig.4.

This is because LVQP quantizes the difference vector as a whole so that it reduces the impact of noise corruption over pixels in a local pattern. LBP or LBP- like methods, instead, quantizes each dimension of the difference vector separately which leads to entirely different codes even with a small noise. Noise might change the values of some dimensions of a difference vector, but when it computes Euclidean distance with a codeword in the codebook, it behaves like if a small noise was averagely added to all the pixels. Therefore, noise pollution usually does not prevent LVQP from finding the correct best match codeword, or LVQP code, increasing the noise robustness of the proposed method.

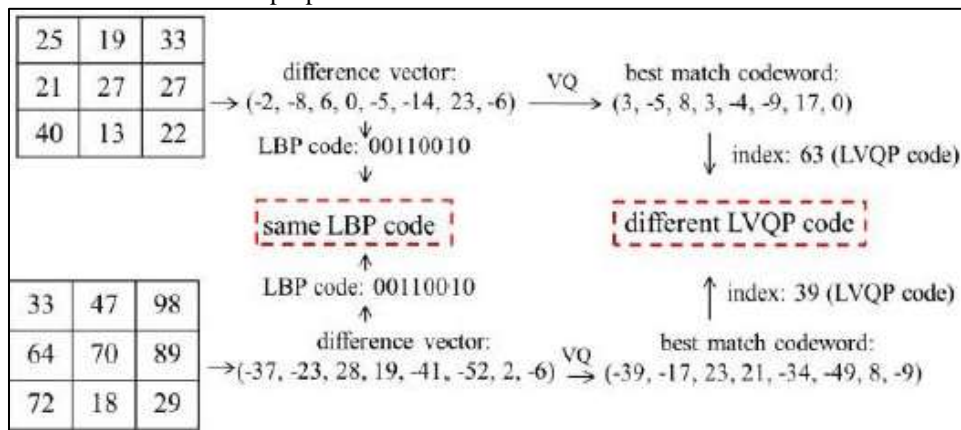


Fig. 5: Different structural patterns with different LVQP code

In Fig.5, it is shown that different structural patterns find different best match codewords in the codebook and are characterized by different LVQP codes.

The central pixel, which directly describes the image gray level, also presents effective discriminative information. In fact, the gray level of the central pixel expresses whether it is a bright point or a dark point, when compared with its neighbors in the local region. Therefore, in order to achieve robustness against illumination changes, a local adaptive threshold is proposed instead of the conventional fixed global threshold. The central pixel is converted by the local adaptive threshold into a binary format named LVQP\_C. We first divide the image into non-overlap blocks of 3 x 3 pixels and assign the same threshold to pixels in the same block. Then, a mean filter of n x n blocks is used to identify the local adaptive threshold of the central block in the n x n blocks.

Random forest algorithm is the statistical and machine learning algorithm which uses multiple learning algorithms to obtain better predictive performance than others. It is a method used for classification, regression and other tasks that operate by constructing a multiple decision trees at training time and outputting the class of the texture image. Random decision forests correct for decision trees' habit of overfitting to their training set.

In Random Forest, multiple trees are grown as opposed to a single tree in CART model. To classify a new object based on attributes, each tree gives a classification and it is said to be the tree votes for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

It works in the following manner. Each tree is planted and grown as follows:

- 1) Random Record Selection: Each tree is trained on roughly 2/3rd of the total training data (exactly 63.2%). Cases are drawn at random with replacement from the original data. This sample will be the training set for growing the tree.
- 2) Random Variable Selection: Some predictor variables (say, m) are selected at random out of all the predictor variables and the best split on these m is used to split the node.

Note: By default, m is square root of the total number of all predictors for classification. For regression, m is the total number of all predictors divided by 3.

The value of m is held constant during the forest growing.

Note: In a standard tree, each split is created after examining every variable and picking the best split from all the variables.

- 3) For each tree, using the leftover (36.8%) data, calculate the misclassification rate - out of bag (OOB) error rate. Aggregate error from all trees to determine overall OOB error rate for the classification.
- 4) Each tree gives a classification, and it is said as the tree "votes" for that class. The forest chooses the classification having the most votes over all the trees in the forest. For a binary dependent variable, the vote will be YES or NO, count up the YES votes. This is the RF score and the percent YES votes received is the predicted probability. In regression case, it is average of dependent variable.

Note: Random Forest does not require split sampling method to assess accuracy of the model. Self-testing possible even if all data is used for training as 2-3rd of available training data is used to grow any one tree and the remaining one-third portion of training data always used to calculate out-of bag error to assess model performance.

#### IV. EXPERIMENTAL RESULT

The method was implemented in MATLAB 2012 prototype. The experiment was carried out using the texture database Outex. To validate the developed method, the computational results obtained by implementing the developed method is evaluated and compared with the methods presented by earlier researchers.

The database contains materials images under significant viewpoint variations. In the texture classification experiment, N training images are randomly chosen from each class, where N = (5, 10, 15, 20, 40), while the remaining images are used as test set.

The proposed texture classification method includes codebook generation, database building, feature extraction, training and predicting.

Initially Codebook training is performed through the Linde-Buzo-Gray (LBG) algorithm. Figure 4.1 shows the codebook generated.



Fig. 6: Codebook Generated

After the generation of the codebook, the database is built. Each image in the database is read one by one. Then find the LVQP code of each image in the database and its corresponding LVQP histogram. This histogram is stored so that it is used for the classification.

Next the training of the forest is done using the random forest classifier. A random forest is an ensemble of decision trees which will output a prediction value. Each decision tree is constructed by using a random subset of the training data. It constructs a multiple decision trees at training time.

To classify the input image, initially an image is read as input from the test set.



Fig. 7: Input image

Next the LVQP code of each image is calculated. With LVQP, the difference vectors of each pixel of the input image are generated and assign the index of the best match codeword as LVQP code. Fig. 8 shows the LVQP image of the input image.

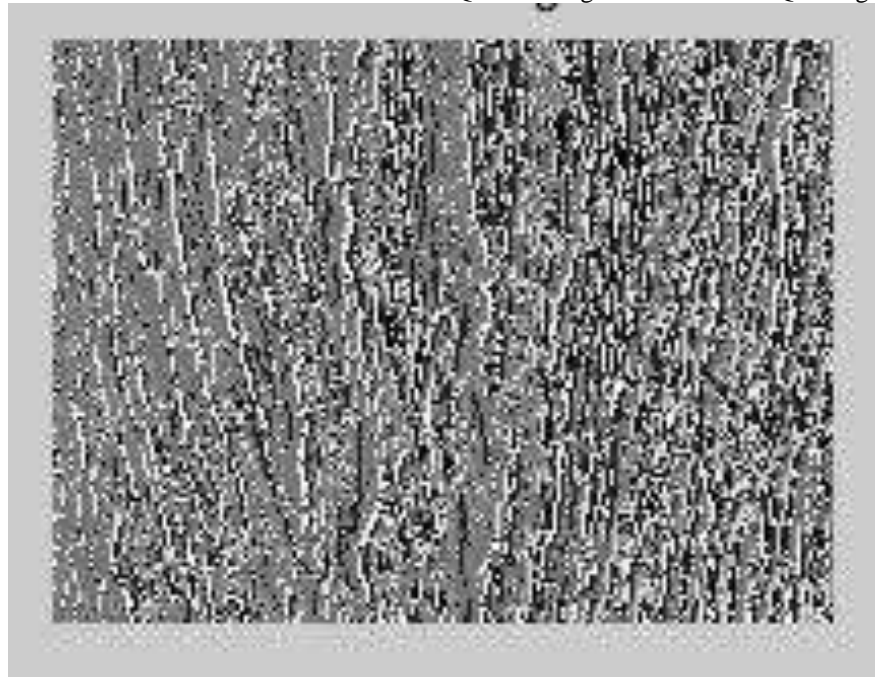


Fig. 8: LVQP image

Using the LVQP image, corresponding LVQP histogram is generated as shown in Fig. 9. The histogram in image processing indicates the histogram image density values. Calculating the histogram of an image provides a vector of length 256 that expresses the image. This histogram represents the feature of the input image.

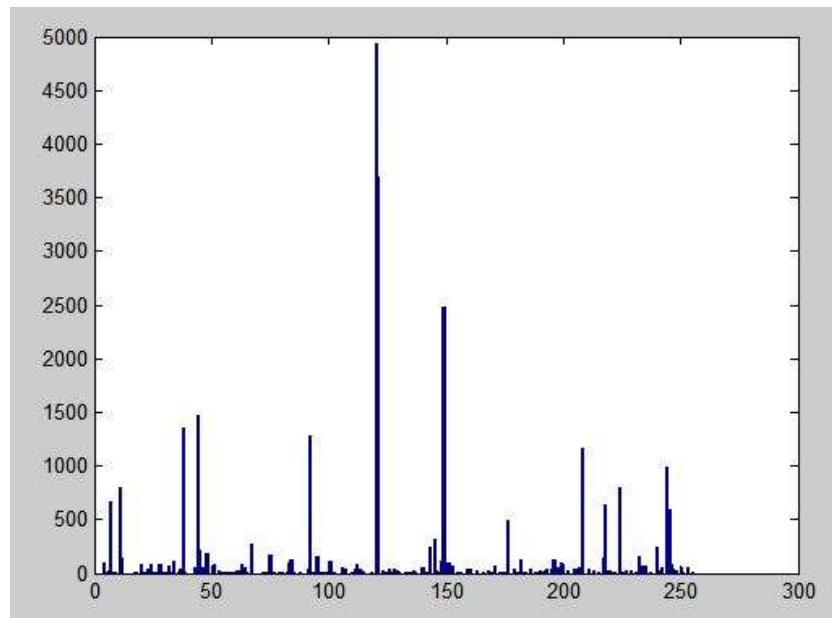


Fig. 9: LVQP histogram

The histogram image is used as the input vector of the RF classifier. The RF classifier then predicts the class of the input image.

```
Command Window

classnum =

     2

predictedClass =

     2

correct
fx >>
```

Fig. 10: Class Prediction

In order to calculate the prediction accuracy, each test row is passed through RF classifier. The algorithm provides approximately 92% accuracy.

```
Command Window
correct
Processing 47 of 50
correct
Processing 48 of 50
wrong
Processing 49 of 50
wrong
Processing 50 of 50
wrong

Prediction Accuracy is: 92.000000
fx >>
```

Fig. 11: Prediction Accuracy

A receiver operating characteristic (ROC), or ROC curve, is a graphical plot that illustrates the performance of a classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

If the outcome from a prediction is  $p$  and the actual value is also  $p$ , then it is called a true positive (TP); however if the actual value is  $n$  then it is said to be a false positive (FP). Conversely, a true negative (TN) has occurred when both the prediction outcome and the actual value are  $n$ , and false negative (FN) is when the prediction outcome is  $n$  while the actual value is  $p$ .

To draw a ROC curve, only the true positive rate (TPR) and false positive rate (FPR) are needed (as functions of some classifier parameter). The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

In Fig. 12, the diagonal divides the ROC space. Points above the diagonal represent good classification results (better than random), points below the line represent poor results (worse than random). Each prediction result or instance of a confusion matrix represents one point in the ROC space.

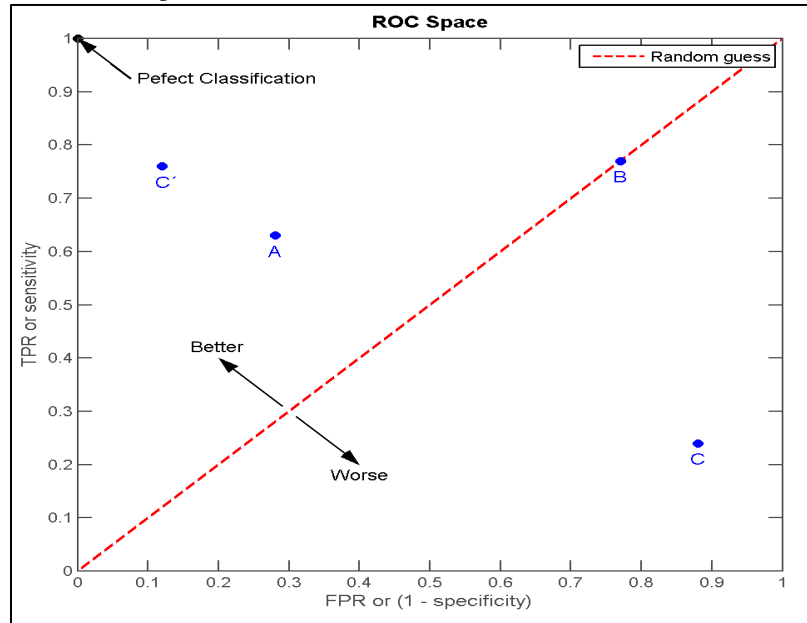


Fig. 12: ROC space

Fig. 12 shows the ROC curve of the random forest classification. The curve shows that the random forest classifier gives a good classification result.

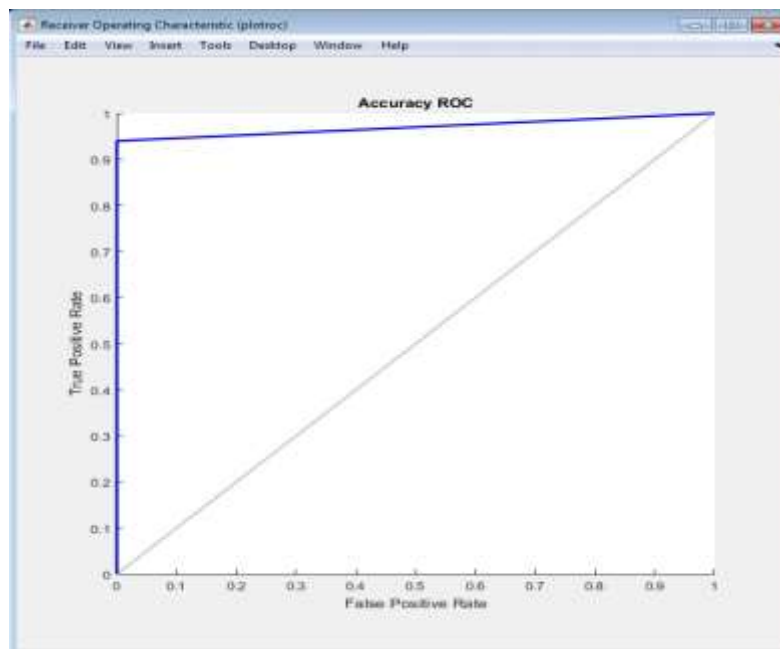


Fig. 13: ROC curve

## V. CONCLUSION

In this work, a new method for texture classification using random forest classifier based on Local Vector Quantization Pattern (LVQP) is proposed. In LVQP, we quantize the difference vector as a whole by searching it in a trained codebook and use the index of the best match codeword as the LVQP code for the central pixel. To further increase the predictive accuracy, random forest classifier is used. The experimental result shows that the proposed method has higher texture classification accuracy than nearest neighbour method.

## REFERENCES

- [1] T. Ojala, M. Pietikinen, and T. Menp, "Multiresolution gray-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.
- [2] Z. Pan, K. Kotani, and T. Ohmi, "A unified projection method for fast search of vector quantization", IEEE Signal Process. Lett., vol. 11, no. 7, pp. 637-640, Jul. 2004.
- [3] S. Liao, M. W. K. Law, and A. C. S. Chung, , "Dominant local binary patterns for texture classification", IEEE Trans. Image Process., vol. 18, no. 5, pp. 1107-1118, May 2009.
- [4] M. Varma and A. Zisserman, "A statistical approach to texture classification from single images", Int. J. Comput. Vis., vol. 62, no. 1, pp. 61-81, Apr. 2005.
- [5] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions", IEEE Trans. Image Process., vol. 19, no. 6, pp. 1635-1650, Jun. 2010.
- [6] Z. Guo, L. Zhang, and D. Zhang, "A completed modeling of local binary pattern operator for texture classification", IEEE Trans. Image Process., vol. 9, no. 16, pp. 1657-1663, Jun. 2010.
- [7] J. Chen et al., "WLD: A robust local image descriptor", IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1705-1720, Sep. 2010.
- [8] Z. B. Pan, G. H. Yu, and Y. Li, "Improved fast LBG training algorithm in Hadamard domain", Electron. Lett., vol. 47, no. 8, pp. 488-489, Apr. 2011.
- [9] Y. Zhao, D.-S. Huang, and W. Jia, "Completed local binary count for rotation invariant texture classification" IEEE Trans. Image Process., vol. 21, no. 10, pp.4492-4497, Oct. 2012.
- [10] Z. Lei, M. Pietikainen, and S. Z. Li, "Learning discriminant face descriptor", IEEE Trans. Pattern Anal. Mach. Intell., vol. 36, no. 2, pp. 289-302, Feb. 2014.
- [11] Zhibin Pan, Hongcheng Fan, and Li Zhang, "Texture Classification Using Local Pattern Based on Vector Quantization", IEEE Trans. Image Process., vol. 24, no. 12, Dec. 2015