

Full Direction Local Neighbors Pattern (FDLNP)

A Novel Method for Multimedia Image Retrieval System using Ensemble Classifiers on Distributed Framework

Maher Alrahhal¹

Ph.D. Scholar in Computer Science and Engineering Dept.
JNTUH College of Engineering, Hyderabad
Hyderabad, India

Supreethi K.P²

Professor in Computer Science and Engineering Dept.
JNTUH College of Engineering, Hyderabad
Hyderabad, India

Abstract—In this paper, we proposed the Full Direction Local Neighbor Pattern (FDLNP) algorithm, which is a novel method for Content-Based Image Retrieval. FDLNP consists of many steps, starting from generating Max and Min Quantizers followed by building two matrix types (the Eight Neighbors Euclidean Decimal Coding matrix, and Full Direction Matrixes). After that, we extracted Gray-Level Co-occurrence Matrix (GLCM) from those matrixes to derive the important features from each GLCM matrixes and finishing with merging the output of previous steps with Local Neighbor Patterns (LNP) histogram. For decreasing the feature vector length, we proposed five extension methods from FDLNP by choosing the specific direction matrixes. Our results demonstrate the effectiveness of our proposed algorithm on color and texture databases, comparing with recent works, with regard to the Precision, Recall, mean Average Precision (mAP), and Average Retrieval Rate (ARR). For enhancing the image retrieval accuracy, we proposed a novel framework that combined the image retrieval system with clustering and classification algorithms. Moreover, we proposed a distributed model that used our FDLNP method with Hadoop to get the ability to process a huge number of images in a reasonable time.

Keywords—Content-Based image retrieval; full direction local neighbor patterns; local neighbor pattern; gray-level co-occurrence matrix; ensemble classifiers; k-means clustering; hadoop

I. INTRODUCTION

In the latest decade, the number of digital photos and videos accessible has increased dramatically. Although software and hardware are available to digitize, archive, and compress multimedia data, there are no clear ways to retrieve the kept info. In traditional Text-Based Image Retrieval (TBIR), metadata that describes the image contents are manually added to image files. This metadata is used to retrieve similar images by word matching. TBIR has two major difficulties: (a) labeling the images manually, therefore, a significant volume of time is required, and (b) human sensitivity for images is not precise and unique. Content-based image retrieval (CBIR) [1-5] is an alternative to traditional TBIR that overcomes the above limitations because images are retrieved based on their content such as color, texture, shapes, and contour.

CBIR is a very vital area, specifically in the latest era due to the increasing essentials to retrieve images from a multimedia datasets. It intersects with many areas such as

image segmentation, machine learning for classification and clustering, big data for processing huge databases, and deep learning for extract semantic features. For that, CBIR has several applications in different areas. For example, CBIR systems are utilized in satellite imagery to discover ground natural resources, aerial surveys, monitor agriculture, and create climate reports and path surface objects. Medical imaging is a prominent area of CBIR application, which utilizes to monitor patient health reports, to assist in diagnosis by detecting related previous studies [6], etc. Fingerprint matching images that lead to a distinct verification is another domain that can use in the banking area, universities, business companies, and scientific laboratories [7-9].

A. Problem Statement

Image search engines are very important tools for searching similar images from huge databases, which depend on the CBIR field to automatically retrieve the best matching images by extracting the information from the query image. However, many studies in the CBIR focus on the low feature extraction from the images. Therefore, there is still a semantic gap between the low-features extracted using those algorithms and how the human can understand the images. This is the first problem of the CBIR domain, whereas the second main problem is retrieving similar images from a huge database in a reasonable time.

B. Contribution of our Work

To address the above problems, we proposed a new framework that combined image processing, machine learning, and big data, as the following:

First, we reduced the semantic gap between the low-level and high-level feature extraction by proposing a novel and efficient method for feature extraction, namely Full-Directions Local Neighbors Pattern that focused on the color changes of minimum and maximum image blocks, fused with the texture features that extract the relationships of the center pixel with its neighbors.

Second, to get the ability to process a huge number of images, we reduced the size of FDLNP's feature vector by proposing extended versions of our proposed method.

Third, we studied the importance of using the machine learning algorithms to enhance the effectiveness of the system,

by using our proposed FDLNP for training the classifier and retrieving the top k images from the classifier's output class.

Forth, we applied our proposed method in the Map Reduce framework to get the ability to process a large image database.

II. RELATED WORK

Low-level features are very popular in image representation and pattern recognition technologies that also proved powerful in image classification and retrieval [10-13]. Many methods discussed the using of the color feature in CBIR, some of them are color histogram, color moments [14], color correlogram [15], and color co-occurrence matrix [16].

Xiaoyin [14] proposed a new technique for improving color features extracted from the image that is color moment invariant. The color feature was extracted for each color component (red, green, and blue) to make an accurate features vector of the image. Color co-occurrence matrix (CCM) [17] is another approach that is interested in the spatial relationship between color channels. Where the image can be considered as an appropriate composition for "elementary structures". The elements of these pixels have visible properties of colors and relationships between the colors.

Jhanwar et al. [18] have employed another approach in the CBIR domain, known as motif co-occurrence matrix (MCM), which is conceptually near to the CCM. Where the image is subdivided into blocks, and each block replaced by a scan motif. Guoping [19] has proposed block truncation coding (BTC) which is a type of image compression method. The authors used the features extracted using BTC for retrieving the images. The main idea of BTC is to build two types of features used to retrieve both texture and color images. Those features are block color co-occurrence matrix and block pattern histogram. Many researchers used this idea to propose another BCT variants [20, 21]. The dominant color descriptor (DCD) [22] has proposed as one application of MPEG-7 to use it for extracting color features from the image with important merits that are small feature vector size with good performance.

The texture is another important feature in the CBIR domain that is interested in extracting the spatial arrangement from an image. Many researchers worked on the texture features for enhancing the image retrieval area, some of them are: Gabor filters, Wavelet transforms, gray-level co-occurrence matrix (GLCM) [23], Markov random field (MRF) [24], edge histogram descriptor (EHD) [25], steerable pyramid decomposition (SPD) [26], and Tamura features [27].

Some researchers have turned to merge feature types to make a robust descriptor of the image as Xingyuan and Zongyu [28] whose combined color and texture features in their proposed method namely a structure element descriptor (SED). Liu et al. [29] proposed a micro-structure descriptor (MSD) that extracts edge orientation and fused with color features to represent the image, whereas Chatzichristofis et al. [30] represented the image by fuse the fuzzy color and texture histograms. His method generally is suitable for dealing with large databases due to the small size of image representation. Lai et al. [31] proposed a framework that combined the features extracted from the image and genetic algorithm for

getting the robust features that rely on the fusion of both color and texture features.

Shape feature is another important descriptor of the images that is used in many fields like image segmentation, object detection, and image retrieval. Many researchers have a focus on extracting the shape descriptors that characterize rotation, translation, and scaling invariant. Some proposed methods proved an accurate shape descriptor such as shape context SIFT, HOG, and SURF descriptors [32-35]. The SIFT and SURF descriptors rely on extracting interest points from the image and extract the features from those points. And because of the large size of those descriptors, many researchers proposed methods for reducing the size of those features by using a bag of visual word techniques that used in many vision applications [36-39].

In recent years, deep learning domain has been proved a very efficient way to use in vision tasks like object detection, image recognition, image segmentation, and image analysis [40-47]. Convolutional neural networks (CNNs) had used in image retrieval to extract semantic features by training the network and using it to extract features that consider somehow high-level features in image retrieval [40, 43-45].

III. METHODOLOGY

The general architecture of CBIR system consists of several steps as shown in Fig.1:

Query Request: the user inputs an image as a query image.
Feature Extraction: extracts the features from the query image using our proposed method FDLNP.

Similarity Measures: measure the similarity between the feature vector of the query image and all feature vectors in the feature database; which made in the offline phase of the CBIR system by repeating the feature extraction step for all the image in the database, using this equation (1) [48]:

$$D_L(Q, DB) = \sum_{i=1}^L \left| \frac{f_{DB_{ji}} - f_{Q_i}}{1 + f_{DB_{ji}} + f_{Q_i}} \right| \quad (1)$$

Where $f_{DB_{ji}}$ is the i^{th} feature of the j^{th} image in the database, and f_{Q_i} is the feature vector of the query image.

Sorting and Retrieving: Sort the distance in a rising way and retrieve the top k images as the output of the system.

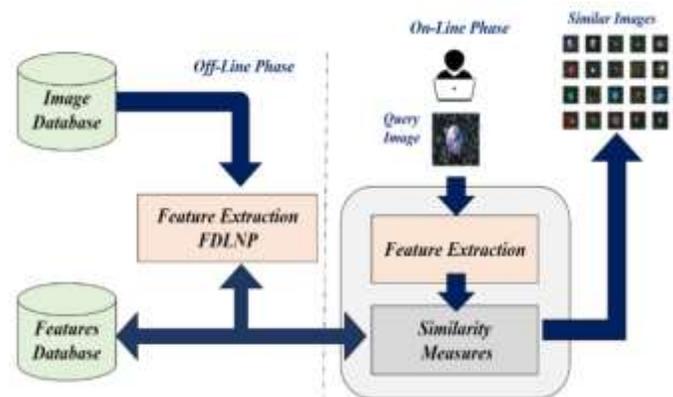


Fig. 1. The Architecture of CBIR System.

The feature extraction step considers the most important step in the whole system because the accuracy and effectiveness of the CBIR system depend mainly in this step. A novel and efficient visual feature descriptor, namely, Full Direction Local Neighbor Pattern is proposed for image representation and retrieval as shown in Fig.2, which consists of the following main steps:

Step 1: Min and Max Quantizers: Suppose a color image of size $M \times N$ is partitioned into multiple non-overlapping image blocks of size $m \times n$. Let $f(x, y) = \{f_R(x, y), f_G(x, y), f_B(x, y)\}$ be an image block, where $x = 1, 2 \dots m$ and $y = 1, 2 \dots n$. The minimum and maximum value of the image block for each RGB color space are computed as equation (2), (3):

$$q_{min}(x, y) = \{\min_{v,x,y} f_R(x, y), \min_{v,x,y} f_G(x, y), \min_{v,x,y} f_B(x, y)\} \quad (2)$$

$$q_{max}(x, y) = \{\max_{v,x,y} f_R(x, y), \max_{v,x,y} f_G(x, y), \max_{v,x,y} f_B(x, y)\} \quad (3)$$

Step 2: Right Neighbor Euclidean Coding (RNEC): For each pixel in min and max quantizers, we calculated the RNEC matrix using the right neighbors (RN_1, RN_2, RN_3) of the center pixel (cp) as shown in Fig.3, using this equation (4):

$$RNEC(cp) = \sqrt{(RN_1 - cp)^2 + (RN_2 - cp)^2 + (RN_3 - cp)^2} \quad (4)$$

Step 3: Eight Neighbors Euclidean Decimal Coding: Calculated Eight Neighbors Euclidean Decimal Coding (ENEDC) matrix using the equations (5), (6):

$$ENEDC = \sum_{p=1}^8 2^{(p-1)} \times f_1(g_p - g_c) \quad (5)$$

$$f_1(x) = \begin{cases} 1 & x \geq 0 \\ 0 & otherwise \end{cases} \quad (6)$$

Where g_c is the RNEC value of the center pixel, g_p is the RNEC value of its neighbors, and p is the number of neighbors.

Step 4: Directions Matrix: Calculated the direction of each pixel and stored it in dir_Matrix , by using the right neighbor pattern as illustrated in Fig.1. The direction of the center pixel defined as the direction from the smallest element in the pattern to the largest element in the pattern. In the case of the existing more than one minimum (maximum) element, we take the first minimum (maximum) element as the smallest (largest) value. Moreover, if the pattern has the same value, then we consider the direction of the pattern is "one". We have eight directions derived from right neighbor pattern as shown in Fig.4.

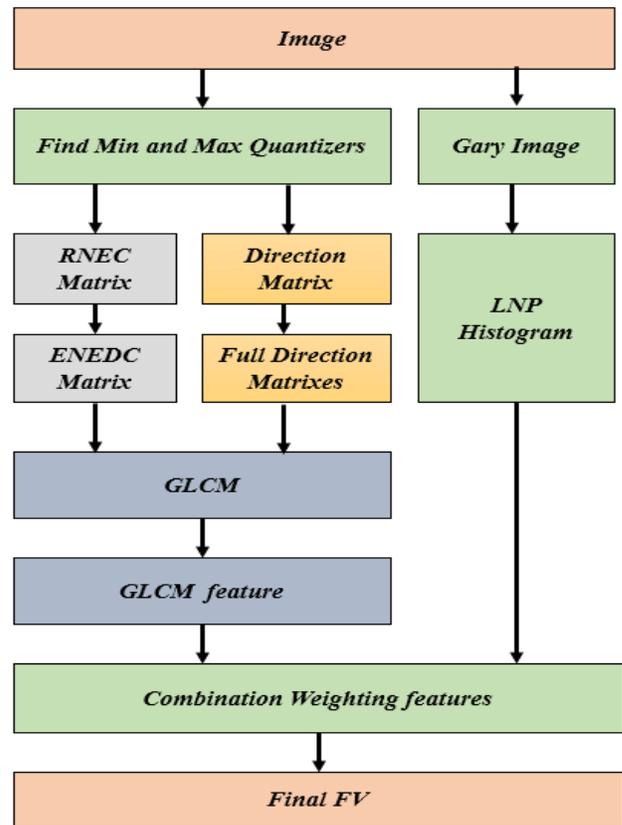


Fig. 2. Flow Chart of our Proposed Feature Extraction Method (FDLNP).

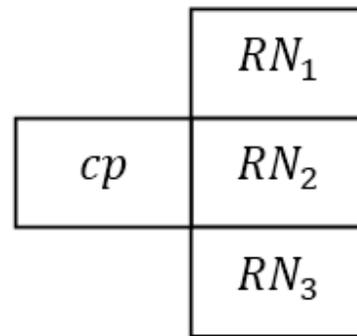


Fig. 3. Right Neighbor Pattern.

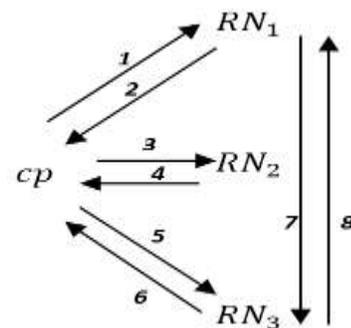


Fig. 4. Eight Directions in Neighbor Pattern.

Step 5: Full Directions Matrixes

From *dir_matrix*, derive *Full Directions Matrixes*, which is a set of 64 matrixes corresponding to the pair (*dir_value*, *Nbr_value*) where:

dir_value is the direction value of the center pixel of 3 × 3, and its value ranges from one to eight.

Nbr_value is the direction value of the 3 × 3 neighbors around *dir_value*, and their values range from 1 to 8.

To illustrate the calculation of the *Full Directions Matrixes*, we use this example like the following:

Suppose we want to derive the matrix corresponding to *dir_value* = 5, and *Nbr_value* = 2.

First, we found each pixel in *dir_value* = 5. Second, look at *Nbr_value* = 2 in the eight neighbors around the center pixel, and if it is found, replace its location by one, otherwise zero. Third, replaced the center pixel by the decimal value of the eight neighbors starting from the top-left pixel, using these equations (7), (8):

$$decimal_Value = \sum_{p=1}^8 2^{p-1} f_{dir_value}(i) \tag{7}$$

$$f_{dir_value}(i) = \begin{cases} 1 & ; i = Nbr_value \\ 0 & \text{Otherwise} \end{cases} \tag{8}$$

Where the value of *f_{dir_value}* determines if the value of the neighbor pixels equals *Nbr_value* or not.

At the end of the fifth step, we have 65 matrixes (1 for *ENEDC*, and 64 matrixes for Full Directions Matrixes) for each Min quantizer and Max quantizer. Fig.5 is an illustrating example of *step 2 to step 5* in our proposed method.

Step 6: Gray-Level Co-occurrence Matrix (GLCM): This step is for deriving the GLCM matrix for each of the 65 matrixes, where we calculated the GLCM [49] matrix with distance *d* = 1, and direction $\theta = 0^\circ$. From each GLCM matrix, we extracted 10 feature values [49] as follows: Contrast, Cluster prominence, Entropy, Sum of Square Variance, Sum Average, Sum Entropy, Difference Variance, Difference Entropy, Information Measure of Correlation, and Inverse Difference Moment Normalized. Finally, we combined those values to get the first feature vector *FV₁* whose size is 1300.

Step 7: Local Neighbor Pattern (LNP): This step starts with converting the query image to a grayscale image and then finds the Local Neighbor Pattern (LNP) histogram [48] to get the second feature vector *FV₂*, whose size is 256. Fig.6 illustrates the mechanism of building the LNP matrix.

Step 8: Fusion Features: For getting the final feature vector of our proposed method, we fused *FV₁* with *FV₂* using the equation (9) as follows:

$$FV = \alpha \times FV_1 + \beta \times FV_2 \tag{9}$$

Knowing that, we used $\alpha = 1$, and $\beta = 0.1$, and the size of the final feature vector is $1300 + 256 = 1556$, where “+” represents the concatenating operator.

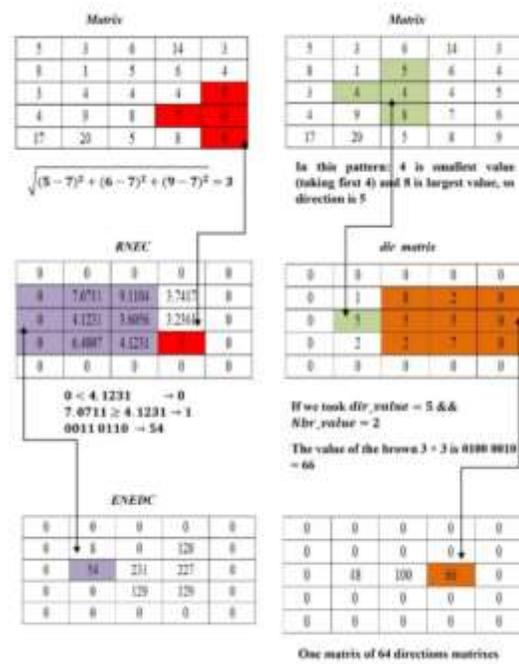


Fig. 5. Example of Calculating RNEC, ENEDC, Direction Matrix, and One Matrix of 64 Directions Matrixes.

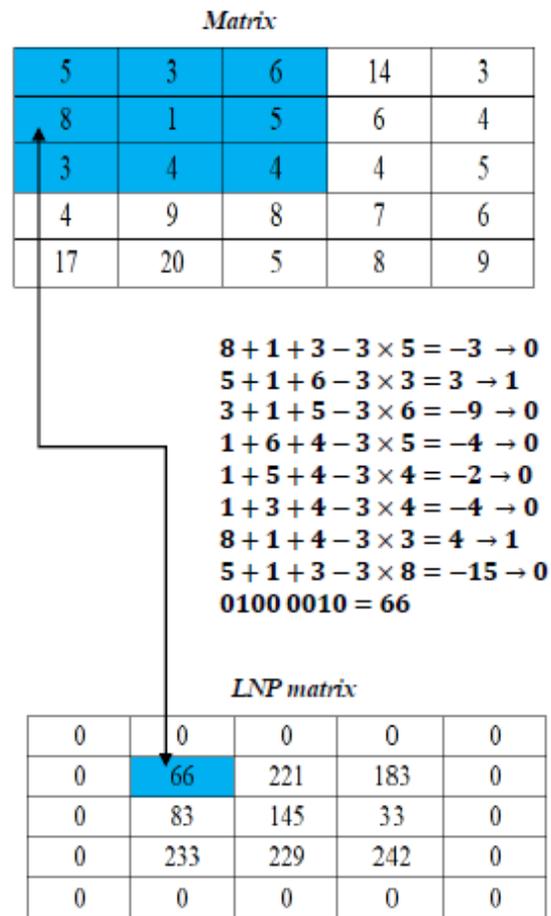


Fig. 6. Example of Calculating Local Neighbors Pattern Matrix.

IV. PERFORMANCE MEASUREMENT

Precision and Recall [50] consider the main performance measures in the image retrieval system, which defined as equations (10-12):

$$P(I_q, n) = \frac{1}{n} \sum_{i=1}^{|DB|} \varphi(\gamma(I_i), \gamma(I_q)) ; Rank(I_i, I_q) \leq n \quad (10)$$

$$\varphi(x, y) = \begin{cases} 1 & ; x = y \\ 0 & \text{else} \end{cases} \quad (11)$$

Where n is the number of retrieved images, $|DB|$ is the size of the database, $\gamma(x)$ is the class of the image x , and $Rank(I_i, I_q)$ returns the rank of the image I_i for the query image I_q among all images in the database.

$$R(I_q, n) = \frac{1}{N_c} \sum_{i=1}^{|DB|} \varphi(\gamma(I_i), \gamma(I_q)) ; Rank(I_i, I_q) \leq n \quad (12)$$

Where N_c indicated the max number of images in the database's class.

Average Retrieval rate (ARR) and the mean Average Precision (mAP) are another performance measures [50], which mathematically reflects the efficiency of CBIR. ARR and mAP defined as equations (13), (14):

$$mAP = \frac{1}{|DB|} \sum_{i=1}^{|DB|} P(I_i, n) \quad (13)$$

$$ARR = \frac{1}{|DB|} \sum_{i=1}^{|DB|} R(I_i, n) ; n \geq N_c \quad (14)$$

A higher value of the previous measures reflects a better efficient retrieving system.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. First Experiment: Image Retrieval using FDLNP

We used two types of databases for evaluating our proposed method that are color databases (Corel 1k and Corel 10k) and Texture databases (Brodatz and Vistex).

Corel 1k database [51] considers as standard database for color image retrieval that contains ten different categories, and each category consists of hundred color images. Some of those categories are African, Beach, etc.

Corel 10 k database [52] used for evaluating the methods that are interested in color features extraction. Corel 10k consists of hundred categories, and each category contains a hundred images. Therefore, it is one of the important large retrieval databases. The images in the same category contain similar semantic information. Some of the categories included in Corel 10 k are flowers, birds, beaches, buildings, etc.

Brodatz texture database [53, 54] used mainly for evaluating the methods of texture image retrieval. Brodatz database consists of 116 grayscale images, 109 images gathered Scale images from the USC database. The dimension of every image is 512×512 , which separated into 16 non-overlapping images of size 128×128 . Thus, the total number of images in this database are 1856.

Vistex database [55] contains 40 images. The dimension of every image is 512×512 , which separated into 16 non-overlapping images of size 128×128 . Thus, the total number of images in this database are 640. Vistex database used for evaluating the methods of color texture retrieval.

For color databases, we are interested in Precision (top 10 images), Recall (top k; where k is the maximum number of images in the database's class, i.e. k=100), and mAP (mean Average Precision). The mAP measure calculated as follows, for each image in the database, we consider top 10, top 20... top 100, then take the average, and repeats this procedure for all images in the database.

For texture databases, we are interested in Precision (top 10 images), Recall (top k; k=16), mAP (mean Average Precision) by taking the average of top 4, top 8, top 10, top 12, and top 16 for all images in the database, and Average retrieval rate (ARR). The ARR calculated by taking the average of top 16, top 32, top 48, top 64, top 80, top 96, and top 112 for all images in the database.

We compared our proposed method with the following methods:

1) Methods that used specifically for texture image retrieval like LDP [56], LTrP [50], LOtP [57], LHDP [57], and LNP [48]. Those methods work on the pixels' values directly and encrypt the relationship between the center pixel and its neighbors.

2) Histogram of oriented gradient method [34], which is interested in the shape of the object, by calculating the gradient orientation for each block in the images, and used mainly in object recognition.

3) Bag of Visual Words method [36], which relies on detecting the interest points and extracts the features from these points.

4) EDBTC [58] and MIFM [59] methods, which depend on extracting both texture and color features. Our proposed method gave better performance than previous methods regarding Precision, Recall, mAP, and ARR for all databases, because it works on extracting the color changes of minimum and maximum intensities of images blocks, in addition to the texture features of the images.

We used Desktop with Intel Core i7 CPU, 8 GB RAM, and Matlab R2020a program to produce those results. The results of the method compared to other methods for both color and texture databases illustrated in table I.

Fig.7 illustrates examples of retrieving similar images using our proposed method, for all the databases, where the top left images are the query images. In the color databases (Corel 1k and Corel 10k), we consider top 20 images as the output, whereas top 16 images as output for texture databases (Brodatz and Vistex). We observe that the precision of those examples are 100%.

TABLE I. COMPARISONS BETWEEN OUR PROPOSED METHOD WITH OTHER METHODS REGARDING PRECISION, RECALL, MAP, AND ARR FOR ALL DATABASES.

Corel 1k	Methods									
Performance Measure	LDP	LTrP	LOtP	LHdP	LNP	HOG	BOW	EDBTC	MIFM	FDLNP
Precision	55.10	57.20	60.80	62.40	61.30	61.24	62.43	79.70	71.20	83.20
Recall	31.20	34.90	36.30	37.42	36.44	37.42	38.45	59.20	57.24	62.80
mAP	42.40	43.60	44.80	47.10	45.20	46.25	47.61	66.50	61.20	73.50
Corel 10k	Methods									
Performance Measure	LDP	LTrP	LOtP	LHdP	LNP	HOG	BOW	EDBTC	MIFM	FDLNP
Precision	31.89	33.41	36.71	37.21	36.39	39.12	43.04	68.24	59.60	74.11
Recall	11.86	12.17	13.22	13.98	13.56	16.20	18.25	45.42	40.24	50.31
mAP	17.11	18.25	19.87	21.21	20.31	22.46	26.16	56.98	49.26	66.27
Brodatz	Methods									
Performance Measure	LDP	LTrP	LOtP	LHdP	LNP	HOG	BOW	EDBTC	MIFM	FDLNP
Precision	81.8	82.8	84.5	86.0	85.8	82.7	83.1	85.6	86.1	88.4
Recall	73.5	75.2	76.9	78.4	77.5	74.4	75.1	80.8	81.4	84.2
mAP	82.8	84.6	85.4	86.7	86.5	83.8	84.4	88.8	89.6	90.9
ARR	86.8	87.8	88.8	89.7	88.8	86.1	86.6	90.2	91.2	93.2
Vistex	Methods									
Performance Measure	LDP	LTrP	LOtP	LHdP	LNP	HOG	BOW	EDBTC	MIFM	FDLNP
Precision	87.9	88.7	90.8	91.8	92.6	89.0	89.4	93.1	93.4	95.8
Recall	77.3	78.8	80.3	81.8	82.5	80.2	81.3	84.7	85.0	87.1
mAP	88.3	89.5	90.7	91.9	92.7	90.2	91.6	94.5	94.7	97.2
ARR	90.6	90.9	91.3	92.1	92.5	91.4	92.7	95.9	96.3	98.8



Fig. 7. CBIR Examples for all Databases using our Proposed Method.

B. Second Experiment: Feature Selection Method

Our proposed method extracts the visual information from the images, and gives an efficient performance for the standard databases. However, the feature vector length considers a bit high. Our aim is to reduce the feature vector dimensions and therefore reducing the complexity of FDLNP. For that, we proposed extended versions of FDLNP by generating partial direction matrixes instead of full direction matrixes via selecting specific directions for generating GLCM and those directions are shown in Fig.8. The extended versions of FDLNP are:

1) *Top Right Direction Local Neighbors Pattern (TRDLNP)*: we take the upper main diagonal directions of *dir_matrix* to get 36 matrixes for each quantizer instead of 64 matrixes. Therefore, the size of feature vector $2 \times (10 + (36 \times 10)) + 256 = 996$.

2) *Down Left Direction Local Neighbors Pattern (DLDLNP)*: by taking lower main diagonal directions of

dir_matrix to get 36 matrixes instead of 64 matrixes and the size of feature vector is $2 \times (10 + (36 \times 10)) + 256 = 996$.

3) *Odd Direction Local Neighbors Pattern (ODLNP)*: by taking odd directions numbers from *dir_matrix* to get 32 matrixes instead of 64 matrixes and the size of feature vector calculates as follows: $2 \times (10 + (32 \times 10)) + 256 = 916$.

4) *Even Direction Local Neighbors Pattern (EDLNP)*: by taking even directions number from *dir_matrix* to get 32 matrixes instead of 64 matrixes and the size of feature vector is $2 \times (10 + (32 \times 10)) + 256 = 916$.

5) *Diagonal Direction Local Neighbors Pattern (DDLNP)*: both main and secondary diagonal from *dir_matrix* was taken to get 16 matrixes instead of 64 matrixes and the size of feature vector calculates as follows: $2 \times (10 + (16 \times 10)) + 256 = 596$.

In this experiment, we used the same four databases and mAP and ARR performance measures. We perceive, as shown in table II, that all extended versions of FDLNP gave very good accuracy closed to FDLNP and all the result better than comparing methods.

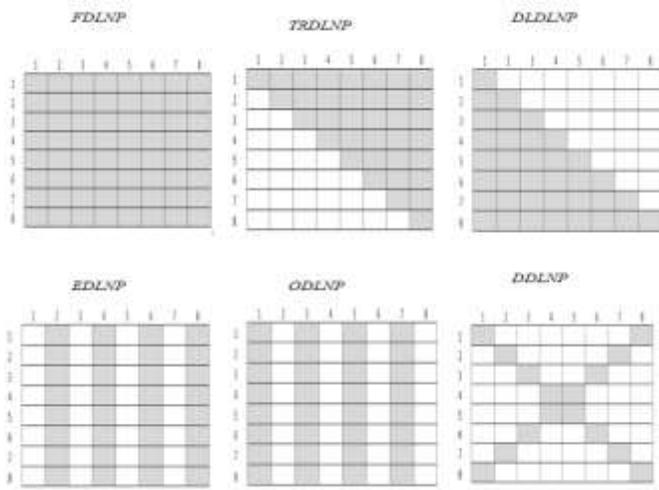


Fig. 8. Illustration Scheme for the Directions of Extended Versions of FDLNP.

TABLE II. COMPARISON BETWEEN EXTENDED VERSIONS OF FDLNP REGARDING MAP AND ARR FOR COREL 1k, COREL 10k, BRODATZ, AND VISTEX DATA

Methods	Databases					
	Vistex		Brodatz		Corel 1k	Corel 10k
	mAP	ARR	mAP	ARR	mAP	mAP
FDLNP	97.2	98.8	90.9	93.2	73.5	66.27
TRDLNP	96.8	98.0	89.1	92.6	73.1	65.9
TLDLNP	96.8	98.2	89.0	92.5	73.2	66.0
EDLNP	97.0	97.6	98.2	91.8	72.6	65.1
ODLNP	96.8	97.7	98.3	91.9	72.4	65.3
DDLNP	96.6	97.2	97.6	91.0	72.0	64.5

C. Third Experiment: Image Retrieval using Machine Learning Classification

In this experiment, we studied the role of machine learning algorithms in the image retrieval system, and its ability for enhancing the retrieval accuracy. We can use the classification algorithms in the CBIR system in two ways:

First, supporting image retrieval outputs, where both classification and content-based image retrieval share in the feature extraction step. Then the retrieval system uses those features for measuring the similarity, whereas in the classification algorithms, we feed those features to the classifier to recognize the class of the query image. In this way, the classification task only provides the class of the query image, alongside the retrieval outputs.

Second, enhancing the image retrieval outputs as following:

Step 1- Extract the features (using the proposed method) from training images to train the classifier.

Step 2- Predict the class of each image using the trainer classifier.

Step 3- When the user inputs the query, the system will extract the feature vector from this image, and then feed it to the classifier, which in its turn predict the class of the query image, let us say *SI*.

Step 4- Measure the similarity distances between the features of the query image and features of all images whose prediction; in the step2, is *SI*, to retrieve top k images.

In this experiment, we used the second way that has the following advantages, enhancing the accuracy of the CBIR as we proved in our previous work [48], reducing the number of comparisons, and make it confined in a specific class. Finally, the output of the classification-based image retrieval system always retrieves similar images, unless the classifier misclassifies the query image, which leads to output completely wrong results, and this is the main disadvantage of this way. For solving this disadvantage, we introduced the *K-Means Based Decision Maker* as shown in Fig.9, which aims to check if the classifier correctly classifies the query image, then it forwards the system to use the classification-based image retrieval, otherwise use the similarity-based image retrieval.

The algorithm of the K means based decision maker summarized with the following steps as shown in Fig. 10:

- Cluster the features database to K clusters using K-Means algorithm [60], where K refers to the number of classes in the database.
- Add label to each cluster's centroid.
- Measure the distances between the features vector of the query image and centroid of each cluster.
- Sort these distances ascendingly.
- If the class of the minimum distance matches with the classifier's label, then use classification-based image retrieval, otherwise use similarity-based image retrieval.

In our work, we use our proposed method FDLNP with different ensemble classification methods, for all databases. The description of used classifiers illustrated in table III.

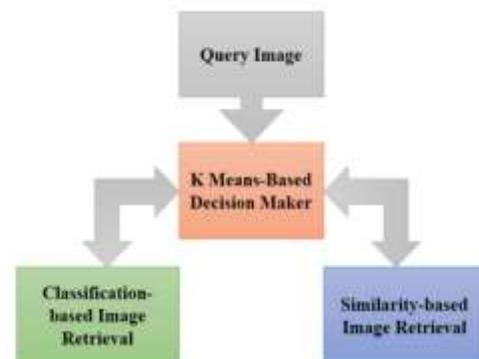


Fig. 9. Illustration Scheme of Decision Maker in Image Retrieval System.

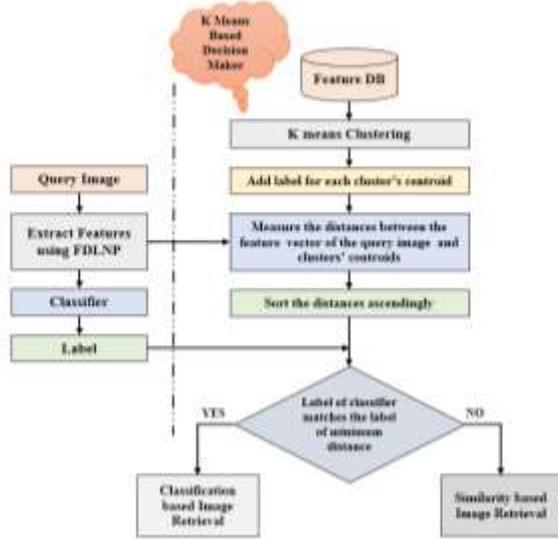


Fig. 10. The Scenario Scheme of the K-Means based Decision Maker in the Retrieval System.

TABLE III. ENSEMBLE CLASSIFIER DETAILS USED IN OUR STUDY

Model Name	Ensemble Method	Learner Type	Number of Learners
Bagged Trees [61]	Bag	Decision Tree	30
Booted Tree [62]	Adaboost	Decision Tree	30
Subspace Discriminant [63]	Subspace	Discriminant	30
Subspace KNN [64]	Subspace	Nearest Neighbors	30

In this experiment, we used our proposed method to train the classifiers with and without PCA (Principal component Analysis) [65], and compared between both methods. For training and testing the classifiers, we used k fold cross-validation [66], and in our work k=10. The results of using FDLNP with ensemble classifiers summarized in table IV.

As we observe from the table IV, FDLNP with Ensemble Bagged Trees classifiers gave the best accuracies for all databases, and we can observe that FDLNP with PCA gave also very good results however, those results are less than using the complete feature vector. The reason for that is being the PCA reduces the feature vector dimension by preserve 95% information from the complete vector.

Using classification based image retrieval system improved the accuracy of the system from (62.80, 50.31, 84.2, and 87.1) to (97.9, 93.2, 99.3, and 99.7) for Core11k, Core10k, Brodatz, and Vistex databases respectively. Moreover, we notice that the enhancements are greater in color databases than texture databases.

D. Fourth Experiment: Image Retrieval using Hadoop Distributed Framework

Hadoop [67-69] has two major layers namely: Processing Computation (Map Reduce), and Storage layer (Hadoop Distributed File System). Our proposed Hadoop framework consists of two phases as shown in Fig.11, off line phase for building the feature database, and online phase for retrieval task.

TABLE IV. CLASSIFICATION ACCURACIES OF ENSEMBLE CLASSIFIERS WITH OUR PROPOSED METHOD FOR BOTH COLOR AND TEXTURE DATABASES

Databases	Classifier	Accuracy without PCA	Accuracy with PCA
Core11K	Bagged Trees	97.9	97.2
	Booted Tree	95.9	93.5
	Subspace Discriminant	97.7	96.7
	Subspace KNN	97.6	96.1
Core10K	Bagged Trees	93.2	92.5
	Booted Tree	91.4	90.8
	Subspace Discriminant	92.8	92.6
	Subspace KNN	92.6	92.1
Brodatz	Bagged Trees	99.3	98.8
	Booted Tree	98.0	97.7
	Subspace Discriminant	99.0	98.6
	Subspace KNN	98.7	98.3
Vistex	Bagged Trees	99.7	99.4
	Booted Tree	98.2	97.4
	Subspace Discriminant	99.4	99.1
	Subspace KNN	99.0	98.6

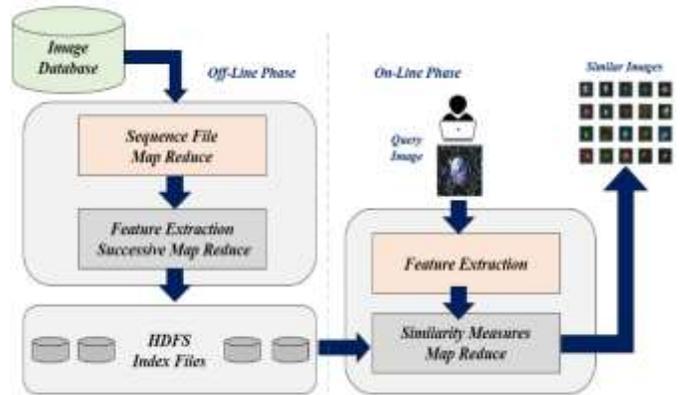


Fig. 11. The Architecture of our Proposed Hadoop Framework.

The aim of the offline phase is converting the image database; that loaded to HDFS, to the features database that stored in the HDFS, using the following steps:

1) *Sequence file map reduce*: The is the first Map Reduce job for converting a large number of small image files into a small number of large files, each containing a number of key-value pairs where the key is an image number and the value is the image data. The size of each large file is 64 MB.

2) *Successive map reduce jobs*: The aim of this step is to distribute our FDLNP method for extracting the features from sequence files by executing successive map-reduce jobs, and building the features database. The function of each job illustrated in the Fig.12. The output of each map reduce job is the input of the next one, and the output of the last job is the feature database in form of <image number, feature vector> that stored in the HDFS.



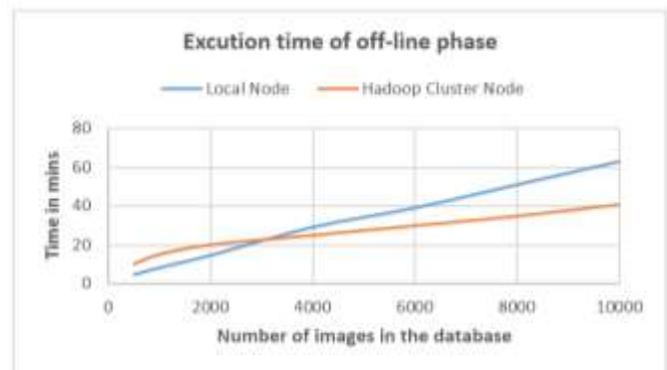
Fig. 12. Our Proposed Method in mAP Reduce Model.

In this experiment, we are interested in measuring the execution time of offline and online phases, on the Hadoop cluster that contains one master node and two slave nodes.

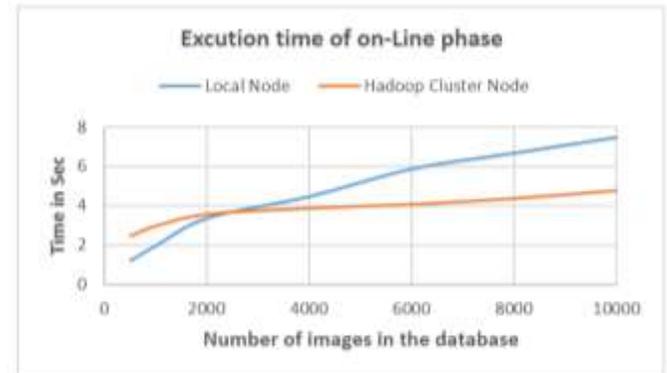
We used partial Corel 10 k database, by taking 5 images from each class, then 10, 20 and so on, for measuring the time of executing the proposed system in the cluster comparing to the time of executing the proposed system in local node.

The Fig.13-a, illustrated the comparisons of executing the proposed method on local and cluster nodes in the offline phase. We noticed that the time of proposed method in the local node is less than cluster for a small number of images (until 3000 images) because the Hadoop starts with a preprocessing step for generating Sequence File. Whereas the Hadoop cluster spends less time for processing a large number of images comparing with the local node.

In the online phase, the aim of the cluster is to calculate the similarity distance, whereas extracting features from the query image is done by the server node. The Fig.13-b, illustrated the time comparison between local and cluster nodes in similarity measuring. As in the offline phase, the local node is better than the Hadoop for a small number of images, whereas the need of the Hadoop is in processing a large number of images for reducing the execution time as much as possible. The online phase connects directly with the user who inputs the query image and waits for similar retrieving images, so we are much interested in reducing the processing time in the online phase, and this is the reason for using map-reduce in measuring the similarity distances.



(a)



(b)

Fig. 13. Comparison of the Execution Time between Local and Cluster Nodes for the Offline and Online Phases in CBIR System.

VI. CONCLUSIONS

In this work, we proposed a novel method for retrieving color and texture images, known as FDLNP that starts with generating the min and max Quantizer, followed by Eight Neighbors Euclidean Decimal Coding matrix and Full Directions Matrixes. After that, we extract GLCM matrix from previous matrixes to derive the features from each GLCM. Finally, we fused those features with the LNP histogram to get the final feature vector, which is a bit long feature vector. For reducing the feature vector length, we proposed five extended versions of FDLNP, by generating partial directions matrixes instead of full directions matrixes. Our proposed methods improved the performance measures in terms of Precision, Recall, mAP, and ARR comparing with some recent works. For enhancing the image retrieval accuracy, we combined our proposed method with machine learning algorithms. Moreover, we proposed a distributed framework to execute our method in map reduce model to get the ability to process a huge number of images in a reasonable time.

VII. FUTURE WORK

In the future, we can enhance our proposed methods to give efficient outcomes for special characterization images as satellite, thermal, and medical images. Moreover, we can use CBIR with the SPARK framework for decreasing the processing time and comparing the spark framework with the Hadoop framework in terms of scalability, fault tolerance, performance, and response time.

REFERENCES

- [1] Y. Rui, T. S. Huang, and S.F. Chang, "Image Retrieval: Current Techniques, Promising Directions and Open Issues," *Journal of Visual Communication and Image Representation*, vol. 10, pp. 39-62, Jan 1999.
- [2] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 1349-1380, Dec 2000.
- [3] M. Kokare, B. N. Chatterji, and P. K. Biswas, "A Survey on Current Content based Image Retrieval Methods," *IETE Journal of Research*, vol. 48, pp. 261-271, 2020.
- [4] Y. Liu, D. Zhang, G. Lu, and W.Y. Ma, "A survey of content-based image retrieval with high-level semantics," *Pattern Recognition*, vol. 40, pp. 262-282, 2007.
- [5] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval," *ACM Computing Surveys*, vol. 40, pp. 1-60, 2008.
- [6] Jose Ramos, Thessa T. J. P. Kockelkorn, Isabel Ramos, Rui Ramos, Jan Grutters, Max A. Viergever, "Content-Based Image Retrieval by Metric Learning from Radiology Reports: Application to Interstitial Lung Diseases," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, pp. 281-292, 2016.
- [7] Zegarra, Javier A. Montoya, Neucimar J. Leite, and Ricardo da Silva Torres, "Wavelet-based fingerprint image retrieval," *Journal of computational and applied mathematics*, vol. 227, pp. 294-307, 2009.
- [8] Gavrielides, Marios A., Elena Sikudova, and Ioannis Pitas, "Color-based descriptors for image fingerprinting," *IEEE Transactions on Multimedia*, vol. 8, pp. 740-748, 2006.
- [9] M. Alkhwilani, M. Elmoogy and H. El Bakry, "Text-based, Content-based, and Semantic-based Image Retrievals: A Survey," *International Journal of Computer and Information Technology*, vol. 4, pp. 58-66, 2015
- [10] M. Alrahhah and Supreethi K.P., "Multimedia Image Retrieval System by Combining CNN With Handcraft Features in Three Different Similarity Measures," *International Journal of Computer Vision and Image Processing*, vol. 10, pp. 1-23, Jan 2020.
- [11] G.-H. Liu and J.-Y. Yang, "Image retrieval based on the texton co-occurrence matrix," *Pattern Recognition*, vol. 41, pp. 3521-3527, 2008.
- [12] G.-H. Liu, L. Zhang, Y.-K. Hou, Z.-Y. Li, and J.-Y. Yang, "Image retrieval based on multi-texton histogram," *Pattern Recognition*, vol. 43, pp. 2380-2389, 2010.
- [13] J. Luo and D. Crandall, "Color object detection using spatial-color joint probability functions," *IEEE Transactions on Image Processing*, vol. 15, pp. 1443-1453, 2006.
- [14] D. Xiaoyin, "Image retrieval using color moment invariant," in: *The Seventh International Conference on Information Technology: New Generations (ITNG)*, Las Vegas, NV, pp. 200-203, 2010.
- [15] H. Jing, S.R. Kumar, M. Mitra, W.J. Zhu, R. Zabih, "Image indexing using color correlograms," in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 762-768, 1997.
- [16] G.P. Qiu, "Color image indexing using BTC," *IEEE Trans. Image Process*, vol. 12, pp. 93-101, 2003.
- [17] V. Kovalev, M. Petrou, "Multidimensional co-occurrence matrices for object recognition and matching," *Graph. Models Image Process*, vol. 58, pp. 187-197, 1996.
- [18] N. Jhanwar, S. Chaudhuri, G. Seetharaman, B. Zavidovique, "Content based image retrieval using motif co-occurrence matrix", *Image Vis. Comput*, vol. 22, pp. 1211-1220, 2004.
- [19] G. Qiu, "Color image indexing using BTC," *IEEE Trans. Image Process*, vol. 12, pp. 93-101, 2003.
- [20] J. Mathews, M.S. Nair, L. Jo, "A novel color image coding technique using improved BTC with k-means quad clustering," *Advances in Signal Processing and Intelligent Recognition Systems*, pp. 347-357, 2014.
- [21] J.M. Guo, H. Prasetyo, JH. Chen, "Content-based image retrieval using error diffusion block truncation coding features," *IEEE Trans. Circ. Syst. Video Technol*, pp. 99-105, 2014.
- [22] B.S. Manjunath, P. Salembier, T. Sikora, "Introduction to MPEG-7: Multimedia Content Description Interface," Wiley, Chichester, 2002.
- [23] R. Haralick, "Statistical and structural approaches to texture," *Proc. IEEE*, vol. 67, pp. 786-804, May 1979.
- [24] G. Cross, A. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Mach. Intell*, vol. 5, pp. 25-39, 1983.
- [25] D.K. Park, Y.S. Jeon, C.S. Won, "Efficient use of local edge histogram descriptor," *ACM workshops on Multimedia*, ACM, pp. 51-54, 2000.
- [26] E.P. Simoncelli, W.T. Freeman, "The steerable pyramid: A flexible architecture for multi-scale derivative computation," in: *International Conference on Image Processing*, vol. 3, p. 3444, 1995.
- [27] H. Tamura, S. Mori, T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transaction System, Man Cybernet*, vol. 8, pp. 460-473, 1978.
- [28] X. Wang, Z. Wang, "A novel method for image retrieval based on structure elements' descriptor," *J. Vis. Commun. Image Represent*, vol. 24, pp. 63-74, 2013.
- [29] G.H. Liu, Z.Y. Li, L. Zhang, Y. Xu, "Image retrieval based on micro-structure descriptor," *Pattern Recognition*, vol. 44, pp. 2123-2133, 2011.
- [30] S.A. Chatzichristofis, Y.S. Boutalis, "Fctch: Fuzzy color and texture histogram-a low level feature for accurate image retrieval," *9th International Workshop on Image Analysis for Multimedia Interactive Services*, pp. 191-196, 2008.
- [31] C.C. Lai, Y.C. Chen, "A user-oriented image retrieval system based on interactive genetic algorithm," *IEEE Trans. Instrument. Measur*, vol. 60, pp. 3318-3325, 2011.
- [32] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509-521, 2002.
- [33] D. G. Lowe, "Distinctive image features from scale-invariant key points," *Internatiozal Journal of Computer Vision*, vol. 60, pp. 91-110, 2004.
- [34] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 886-893, June 2005.
- [35] H. Bay, T. Tuytelaars, and L. V. Gool, "SURF: speeded up robust features," in *European Conference on Computer Vision*, vol. 1, pp. 404-417, 2006.
- [36] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 591-606, 2009.
- [37] Y. Su and F. Jurie, "Improving image classification using semantic attributes," *International Journal of Computer Vision*, vol. 100, pp. 59-77, 2012.
- [38] L. Wang, L. Zhou, C. Shen, L. Liu, and H. Liu, "A convolutional word-merging algorithm with class separability measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, pp. 417-435, 2014.
- [39] H. Lobel, R. Vidal, and A. Soto, "Learning shared, discriminative, and compact representations for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, pp. 2218-2231, 2015.
- [40] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *European Conference on Computer Vision*, 2014.

- [41] M. Tzelepi and A. Tefas, "Deep convolutional learning for content based image retrieval," *Neurocomputing*, vol. 275, pp. 2467–2478, 2018.
- [42] F. Radenovic, G. Tolias, and O. Chum, "Fine-tuning CNN image retrieval with no human annotation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, pp. 1655–1668, 2019.
- [43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the Advances in Neural Information Processing Systems*, Dec 2012.
- [44] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, June 2014.
- [45] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pp. 1556–1564, June 2015.
- [46] A. Qayyum, S. M. Anwar, M. Awais, and M. Majid, "Medical image retrieval using deep convolutional neural network," *Neurocomputing*, vol. 266, pp. 8–20, 2017.
- [47] A. Alzu'bi, A. Amira, and N. Ramzan, "Content-based image retrieval with compact deep convolutional features," *Neurocomputing*, vol. 249, pp. 95–105, 2017.
- [48] M. Alrahhah and K. Supreethi, "Content-Based Image Retrieval using Local Patterns and Supervised Machine Learning Techniques," in *Proceedings of Amity International Conference on Artificial Intelligence*, Feb 2019.
- [49] M. Islam, K. Kundu and A. Ahmed, "Texture Feature based Image Retrieval Algorithms", *International Journal of Engineering and Technical Research*, vol. 2, pp. 169-173, 2014.
- [50] S. Murala, R. P. Maheshwari, and R. Balasubramanian, "Local Tetra Patterns: A New Feature Descriptor for Content-Based Image Retrieval," *IEEE Transactions on Image Processing*, vol.21, pp. 2874-2886, 2012.
- [51] Corel 1000 image database [Online]. Available: <http://wang.ist.psu.edu/docs/related/>.
- [52] "Content Based Image Retrieval / Image Database Search Engine (SIMPLICity, WIPE, Virtual Microscope)", [Wang.ist.psu.edu](http://wang.ist.psu.edu), 2021. [Online]. Available: <http://wang.ist.psu.edu/docs/related/>.
- [53] P. Brodatz, *Textures: A Photographic Album for Artist and Designers*. New York, NY, USA: Dover, 1996.
- [54] "SIPI Image Database", [Sipi.usc.edu](http://sipi.usc.edu), 2021. [Online]. Available: <http://sipi.usc.edu/database/>.
- [55] "Index of /pub/VisTex", [Vismod.media.mit.edu](https://vismod.media.mit.edu/pub/VisTex/), 2021. [Online]. Available: <https://vismod.media.mit.edu/pub/VisTex/>.
- [56] B. Zhang, Y. Gao, S. Zhao, and J. Liu, "Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor," *IEEE Transactions on Image Processing*, vol. 19, pp. 533-544, 2010.
- [57] U. Raju, Suresh Kumar, K., Haran, P., Boppana, R. and Kumar, N. "Content-based image retrieval using local texture features in distributed environment," *International Journal of Wavelets, Multiresolution and Information Processing*, p.1941001, 2019.
- [58] Jing-Ming Guo, H. Prasetyo and Jen-Ho Chen, "Content-Based Image Retrieval Using Error Diffusion Block Truncation Coding Features," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, pp. 466-481, 2015.
- [59] K. Chu and G. Liu, "Image Retrieval Based on a Multi-Integration Features Model," *Mathematical Problems in Engineering*, pp. 1-10, 2020.
- [60] H. Shen and Z. Duan, "Application Research of Clustering Algorithm Based on K-Means in Data Mining," *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, 2020.
- [61] E. Ampomah, Z. Qin and G. Nyame, "Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement," *Information*, vol. 11, p. 332, 2020.
- [62] S. Rahman, M. Irfan, M. Raza, K. Moyeezullah Ghori, S. Yaqoob and M. Awais, "Performance Analysis of Boosting Classifiers in Recognizing Activities of Daily Living," *International Journal of Environmental Research and Public Health*, vol. 17, p. 1082, 2020.
- [63] R. K, "Satellite Image Classification based on Ensemble Subspace Discriminant method using Random Subspace Algorithm," *HELIX*, vol. 8, pp. 2714-2718, 2018.
- [64] T. Mokoaleli-Mokoteli, S. Ramsumar and H. Vadapalli, "The Efficiency Of Ensemble Classifiers In Predicting The Johannesburg Stock Exchange All-Share Index Direction," *Journal of Financial Management, Markets and Institutions*, vol. 07, p. 1950001, 2019.
- [65] S. Sehgal, H. Singh, M. Agarwal, V. Bhasker and Shantanu, "Data analysis using principal component analysis", *International Conference on Medical Imaging, m-Health and Emerging Communication Systems*, 2014.
- [66] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," *arXiv:1811.12808*, 2020.
- [67] Welcome to Apache™ Hadoop®! <http://hadoop.apache.org/>.
- [68] Garry Turkington, "Hadoop Beginner's Guide", *PACKT Publishing, Shroff Publishers and Distributors Pvt. Ltd. Navi Mumbai, India*, 2013.
- [69] Srinath Perera and Thilina Gunarathne, "Hadoop MapReduce Cookbook," *PACKT Publishing, Shroff Publishers and Distributors Pvt. Ltd. Navi Mumbai, India*, 2013.

AUTHORS' PROFILES



Maher Alrahhah was born in Syria in 1992. He received the B.E. degree in computer engineering from Electrical and Electronics Engineering College, Aleppo University, Syria, in 2014, the M.Tech. Degree in Computer Science and Engineering from the National Institute of Technology Warangal, India, in 2018, he is currently pursuing a Ph.D. degree in Computer Science and Engineering at JNTUH College of Engineering, Hyderabad, India. His major fields of interest are content-based image retrieval, biomedical imaging, deep learning, and multimedia systems.



Supreethi K.P. received the Bachelor of Engineering degree in Computer Science and Engineering, Gulbarga University, India, the M.Tech degree in Computer Science in 2004 and a Ph.D. in Computer Science and Engineering from Jawaharlal Nehru Technological University, India, in 2011. She is a professor of CSE at JNTUH College of Engineering, Hyderabad. Her research interests are databases, data mining, medical imaging, and image processing. She is a life member of CSI.