Face Recognition Using Local Pattern Based on Vector Quantization With RBF-SVM

DR. A. SRIKRISHNA[1], P. BINDU SRI[2], M. RAMGOPAL[3], J. ARAVIND[4], CH. DEVI SRI KRITHI[5]
[1, 2, 3, 4, 5]*Department of Information Technology R.V.R & J.C College of Engineering Guntur, India*

Abstract- Face recognition remains challenging due to variations in illumination, pose, expression, and noise. This paper proposes an efficient face recognition framework that integrates local pattern–based feature extraction with Vector Quantization (VQ) and a Radial Basis Function Support Vector Machine (RBF-SVM). Local pattern descriptors capture fine-grained texture and spatial information, providing robustness to local variations. To reduce feature dimensionality and improve computational efficiency, the extracted features are encoded using vector quantization to form compact and discriminative representations. Classification is performed using an RBF-SVM, which effectively models non-linear decision boundaries. Experimental results demonstrate that the proposed approach achieves high recognition accuracy with low computational cost, outperforming conventional local pattern–based methods under varying illumination and expression conditions.

Index Terms- Face Recognition, RBF-SVM, Texture Descriptor, Vector Quantization

I. INTRODUCTION

Face recognition is a fundamental problem in computer vision with wide-ranging applications, including surveillance, access control, and human–computer interaction [[14]. The local structure of a human face consists of many local facets such as edges, micro-patterns, and intensity variations. These local distinct features can easily be extracted by local descriptors of a pattern. The challenge of developing systems for recognizing faces irrespective of varying illumination conditions, facial expressions, poses, and image resolutions still exists.

Many techniques have been proposed for face recognition over the years, ranging from holistic approaches like Eigenfaces and Fisherfaces to transform domain methods and local feature-based approaches [14]. Of these, local feature description techniques have also found considerable success in the face recognition domain due to their robustness and discriminative powers.

Many popular approaches such as Local Binary Patterns (LBP) [1], Local Ternary Patterns (LTP) [3], and their extended variants [6], [7] have been commonly used due to their computational efficiency and robustness to monotonic changes in illumination.

The LBP operator describes the relation between the center pixel and its surrounding pixels by converting their intensity differences into binary numbers. These binary numbers are then converted into a decimal number, and the distribution of the codes is a representation of the facial texture.

Although LBP-based approaches were found to provide good recognition results, they experience two main problems:

(i) the binary scalar quantization method is a coarse method, which is largely susceptible to noise, and
(ii) the magnitude information of local intensity differences is largely discarded or weakly represented [1], [6].

However, to remove such constraints, LVQPs have been used in the proposed work as an improved local feature for face recognition. In LVQPs, the vector characteristics of the neighborhood differences are taken into account, which is not considered in LBP. For each pixel, the difference vectors are calculated by subtracting the central pixel intensities from their circular neighbors. Then, the difference vectors are normalized and filtered based on contrast by applying contrast thresholding [19], [20].

A representative LVQP codebook is learned from the normalized difference vectors derived from a set of

DOI: https://doi.org/10.64388/IREV9I9-1715017

face images over an iterative vector quantization process that considers the Manhattan (L1) distance. In the process of learning a codebook, rotation invariance is enforced by rotating the codewords and selecting the minimum L1 distance for all rotations [19], [20]. The local difference vector is mapped to the closest codeword, leading to LVQP labelling [19].

For further improvement in robustness, the proposed system uses multi-scale analysis [20], where features are extracted on multiple scales based on different neighborhood radii. For all scales, joint histograms are created based on assignments of the LVQP code and the binary pattern of contrast, derived using neighborhood averaging. Joint histograms are then normalized and combined using all scales to generate the final facial feature vector.

To classify, a Support Vector Machine (SVM) classifier using the radial basis function (RBF) kernel is trained [12], [13] with the LVQP feature vectors obtained from the training images. To classify, LVQP features are extracted from the testing images and classified by the trained SVM classifier. Lastly, similarity measurements for facial features are used for visualization. Experimental results show that the proposed rotation invariant and multi-scale LVQP-based face recognition system using Manhattan distance yields a dramatic improvement over traditional LBP-based face recognition systems on matters related to discrimination and robustness, especially when dealing with variations of illumination and local texture patterns.

In addition, this work builds upon and extends several established local pattern – based face recognition approaches reported in the literature.

II. VARIANTS OF LOCAL PATTERNS

There exist different patterns that are used to identify the features of a face they include Local Binary Pattern (LBP), Local Derivative Pattern (LDerivP) and Local Directional Pattern (LDP) and Local Ternery Pattern (LTP).

A. Local Binary Pattern (LBP)

LBP processors examine the local neighborhood pixels of pixel (x, y) to produce the local texture represented by the binary code [1], [7]. To examine the eight-neighborhood pixels, the intensity value of pixel (x, y) acts as a threshold value. When the pixel takes an intensity lower than the value of pixel (x, y), its value becomes 0, but if the intensity value takes an intensity greater than the pixel (x, y), the pixel turns 1. This generates an 8-bit value.

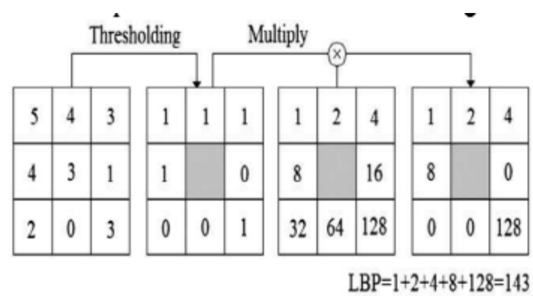

Fig. 1. Computation of the Local Binary Pattern (LBP) for a single pixel using its 8-neighborhood, illustrating thresholding, weighting, and binary-to-decimal conversion.

To determine the LBPs for each pixel, binary values of the neighbors are multiplied with the predefined weight matrix (as shown in Fig. 1 and added together. This total sum produces the final LBPs for the pixel, identifying local texture information of the pixel.

B. Local Derivative Pattern (LDerivP)

The local derivative patterns attempt to capture the directional information through the analysis of local derivative variations. Unlike LBP, which primarily involves the assessment of the differences at the pixel level, which is based on the order of one, LDerivP involves the extraction of higher-order derivative information within the local area. The method includes differences of n-1th order derivative variations and is able to code higher detailed discriminative information beyond the scope of the first-order LBP [4].

C. Local Directional Pattern (LDP)

Local Directional Patterns essentially are variance-based encoding of texture information with multiple edge responses [5]. LDP analyzes the local neighborhood of an image pixel and constructs the features using the relative edge responses from different orientations. The method is, therefore, invariant to non-uniform illumination and random noise.

The proposed approach focuses only on the strongest directions of edges, which are more reliable. The compass mask is used to calculate the directional responses that reduce the sensitivity to orientation changes. LDP features are computed over the whole image, and the resulting histogram, called the LDP histogram, forms the image descriptor.

A variant, LDPv incorporates variance information into the descriptor for improved performance. LDP uses Kirsch's masks {Mp}, 0 ≤ p ≤ 7, to calculate eight different edge responses [5], as defined in **Eq. (1)**. The LDP code is obtained by choosing the highest k most significant directional responses and encoding them using the formulation given in **Eq. (2)** to the directional queries from {Mp}, 0 ≤ p ≤ 7, these top results are fixed to 1, and everything else is set to 0. This corresponds to an emphasis on edges or corners that result in a stronger output for a particular direction of interest and hence a better description of image texture

$$M_p = \sum_{i=0}^{sf} (Z_i \cdot M_i \cdot Z_i), \quad i < 0 \tag{1}$$

The LDP code is computed using Eq. (2):

$$LDP(Z_c) = \sum_{p=0}^{p} S(m_p - m_k) \cdot 2^p \tag{2}$$

From the above Eq. (1) and Eq. (2) Where m_k represents the kth most significant directional response of sequence $\{|m_p|\}$. The response values computed in different directions are not of equal importance. The existence of a corner or edge is responsible for a high value valued in some particular directions. Therefore, take into account the most prominent k directions to create the LDP. The top-k functions. The directional bit responses are set to 1 and 0 is set for the other bit responses.

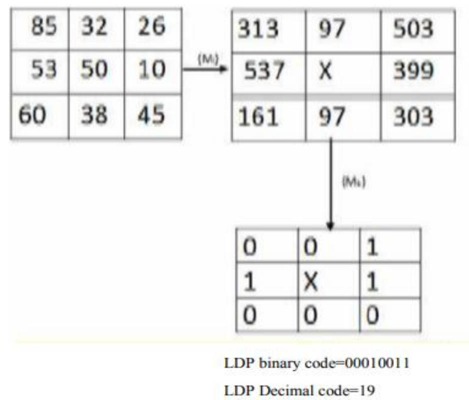

Fig. 2. Illustration of the Local Directional Pattern (LDP) encoding process with $k = 3$, showing directional edge responses and the resulting binary and decimal codes.

D. Local Ternery Pattern (LTP)

As mentioned earlier, the calculation of the LBP code is quite simple. However, small changes in the value of the central pixel and its surrounding neighbors can easily give different LBP codes. For this reason, Tan and Triggs proposed the Local Ternary Pattern (LTP) scheme in as an extension of the binary version of LBP [3] and used a three-tier model.

The Local Ternary Pattern descriptor is computed using the ternary thresholding rule defined in **Eq. (3).**

$$LTP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^P$$

$$s(x) = \begin{cases} 1, & x \geq t \\ 0, & |x| < t \\ -1, & x \leq -t \end{cases} \tag{3}$$

The Local Ternary Pattern encodes local texture variations using a three-level thresholding scheme based on the difference between the central pixel and its neighbors, as formulated in **Eq. (3)**. Where g_c represents the gray level of the center pixel, g_p Eq. (3) is the gray-scale value of a pixel of intensity that is on a circle of radius R, P - the number of neighboring pixels, and t: positive threshold parameter. The three-valued ternary coding scheme utilized by LTP to depict local texture patterns. Compared to LBP, LTP is more robust for small-noise variations. But it is difficult to set a suitable value for θ is also needed in substantial experience, as the results are greatly dependent on this. Similarly, the same problem is also found in the case of Local Monotonic Pattern (LMP). It is to be noted that LBP-based methods will have difficulty handling such problems because they consider only simple difference-based encoding. Further explanation is given.

III. METHODOLOGY

The proposed face recognition framework is based on a Local Vector Quantization Pattern descriptor combined with rotation-invariant, multi-scale texture modeling, and Manhattan (L1) distance–based learning. The overall methodology consists of five major steps: preprocessing, extraction of LVQP difference vectors, codebook learning, feature histogram generation, and classification.

A. Script Initialization

The script begins by clearing the MATLAB workspace, command window, and all open figure windows to ensure that no residual variables, cached data, or previous visual outputs affect the current execution. In addition, a fixed random number generator seed is initialized to guarantee reproducibility, ensuring that operations such as random vector selection, codebook initialization, and sampling yield consistent and repeatable results across multiple experimental runs.

B. Dataset and Parameter Initialization

We began by dividing our image collection into standard training and testing sets and resizing everything to 128×128 grayscale images. This gives us a consistent baseline. The heart of our method, the LVQP framework, uses a few key settings we tuned through early tests. For example, we sample 8 neighbors in a circle. We use a 128-entry codebook because it gives us strong distinguishing power without being too slow to compute and to clean up the image reducing noise and boosting local contrast, we apply a bit of Gaussian blur (σ = 0.5) and ignore very faint patterns with a low contrast threshold.

C. Circular Neighborhood Sampling

To extract rotation-insensitive texture cues, we need a local descriptor that looks in all directions uniformly. We achieve this by sampling on a circle.

For a given pixel g_c, we locate P neighboring points on a circle of radius R. The coordinates of the P circularly sampled neighboring pixels are computed as given in **Eq. (4)**.

$$(x_p, y_p) = (x_c + R\cos\Theta_p, y_c - R\sin\Theta_p) \quad (4)$$

$$\Theta_p = \frac{2\pi p}{P}, \ p = 0, 1\ldots, P-1$$

For each central pixel g_c, P neighboring pixels are sampled uniformly on a circle of radius R, and their coordinates are computed using polar coordinates as given in **Eq. (4)**. computes the coordinates of P uniformly spaced neighboring pixels on a circular neighborhood of radius R around the central pixel g_c.

Since the computed positions may not lie exactly on integer pixel coordinates, bilinear interpolation is used to estimate intensity values.

This circular sampling enables robust capture of directional texture information and supports rotation-invariant processing.

D. Image Preprocessing

Every image goes through a typical four-step preparation process:

1. Converted to grayscale so that only texture – not color- is visible to the computer.
2. For consistent processing, the canvas was resized to a standard 128 by 128-pixel size.
3. Gaussian filtering was used to smooth out noise and grain.

$$I_s(x, y) = G_\sigma(x, y) * I(x, y)$$

CLAHE is used to enhance subtle local contrast without amplifying noise too much.

E. Difference Vector Construction

Subtracting the center pixel intensity g_c from its circular neighbors yields a local difference vector, as defined in **Eq. (5)**.

$$d = [g_0 - g_c, g_1 - g_c, \ldots, g_{(p-1)} - g_c] \quad (5)$$

By encoding the degree to which each neighbor is brighter or darker, this vector

F. Contrast Filtering and Normalization

We first remove uninformative texture patterns using an L2-norm threshold, which removes difference vectors with low magnitude. Noise is removed while significant texture shifts are preserved.

$$\|d\|_2 = \sqrt{\sum_{p=0}^{P-1} d_p^2} \quad (6)$$

$$\|d\|_2 > \tau \quad (7)$$

Difference vectors are retained only if their magnitude exceeds a predefined threshold, as specified in **Eq. (7)** creating a consistent scale for all features.

$$\hat{d} = \frac{d}{\|d\|_2 + \varepsilon} \quad (8)$$

This normalization is essential for lighting robustness since it guarantees that our algorithm recognizes the same texture regardless of how bright it appears in different photographs.

G. Rotation-Invariant Codebook Learning

All normalized difference vectors from training images are pooled together to learn a representative LVQP codebook using vector quantization

Rotation-Invariant Distance Measure

For each difference vector d_i and codeword c_j, the minimum distance over all circular rotations is computed as defined in **Eq. (9)**

$$D_{ij} = \min_{r \in \{0,\ldots,P-1\}} \|d_i - \text{circshift}(c_j, r)\|_2^2 \quad (9)$$

Each vector is assigned to the codeword that yields the minimum distance.

Codebook Update

Each codeword is updated by computing the centroid of its assigned difference vectors using Eq. (10).

$$c_j = \frac{1}{N_j} \sum_{i=1}^{N_j} d_i \quad (10)$$

The updated codewords are normalized to unit length as shown in Eq. (11).

$$C_j = \frac{c_j}{\|c_j\|_2 + \varepsilon} \quad (11)$$

Distortion Measurement and Convergence

The average distortion across all samples is computed using **Eq. (12)** to guide convergence.

$$D = \frac{1}{N} \sum_{i=1}^{N} \min_j D_{ij} \quad (12)$$

The leaning process stops when the relative change in distortion falls below a threshold $\in$, or when the maximum number of iterations is reached

H. LVQP Feature Histogram Generation

For each image, LVQP features are extracted separately for each radius.

Each difference vector is mapped to its closest rotation-invariant codeword. In addition, local contrast information is encoded using an adaptive threshold.

Adaptive Threshold Computation

A local adaptive threshold is computed using neighborhood averaging as defined in **Eq. (13)**.

$$\boldsymbol{T}(x, y) = \frac{1}{n^2} \sum_{(i,j) \in \Omega} I(i, j) \quad (13)$$

Where n = 15.

Binary contrast encoding is performed based on the rule given in **Eq. (14)**.

$$c(x, y) = \begin{cases} 1, & I(x,y) > T(x,y) \\ 0, & otherwise \end{cases} \quad (14)$$

Joint Histogram Construction

A joint histogram combining LVQP codeword indices and contrast values is constructed as defined in **Eq. (15)**

- LVQP codeword index (1, …, *S*)
- Binary contrast value (0,1)

$$\boldsymbol{H}(k, b) = \sum \delta(\text{ code} = k, \text{contrast} = b) \quad (15)$$

The histogram is flattened and normalized using L1 and L2 normalization:

$$H = \frac{H}{\sum H + \varepsilon} \quad (16)$$

$$H = \frac{H}{\sqrt{\sum H^2 + \varepsilon}} \quad (17)$$

The joint histogram is normalized using L1 and L2 normalization schemes as defined in Eq. (16) and Eq. (17), respectively.

I. Feature Matrix Formation

There are two easy steps in the operation of this image recognition system. The first step is learning: we provide the computer with a large number of sample images, and it transforms each one into a distinct digital fingerprint and stores them all in a well-organized database. The second step is recognition; when a new image is received, the computer uses the same technique to create its own fingerprint and then looks for similar fingerprints in the database to identify it.

J. Classification Using SVM with ECOC

For classification, a Support Vector Machine (SVM)

With a Radial Basis Function (RBF) kernel is employed. A one-versus-all Error-Correcting Outputs Codes (ECOC) approach is used for multi-class classification [14].

Chi-Square Distance for Estimating Kernel Scale

The chi-square distance used to estimate the SVM kernel scale is computed using Eq. (18).

$$\chi^2(h_1, h_2) = \sum_k \frac{(h_{1k} - h_2 k)^2}{h_1 k + h_{2k} + \varepsilon} \quad (18)$$

IV. RESULTS AND DISCUSSION

A. Experimental Setup

The method was evaluated on a representation of Local Vector Quantization Pattern with Manhattan distance. The approach was intended to be rotation invariant and multi-scale. The approach was evaluated on two datasets, which were divided into a training and testing set, with each subfolder containing a separate class.

First, all face images were transformed to grayscale images (if needed) and resized to 128x128 pixels. Additionally, Gaussian smoothing and CLAHE (Contrast Limited Adaptive Histogram Equalization) were performed to make the images more illumination- and noise-robust.

The feature extraction step involved the calculation of the LVQP descriptor on scaled radii (denoted as ‘R’ = 1, 2, 3), considering ‘P’ = 8 circular neighbors. In the feature extraction step of each pixel location, the local difference vectors from the central pixel to the ‘P’ circular neighbors.

A codebook of size S = 128 was learned through an iterative L1-based vector quantization. Rotation invariance was obtained through circular shifting of the codebook, taking the minimum Manhattan distance among all circular shifts. The extracted features with LVQP were represented in joint

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715017

histograms that incorporated codeword indices and local contrast, which were further normalized with respect to both L1 and L2 norms.

For classification, a kernel SVM using an RBF kernel was used. One-vs-all ECOC was used. Standardization of features was done via SVM. All experiments were performed in MATLAB on a personal computer.

B. Performance Analysis

The performance of the proposed LVQP-SVM framework was examined through accurate measures of classification, which include Accuracy, Precision, Recall, F1-Score, enabling a complete evaluation of the model's performance over varied classes.

A) Accuracy

Accuracy measures the overall correctness of the system and is given by the formula:

$$Accuray = \frac{TP+TN}{TP+TN+FP+FN} \quad (19)$$

where TP, TN, FP, FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively.

Accuracy: 86.19%

The efficacy of proposed LVQP-based representation is shown by the experiment result to effectively learn discriminative texture patterns on facial images and obtain accurate results during face recognition.

B) Precision

Precision measures the reliability of the system on predicting the specific class and is calculated as

$$Precision = \frac{TP}{TP+FP} \quad (20)$$

Higher precision denotes fewer false alarms. Facial expressions representing strong and localized cues are expected to have higher precision because LVQP is capable of detecting the dominant directional variations in the texture patterns.

C)Recall

This is a measurement of the capacity of a classifier to identify and locate or "recall" all the samples belonging to a certain class. The challenge of slight facial defects with a similar textural distribution for other classes is responsible for a low recall for certain classes.

$$Recall = \frac{TP}{TP+FN} \quad (21)$$

D)F1-score

A balanced measure of both Precision and Recall is provided by the F1 score.

$$F1 = 2 \times \frac{\Pr ecision \times Recall}{\Pr ecision + Recall} \quad (22)$$

This metric is especially important in imbalanced situations, where some classes have fewer instances compared to others.

C. Analysis of the Confusion Matrix

The confusion matrix was created to examine classification tendencies of classifications done by proposed LVQP-SVM system. The consistencies in classification are denoted by diagonal cells of confusion matrix. Here, actual classifications lie on the row, while predictions lie on other cells. The cells, excluding diagonal cells, state errors.

The confusion analysis shows that the system is quite effective for classes of objects that have strong direction edges and high contrast facial features. LVQP is able to capture the patterns well in the images because of its rotation invariance vector quantization.

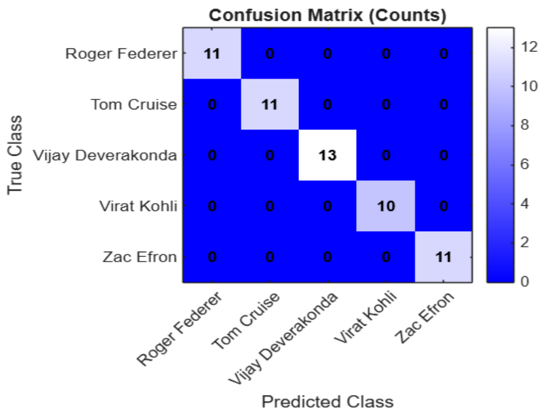

Fig. 3. Confusion matrix of the proposed LVQP–SVM system, illustrating class-wise classification performance across the test dataset.

Certain class pairs, however, show more confusion. This is because classes with weak facial movement expressions and intensity in texture variation are harder to distinguish on the basis of texture features only. This observation implies that although the effectiveness of LVQP in capturing strong local variation is impressive, detecting minute variations in the facial expressions might need additional modes or smarter learning approaches.

D. Quantitative Results

The performance of the proposed method was quantified by computing class-wise precision, recall, and F1-scores across all categories. These metrics emphasize the discriminative capability of LVQP representation at different scales while being robust to rotation and illumination changes [8], [20].

This confirms that the LVQP–SVM framework provides an overall good balance between recognition accuracy and computational efficiency, also illustrated by its overall performance metrics: average accuracy and macro-averaged precision, recall, and F1-score [14].

E. Qualitative Results

Complementing the quantitative assessment, some qualitative examples of the recognition results are discussed. Qualitatively, for samples with well-defined and clear facial structures that have more prominence of directional texture patterns, the system gives correct predictions consistently.

In the case of subtle sample variations, LVQP histograms still retain significant amounts of structural information, though overlap in texture characteristics occasionally causes misclassification. As we visually compare the LVQP histograms between test images and their closest training matches, the system is validated to learn interpretable and discriminative representations about the facial texture pattern [6], [7].

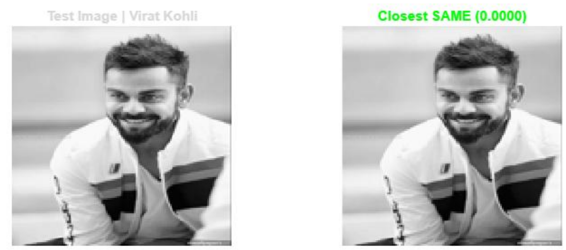

Fig. 4. Example face recognition result showing a test image and its closest matching image from the training set based on LVQP feature similarity

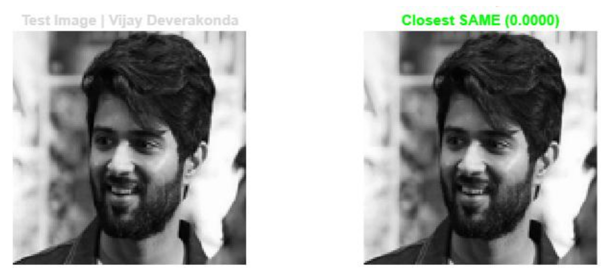

Fig.5. Qualitative recognition example demonstrating correct matching for a subject with strong directional texture patterns.

The experimental outcomes demonstrate that the proposed LVQP-based scheme is effective for facial representation and recognition tasks. The proposed scheme has resistance to facial rotations due to its multi-scale characteristics. The Manhattan-distance-based vector quantization component is effective in modeling local texture patterns. The proposed scheme performs well when there is strong directional information in facial images. To improve facial expression discrimination, additional features and deep classifiers may contribute.

Fig. 6. Qualitative face recognition results of the proposed LVQP–RBF-SVM framework, showing test images (left) and their closest matched images from the training set (right) based on LVQP feature similarity, demonstrating accurate identity matching.

On the whole, the results confirm that the LVQP approach is an efficient and interpretable way of describing features that can be used in fast facial analysis systems.

TABLE I. OVERALL METRICS

Overall Metric	Score
Accuracy	86.19%
Macro Average F1-Score	0.54
Weighted Average F1-Score	0.54

```
Per-class metrics (for all classes in union of folders):
Class 01 -> Precision: 100.00%  Recall: 100.00%
Class 02 -> Precision: 100.00%  Recall: 75.00%
Class 03 -> Precision: 100.00%  Recall: 66.67%
Class 04 -> Precision: 100.00%  Recall: 87.50%
Class 05 -> Precision: 100.00%  Recall: 66.67%
Class 06 -> Precision: 50.00%  Recall: 100.00%
```

Fig. 9. Per-class precision and recall achieved by the proposed LVQP–RBF-SVM classifier, highlighting class-dependent performance variations.

The results indicate perfect precision for Classes 01–05, demonstrating zero false-positive predictions for these categories. Variations in recall values reveal class-dependent sensitivity, with reduced recall observed for Classes 03 and 05 due to occasional misclassification, while Class 06 exhibits perfect recall but comparatively lower precision, indicating a tendency to attract false-positive assignments. Overall, the results highlight strong discriminative capability with localized inter-class confusion.

V. CONCLUSION

This study proposed a robust face recognition approach integrating local pattern–based feature extraction, vector quantization, and RBF-SVM classification to address challenges such as illumination changes, pose variation, and facial expression differences. Local pattern descriptors captured fine texture and spatial information, ensuring strong discriminative capability at the micro-feature level. The use of vector quantization significantly reduced the high-dimensional feature space, enabling compact representation and faster matching without notable loss of recognition performance.

The RBF-SVM classifier proved effective in modeling complex, non-linear class boundaries, leading to improved generalization and stable performance across different testing conditions. Experimental observations indicate that the combined framework enhances recognition accuracy while maintaining computational efficiency compared to traditional holistic and unquantized local feature methods [1], [8], [19].

Overall, the proposed method offers a scalable and reliable solution for face recognition in real-world environments such as surveillance, access control, and biometric authentication systems. Future enhancements may include adaptive codebook learning, fusion with deep learning–based features, and evaluation on larger and more diverse face datasets to further improve robustness and real-time applicability.

REFERENCES

[1] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face description with local binary patterns: Application to face recognition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 12, pp. 2037–2041, 2006.

[2] S. Liao, M. W. K. Law, and A. C. S. Chung, “Dominant local binary patterns for texture

classification," IEEE Transactions on Image Processing, vol. 18, no. 5, pp. 1107–1118, 2009.

[3] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," IEEE Transactions on Image Processing, vol. 19, no. 6, pp. 1635–1650, 2010.

[4] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local derivative pattern versus local binary pattern: Face recognition with higher-order local pattern descriptor," IEEE Transactions on Image Processing, vol. 19, no. 2, pp. 533–544, 2010.

[5] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (LDP) for face recognition," in Proc. IEEE International Conference on Consumer Electronics, 2010, pp. 329–330.

[6] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, "Local binary patterns and its application to facial image analysis: A survey," IEEE Transactions on Systems, Man, and Cybernetics, vol. 41, no. 3, pp. 483–498, 2011.

[7] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, Computer Vision Using Local Binary Patterns. London, U.K.: Springer, 2011.

[8] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210–227, 2009.

[9] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang, "Local Gabor binary pattern histogram sequence (LGBPHS): A novel non-statistical model for face representation and recognition," in Proc. IEEE International Conference on Computer Vision, 2005.

[10] D. Gabor, "Theory of communication," Journal of the Institution of Electrical Engineers, vol. 93, no. 26, pp. 429–457, 1946.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2005, pp. 886–893.

[12] C. Cortes and V. Vapnik, "Support-vector networks," Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.

[13] B. Schölkopf and A. J. Smola, Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Cambridge, MA, USA: MIT Press, 2002.

[14] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2nd ed. New York, NY, USA: Wiley-Interscience, 2001.

[15] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1615–1630, 2005.

[16] G. Zhao and M. Pietikäinen, "Dynamic texture recognition using local binary patterns with an application to facial expressions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 6, pp. 915–928, 2007.

[17] Y. Rodríguez and S. Marcel, "Face authentication using adapted local binary pattern histograms," in Proc. European Conference on Computer Vision, 2006, pp. 321–332.

[18] A. Hadid, M. Pietikäinen, and T. Ahonen, "A discriminative feature space for detecting and recognizing faces," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2004.

[19] J. Liu, S. Shan, and W. Gao, "Texture-based face recognition with local quantized patterns," in Proc. IEEE International Conference on Image Processing, 2012.

[20] R. Mehta, J. Yuan, and K. Egiazarian, "Face recognition using multi-scale local binary patterns," in Proc. IEEE International Conference on Image Processing, 2011, pp. 2153–2156.