An efficient multistage CBIR based on Squared Krawtchouk-Tchebichef polynomials

Ibtihaal M. Hameed1, Sadiq H. Abdulhussain1

1 Department of Computer Engineering, University of Baghdad, Baghdad, Iraq
E-mail: i.munem1305@coeng.uobaghdad.edu.iq

Abstract. Image databases are increasing exponentially because of rapid developments in social networking and digital technologies. To search these databases, an efficient search technique is required. CBIR is considered one of these techniques. This paper presents a multistage CBIR to address the computational cost issues while reasonably preserving accuracy. In the presented work, the first stage acts as a filter that passes images to the next stage based on SKTP, which is the first time used in the CBIR domain. While in the second stage, LBP and Canny edge detectors are employed for extracting texture and shape features from the query image and images in the newly constructed database. The proposed CBIR was tested against existing algorithms on well-known database (Wang’s database), where Manhattan distance is used as a similarity metric. The improvement ratio in terms of computation time between the proposed system and existing system achieves 73.99%, which is considered a promising result.

Keywords: CBIR, Image retrieval, Squared Krawtchouk-Tchebichef polynomials, Feature extraction.

1. Introduction
An enormous amount of image databases has been created in social, educational, medical, industrial, and many other living facilities. These applications require effective search techniques. There are two types of these techniques. The first technique depends on using specific keywords to annotate images to index and search image databases and known as text-based image retrieval (TBIR) [1] and suffers from several limitations. They are: 1) subject to human perception 2) annotate large databases manually is not feasible, and 3) valid for only one language. The second technique comes to convey these limitations. In this technique, the search is done based on the visual content of the query image and not on the text (keywords). This technique is known as content-based image retrieval (CBIR) [2] .This method is characterized to be automated, but it suffers from a “semantic gap”, which is the gap between the low-level features that describe images and the high-level perception (concepts) contained in the images [3]. This gap is the cause of retrieving irrelevant images from the databases. Therefore, it was the subject for research through the last three decades [4].

Color is considered the most important low-level feature in the CBIR domain because of its invariant behavior to the rotation, translation, and changing viewpoint angles [5]. Local or global color features can be extracted based on the extraction method used. A color histogram is the most widespread global extraction method used. But it suffers from a limitation in spatial information because two different images could have the same color histogram [6]. Texture is also considered an important low-level feature, which is a pivotal feature in computer vision because of its existence in many real images. Trends nowadays is directed towards using local methods in extracting texture features because local
image features are superior to global features in terms of giving stable matching in different conditions and being invariant to rotation and scale changes [7]. Many local descriptors have experimented in the CBIR domain. One of the low-level features that describe objects in an image is the shape, which can be extracted either on the base of a region or a boundary (contour) [8][9].

One feature was used to retrieve images in early studies in the CBIR domain. But, results were unsatisfactory since images typically comprise several pictorial characteristics [10]. Feature combination or feature fusion is the process used by researchers to fuse two or more features to achieve better retrieval performance and higher accuracy. Different combinations were used in literature, such as color and texture, color and shape, shape and texture, or even color, texture, and shape. All of these combinations (fusion) are used to reduce the semantic gap and in turn, increase CBIR efficiency. Many studies were concentrated on enhancing systems accuracy, but there is a tradeoff between accuracy and computational cost. It is a hard task to design and implement a CBIR system that enhances both accuracy and running time. In this study, we concentrated on enhancing computational time while trying to preserve accuracy as high as possible in order to be possible to search large databases within a reasonable time.

The rest of this paper is organized as follows: Section 2 reviews the most recent studies on CBIR. Section 3 describes the proposed methodology. Experiments and results are discussed in Section 4. Finally, Section 5 concludes the paper and presented possible future work.

2. Related Work

In this section, the most recent CBIR algorithms based on the fusion of low-level features are discussed. Anandh et al. [11] proposed a novel CBIR system based on the fusion of low-level features (color, texture, and shape). To extract color features, Color auto-correlogram is used while wavelet transform and Gabor transform were used to extract edge and texture features respectively. Manhattan distance was used by authors as a similarity measure to evaluate the performance of the system. Although they achieved high precision value, the main drawback of the proposed system was the increased computational cost due to the fusion of multiple features. Srivastava and Khare [12] constructed a new CBIR system that depends on extracting global and local features. Their proposed system analyzes images at multiple levels, in which other levels may capture some details missed by one level. Local binary pattern (LBP) is used to extract texture features locally and Legendre moments are used to extract shape features from the texture features at multi-resolution levels. To assess the proposed work, five benchmark image datasets were used and the proposed system achieved better accuracy and recall values but increased computational cost because of the multi-resolution analysis.

Nazir et al. [13] presented a methodology for CBIR based on the extraction of color and texture. For color feature extraction, a color histogram is used in the HSV color space and for texture, discrete wavelet transform (DWT) and EHD are used because EDH is efficient in finding relevant images when used in MPEG-7 [14]. Corel dataset was used in evaluating the proposed methodology and achieved better efficiency in terms of precision and recall than other state-of-the-art techniques. Rana et al. [15] presented a CBIR methodology which is based on the integration of texture (nonparametric features) and color and shape (parametric features). Color moments and moment invariants were used to extract parametric features and ranklet transformation was used to extract nonparametric features. The constructed feature vector has a length of 247, which has a negative effect in increases the running time and is considered a major limitation of the presented algorithm. To evaluate the algorithm, five datasets were used. Pavithra and Sharmila [16] introduced a novel multistage CBIR technique. In the first stage, the color feature was extracted by using color moment by calculating the mean and standard deviation for R, G, and B channels separately in order to reduce the search space. Reducing search space means reducing the computational cost. Here the first stage acts as a filter to pass the most relevant images to the next stage where texture and shape (edge) features are extracted from images in the new sub-dataset constructed from the first stage. To extract texture information, LBP was used while to extract edge information Canny edge detector was used. The running time of the second stage depends on the number
of images in the new dataset. Therefore, the decreasing number of images in the second stage will decrease processing time. The proposed methodology improved performance by increasing precision and decreasing running time. Furthermore, to achieve better computational cost, the number of images passed from the first stage must be decreased. Therefore, in this paper, we duplicated the work in [16] with modification in the first stage. We found that calculating Squared Krawtchouk-Tchebichef polynomial (SKTP) [17] for each channel of the RGB color space of images before the calculation of mean and standard deviation has a great impact in reducing the number of images passed from the first stage to the second stage (see section 4). To the best of the author’s knowledge, it is the first time to use SKTP in the CBIR domain.

3. Proposed Methodology

In this section, we describe the feature descriptors utilized to extract color, texture, and edge. The framework architecture of the proposed CBIR is illustrated in Figure 1.

![Figure 1](image)

**Figure 1.** The framework of the proposed CBIR.

3.1. Feature extraction

3.1.1 The design of the first stage (color descriptor).

The first stage in the proposed algorithm is based on a global color descriptor to improve the effectiveness and minimize the complexity of the CBIR. The color feature is considered a robust feature descriptor because color features are considered invariant against translation, rotation, and scale change [18]. Global color descriptors have a faster response than local descriptors for a particular request. Orthogonal moments [17], which proved to reduce the computation cost and increase accuracy in other computer vision fields [19], are selected to characterize the color features of the entire image. The first step in this stage is computing OP, which is used to transform the signal to the transform domain [20]. It provides less computation cost and fewer coefficients [17] because it satisfies two properties: (1) Localization and (2) Energy compaction (EC). To represent the signal, EC property ensures that fewer moments (less component of transformed coefficients) used for signal representation. At the same time, the localization property offers a reduction in computing moments through pre-specifying region of interest. There are different hybrid forms of OP, for instance, Krawtchouk-Tchebichef polynomial (KTP) [21] and squared Krawtchouk-Tchebichef polynomial (SKTP) [17]. KTP and SKTP is formed from Krawtchouk polynomial (KP) [22] and Tchebichef polynomial (TP) [23]. In this paper, SKTP is used because it is outperformed KTP in terms of EC and localization properties [17]. In [17], the mathematical model of SKTP defined as follows:

\[
U_n(x; N) = \sum_{k=0}^{N-1} A_k(x; p, N)B_k(n; p, N), \quad (n, x = 0, 1, ..., N - 1)
\]
where $U_n(x; p, N)$ is the $n$th order of the SKTP, in this work, we experimentally chose $n$ to be 8, \(A_n(x; p, N)\) and \(B_n(x; p, N)\) are OPs generated from two basic OPs as:

\[
A_n(x; p, N) = \sum_{i=0}^{N-1} K_i(n; p)T_i(x) \tag{2}
\]

\[
B_n(x; p, N) = \sum_{i=0}^{N-1} K_i(x; p)T_i(n) \tag{3}
\]

\((n, x = 0, 1, ..., N - 1); p \in (0,1)\)

SKTP can be rewritten, using (1), (2), and (3), as follows [17]:

\[
U_n(x; N) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} K_i(n; p)T_i(i)K_j(i; p)T_j(x) \tag{4}
\]

Equations (2), (3), and (4) can be expressed in the form of matrix multiplication as:

\[
U = (U_k U_l)^2 \tag{5}
\]

the polynomials $U_n(x)$, $K_n(x)$, and $T_n(x)$ are represented in the matrix form $U_k$, $U_l$, and $U_i$, respectively. As mentioned before, SKTP is formed from TP and KP which are computed using fast recurrence relation proposed in [24] and [25]. To compute the moments of the two-dimensional signal (image), the following equation is used [20]:

\[
\Psi = U \times I \times U^T \tag{6}
\]

where $\Psi$ is the moments, $I$ is the image matrix, and \((\cdot)^T\) is the transpose operator. Moments are computed for each R, G, and B channels separately. Then statistical measures are utilized, which are the mean and standard deviation (calculating the absolute value of the mean and the standard deviation). The first one (mean) provides the average information of the distribution of the pixels for the image while the second one (standard deviation) provides information about the closeness of the distribution of the pixels around mean color.

In the second step of the first stage, the mean and standard deviation for the three channels of the RGB color space of the query image are calculated using equations (7) and (8).

\[
Mean(I_c) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \psi_{cij}, c = R, G, B \tag{7}
\]

\[
Std(I_c) = \sqrt{\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\psi_{cij} - Mean(I_c)\right)^2}, c = R, G, B \tag{8}
\]

where $I_c$ is the information of the color channel, and $n$ is the moments order. $\psi_{cij}$ represents the moments of the image at the specified color channel ($c$). As mentioned earlier, the Mean value provides the image’s average moments. Standard deviation is important because it gives detailed information about moments distribution about mean moment information. Therefore, the standard deviation is added and subtracted to and from the mean value computed for the color channels (R, G, and B) of the query image to act as lower and upper threshold (bound). This is represented mathematically by equations (9) and (10), which are utilized from [16]
The mean value of the R, G, and B channels of each image in the dataset is calculated and compared to the low and high thresholds of the query image. If this mean value lies between the two thresholds (including the low and high threshold values), then this image will be selected for the next stage of the proposed CBIR. The logical AND operator (&&) is used to combine the low and high thresholds of the three channels (R, G, and B).

This first stage acts as a filter that passes only images that satisfy the condition mentioned above to form a new dataset which is a subset of the original dataset. The next stage in the proposed CBIR employs this new dataset instead of the original dataset.

3.1.2 Second stage (Texture and edge Descriptor)

Texture features are usually used in image retrieval and pattern recognition because texture features are noticeable patterns that cannot stand alone as a single intensity or color which made them an essential feature in computer vision. LBP is used to extract texture features because it is invariant to any monotonic transformations in the grayscale. Moreover, LBP is computationally simple; thus a simple LBP is used [26]. Texture features are calculated for the new sub-dataset returned from the first stage. As a pre-processing step, images are transformed from RGB to grayscale color space. Then LBP is calculated by taking a $3 \times 3$ overlapping image, comparing the center pixel (CP) of the $3 \times 3$ sub-block and its eight surrounding neighbors, with the center pixel considered as a threshold. Binary representation is created by using this CP through the use of equation (11) [26], and the LBP value is updated in the CP of that block of the image.

\[
LBP_N = \sum_{i=0}^{N-1} f(P_i - CP)^2, \quad f(p) = \begin{cases} 1 & P \geq 0 \\ 0 & P \leq 0 \end{cases}
\]

where $N$ is the total neighbor pixels number surrounding the CP (here $N = 8$). $P_i$ is the value of the neighboring pixels, $i = \{0,1,2,3,\ldots,7\}$.

The edge descriptor is considered a powerful descriptor in content-based multimedia retrieval and indexing domains if it is extracted properly [27]. Usually, the rapid change in the image intensity value forms edges that are detected using edge detection algorithms [27]. The proposed work uses Canny edge detection to represent shape features of the images in the new sub-dataset created at the end of the first stage of the proposed algorithm. It is noteworthy that this part is adopted from [16].

The first step in edge feature extraction is converting images from RGB color space to HSV through the use of equations (12)-(14) [28]. This conversion is important because each color channel is highly correlated thus splitting luminance and chrominance is impossible. Although, edges can be marked in the grayscale, converting gray image to RGB image is impossible.

\[
H = \begin{cases} 60 \times \frac{G-B}{\delta} & R = \max(R,G,B) \\ 60 \times (2 + \frac{B-G}{\delta}) & G = \max(R,G,B) \\ 60 \times (4 + \frac{G-R}{\delta}) & B = \max(R,G,B) \end{cases}
\]

where $\delta = \max(R,G,B) - \min(R,G,B)$

\[
S = \frac{\delta}{\max(R,G,B)} \quad (13)
\]

\[
V = \max(R,G,B) \quad (14)
\]
where $H$ and $S$ hold the image color information and $V$ carries the image intensity. In the proposed work, the Canny edge detector is carried out on the $V$ channel and then fused with the un-processed $H$ and $S$ channel and converted back after combination to the RGB color space. The next step is to estimate the histogram of the R, G, and B to form the color edge features, which is adopted from [16].

3.2. Similarity Measure

In all types of retrieval systems, the similarity measure is an essential stage that gives distance information between the two images. It is noteworthy that the computed distance is considered the key factor of the similarity measure [29]. The smaller distance indicates a more similar image to the query image. Manhattan distance measure [30] is used to estimate the LBP and Canny edge feature similarity through the use of equations (15) and (16) [16]:

$$LBPM(Q_{LBP}(I), D_{BLBP}(I)) = \sum_{i=1}^{N} |f_{Q_{LBP}(I)}(i) - f_{D_{BLBP}(I)}(i)|$$

(15)

where $f_{Q_{LBP}(I)}(i)$ represents the $i$th LBP feature of the query image, $f_{D_{BLBP}(I)}(i)$ represents the $i$th LBP feature of a particular image in the new sub-dataset and $N$ is the total number of LBP features in an image.

$$EDGE_{SM}(Q_{LBP}(I), D_{BEDGE}(I)) = \sum_{i=1}^{N_{R}} |f_{Q_{EDGE(R)}(I)}(i) - f_{D_{BEDGE(R)}(I)}(i)| + \sum_{i=1}^{N_{G}} |f_{Q_{EDGE(G)}(I)}(i) - f_{D_{BEDGE(G)}(I)}(i)| + \sum_{i=1}^{N_{B}} |f_{Q_{EDGE(B)}(I)}(i) - f_{D_{BEDGE(B)}(I)}(i)|$$

(16)

where $f_{Q_{EDGE(B)}(I)}(i)$, $f_{Q_{EDGE(G)}(I)}(i)$, and $f_{Q_{EDGE(R)}(I)}(i)$ are the $i$th edge feature of the R, G, and B color channels, respectively, of the query image, $f_{D_{BEDGE(R)}(I)}(i)$, $f_{D_{BEDGE(G)}(I)}(i)$ and $f_{D_{BEDGE(B)}(I)}(i)$ are the $i$th edge feature on the R, G, and B color channels, respectively, of the new dataset image. $N_{R}$, $N_{G}$ and $N_{B}$ are the total number of image edge features in the R, G, and B color channels, respectively. Min – Max normalization [31] is calculated through equations (17) and (18) on the edge and texture features to bound the low and high variation values to the range of [0 1].

$$NORMAL_{EDGE_{SM}}(i) = \frac{EDGE_{SM}(i) - \min(EDGE_{SM})}{\max(EDGE_{SM}) - \min(EDGE_{SM})}, i = 1, 2, \ldots, k$$

(17)

$$NORMAL_{LBP_{SM}}(i) = \frac{LBP_{SM}(i) - \min(LBP_{SM})}{\max(LBP_{SM}) - \min(LBP_{SM})}, i = 1, 2, \ldots, k$$

(18)

where $k$ represents the total number of images in the newly obtained image dataset. It is important to note that $k$ varies according to the first stage selection rule used. $EDGE_{SM}(i)$ and $LBP_{SM}(i)$ represents
edge and LBP similarity measure value respectively of the \((i)\)th image in the new dataset. 

\(\min(EDG_{SM})\) and \(\min(LBP_{SM})\) are the minimum edge and texture feature similarity value in the entire new image dataset. \(\max(EDG_{SM})\) and \(\max(LBP_{SM})\) are the maximum edge and texture feature similarity value in the whole new image dataset.

The normalized texture and edge features are fused with equal weights. Then the resulted features are sorted ascendingly to get the most similar images at the top.

4. Experimental Results

To assess the proposed framework, experiments are carried out using the Wang database [8]. This dataset contains 1000 colored images and the size of the images is either \(384 \times 256\) or \(256 \times 384\). Images are organized in ten groups namely (African tribes, Sea, Buildings, Bus, Dinosaurs, Elephants, Flowers, Horse, Mountains, and Food). The results of the experiments are computed on Intel core i5 with 8 GB RAM and 250GB hard disk using MATLAB. To reduce computation cost, color feature datasets are constructed offline from the Wang dataset.

To evaluate the proposed framework, precision, and recall [33]–[35] are used to assess the performance of the system in terms of accuracy. Equations listed below are used:

\[
P = \frac{\text{No. of relevant images}}{\text{No. of retrieved images}}
\]

\[
R = \frac{\text{No. of relevant images}}{\text{No. of all relevant images in dataset}}
\]

where \(P\) and \(R\) denote precision and recall respectively. In this experiment, the proposed framework tested through the use of Wang dataset, classes are labeled as follow: 1- African tribes, 2-Sea, 3- Buildings, 4-Bus, 5-Dinosaurs, 6-Elephants, 7-Flowers, 8-Horse and 9-Mountains and 10-Food, twenty images are randomly selected for each class just as mentioned in [16] to duplicate their work and the results for [16] and our work are shown in table (1) and Figure 2. Better accuracy results could be achieved by considering other methods in second stage or may be by using three stages which need more investigations.

| Wang’s Database  | Pavithra [16] | Proposed System |
|------------------|---------------|-----------------|
| African tribes   | 80.25         | 82              |
| Sea              | 62.5          | 60.5            |
| Buildings        | 81            | 78.25           |
| Bus              | 97            | 96.75           |
| Dinosaurs        | 100           | 100             |
| Elephants        | 74            | 71.75           |
| Flowers          | 95.5          | 95.75           |
| Horse            | 98            | 98.25           |
| Mountains        | 75            | 72              |
| Food             | 75.5          | 70.75           |
| Average Precision| 83.875        | 82.6            |

We also measured the number of retrieved images from the first stage for [16] and our work and it is shown in Table 2 and Table 3, respectively; along with the time required for feature extraction and retrieval time.
From Table 1 and Figure 2, the proposed work is slightly less in accuracy than [16] but from observation for Table 2 and Table 3 and Figure 3 the average of total images retrieved from the first stage in the proposed work is less than the average of total images retrieved from the first stage in [16] by a great number of images (approximately 133 images), see Figure 2, which means the more valuable amount of savings in terms of time and storage. In general, there is a tradeoff between accuracy and computation time.

![Figure 2. Performance comparison between the proposed system and [16] using precision %.](image)

Table 2. Performance measurements for the system in [16] in terms of time (in second) and the number of returned images from the first stage.

| Class No. | Color extraction time | search 1000 image for color features and build a new database | LBP and Canny extraction time | building LBP and Canny database time | Distance Measure time | total time | No. of returned images from the first stage |
|-----------|-----------------------|-------------------------------------------------------------|-----------------------------|-------------------------------------|----------------------|-----------|------------------------------------------|
| 1         | 0.0294                | 10.6381                                                     | 0.2725                      | 30.3618                             | 0.0746               | 41.3764   | 706                                      |
| 2         | 0.0268                | 8.7696                                                      | 0.2757                      | 24.8905                             | 0.0678               | 34.0304   | 578                                      |
| 3         | 0.0284                | 9.8877                                                      | 0.2677                      | 27.9141                             | 0.0715               | 38.1694   | 658                                      |
| 4         | 0.0285                | 12.2272                                                     | 0.2685                      | 34.2799                             | 0.0743               | 46.8784   | 787                                      |
| 5         | 0.0285                | 1.8477                                                      | 0.2636                      | 5.4494                              | 0.0634               | 7.6526    | 130                                      |
| 6         | 0.0276                | 6.9907                                                      | 0.2602                      | 19.6075                             | 0.0695               | 26.9555   | 464                                      |
| 7         | 0.0281                | 4.3832                                                      | 0.2575                      | 12.5786                             | 0.0649               | 17.3123   | 298                                      |
| 8         | 0.028                | 4.5869                                                      | 0.2666                      | 12.9737                             | 0.0662               | 17.9214   | 305                                      |
| 9         | 0.0298                | 7.7812                                                      | 0.2637                      | 21.7191                             | 0.0692               | 29.863    | 526                                      |
| 10        | 0.0288                | 8.3453                                                      | 0.2642                      | 23.4548                             | 0.0687               | 32.1618   | 556                                      |
| Average   | 0.0283                | 7.5457                                                      | 0.266                        | 21.3229                             | 0.069                | 29.2321   | ~501                                    |
Table 3. Performance measurements for the proposed system in terms of time (in second) and the number of returned images from the first stage.

| Class No. | Color extraction based on Moments time | search 1000 image for color features and build a new database time | LBP and Canny extraction time | building LBP and Canny database time | Distance Measure time | Total time | No. of returned images from the first stage |
|-----------|---------------------------------------|---------------------------------------------------------------|-------------------------------|--------------------------------------|----------------------|-----------|------------------------------------------|
| 1         | 0.1553                                | 8.0072                                                        | 0.2834                        | 21.8055                              | 0.0729               | 30.3243   | 506                                       |
| 2         | 0.0799                                | 7.3057                                                        | 0.2675                        | 20.5464                              | 0.0692               | 28.2687   | 484                                       |
| 3         | 0.0812                                | 7.6003                                                        | 0.2667                        | 21.3749                              | 0.0693               | 29.3924   | 501                                       |
| 4         | 0.0818                                | 8.1842                                                        | 0.2668                        | 22.7636                              | 0.0700               | 31.3664   | 537                                       |
| 5         | 0.0915                                | 1.9715                                                        | 0.2674                        | 5.6309                               | 0.0614               | 8.0227    | 128                                       |
| 6         | 0.0803                                | 5.7552                                                        | 0.2569                        | 16.5087                              | 0.0666               | 22.6677   | 391                                       |
| 7         | 0.0805                                | 2.4452                                                        | 0.2527                        | 6.9095                               | 0.0627               | 9.7506    | 157                                       |
| 8         | 0.0802                                | 3.4170                                                        | 0.2548                        | 9.6538                               | 0.0628               | 13.4686   | 230                                       |
| 9         | 0.0813                                | 6.7010                                                        | 0.2649                        | 19.1056                              | 0.0655               | 26.2183   | 454                                       |
| 10        | 0.0811                                | 4.2940                                                        | 0.2615                        | 12.1146                              | 0.0659               | 16.8171   | 292                                       |
| Average   | 0.0893                                | 5.5681                                                        | 0.2642                        | 15.6413                              | 0.0666               | 21.6296   | ~368                                      |

In this paper, we were concerned with reducing computation time through reducing the number of returned images from the first stage which acts as a filter to pass images to the second stage to calculate the texture and edge features which means reducing the processing time for the second stage and in turn reducing the overall required time. According to equation (21) shown below:

\[
\text{Computation time improvement ratio} = \frac{T_{\text{new}}}{T_{\text{old}}} \times 100\% \quad (21)
\]

where \(T_{\text{new}}\) denotes the total time in the proposed work while \(T_{\text{old}}\) denotes the total time in [16]. The computation time improvement ratio of the proposed work is 73.99% which is considered a great improvement.

Figure 3. Number of returned images from first stage.
5. Conclusion
This study presented an efficient multistage CBIR method to retrieve images based on low-level features. The number of images to be searched has a direct impact on the computation cost of any searching method. Thus, in this study, the focus was to reduce images passed from the first stage to the second one by depending on the use of SKTP and then calculating the mean and standard deviation of the image. Then, LBP and Canny edge detection methods are utilized for extracting the texture and shape features respectively, from passed images from the first stage which is much less than the original dataset to achieve faster retrieval. Manhattan distance is used as a similarity measure. The proposed system has been tested using Wang’s dataset. As future work, we intended to test our work using other datasets and pay more attention to acquire higher accuracy by experimenting with other low-level feature extracting methods, use machine learning algorithms to extract low level features and examine the performance in case of using three stages.

6. References
[1] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, “A survey of content-based image retrieval with high-level semantics,” Pattern Recognit., vol. 40, no. 1, pp. 262–282, Jan. 2007.
[2] B. Raghunathan and S. T. Acton, “A content based retrieval engine for circuit board inspection,” in Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348), vol. 1, pp. 104–108.
[3] C. Bai, J. Chen, L. Huang, K. Kpalma, and S. Chen, “Saliency-based multi-feature modeling for semantic image retrieval,” J. Vis. Commun. Image Represent., vol. 50, no. November 2017, pp. 199–204, Jan. 2018.
[4] N. Shrivastava and V. Tyagi, “Corrigendum to “Content based image retrieval based on relative locations of multiple regions of interest using selective regions matching” [Information Sciences 259 (2014) 212–224],” Inf. Sci. (Ny), vol. 421, p. 273, Dec. 2017.
[5] X.-Y. Wang, J.-F. Wu, and H.-Y. Yang, “Robust image retrieval based on color histogram of local feature regions,” Multimed. Tools Appl., vol. 49, no. 2, pp. 323–345, Aug. 2010.
[6] G. Pass and R. Zabih, “Histogram refinement for content-based image retrieval,” in Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV’96, 1996, pp. 96–102.
[7] D. G. Low, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vis., pp. 91–110, 2004, [Online]. Available: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf.
[8] D. P. Tian, “A review on image feature extraction and representation techniques,” Int. J. Multimed. Ubiquitous Eng., vol. 8, no. 4, pp. 385–395, 2013.
[9] D. Zhang and G. Lu, “Review of shape representation and description techniques,” Pattern Recognit., vol. 37, no. 1, pp. 1–19, Jan. 2004.
[10] Y. D. Chun, N. C. Kim, and I. H. Jang, “Content-Based Image Retrieval Using Multiresolution Color and Texture Features,” IEEE Trans. Multimed., vol. 10, no. 6, pp. 1073–1084, Oct. 2008.
[11] A. Ponomarev, H. S. Nalamwar, I. Babakov, C. S. Parkhi, and G. Buddhawar, “Content-Based Image Retrieval Using Color, Texture and Shape Features,” Key Eng. Mater., vol. 685, pp. 872–876, Feb. 2016.
[12] P. Srivastava and A. Khare, “Integration of wavelet transform, Local Binary Patterns and moments for content-based image retrieval,” J. Vis. Commun. Image Represent., vol. 42, pp. 78–103, Jan. 2017.
[13] A. Nazir, R. Ashraf, T. Hamdani, and N. Ali, “Content based image retrieval system by using HSV color histogram, discrete wavelet transform and edge histogram descriptor,” in 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Mar. 2018, vol. 2018-Janua, pp. 1–6.
[14] C. S. W. Won, D. K. P. Park, and S.-J. P. Park, “Efficient Use of MPEG-7 Edge Histogram Descriptor,” ETRI J., vol. 24, no. 1, pp. 23–30, Feb. 2002.
[15] S. P. Rana, M. Dey, and P. Siarry, "Boosting content based image retrieval performance through integration of parametric & nonparametric approaches," J. Vis. Commun. Image Represent., vol. 58, pp. 205–219, Jan. 2019.

[16] L. K. Pavithra and T. S. Sharmila, "An efficient framework for image retrieval using color, texture and edge features," Comput. Electr. Eng., vol. 70, pp. 580–593, Aug. 2018.

[17] S. H. Abdulhussain, A. R. Ramli, B. M. Mahmod, M. I. Saripan, S. A. R. Al-Haddad, and W. A. Jassim, "A New Hybrid form of Krawtchouk and Tchebichef Polynomials: Design and Application," J. Math. Imaging Vis., vol. 61, no. 4, pp. 555–570, May 2019.

[18] N. Shrivastava and V. Tyagi, "An efficient technique for retrieval of color images in large databases," Comput. Electr. Eng., vol. 46, pp. 314–327, Aug. 2015.

[19] S. H. Abdulhussain, A. R. Ramli, A. J. Hussain, B. M. Mahmod, and W. A. Jassim, "Orthogonal polynomial embedded image kernel," in Proceedings of the International Conference on Information and Communication Technology - ICICT '19, 2019, pp. 215–221.

[20] S. H. Abdulhussain, S. A. R. Al-Haddad, M. I. Saripan, B. M. Mahmod, and A. Hussien, “Fast Temporal Video Segmentation Based on Krawtchouk-Tchebichef Moments,” IEEE Access, vol. 8, pp. 72347–72359, 2020.

[21] B. M. Mahmod, A. R. bin Ramli, S. H. Abdulhussain, S. A. R. Al-Haddad, and W. A. Jassim, "Signal compression and enhancement using a new orthogonal-polynomial-based discrete transform," IET Signal Process., vol. 12, no. 1, pp. 129–142, Feb. 2018.

[22] S. Abdulhussain, A. Ramli, M. Saripan, B. Mahmod, S. Al-Haddad, and W. Jassim, "Methods and Challenges in Shot Boundary Detection: A Review," Entropy, vol. 20, no. 4, p. 214, Mar. 2018.

[23] S. H. Abdulhussain, A. R. Ramli, S. A. R. Al-Haddad, B. M. Mahmod, and W. A. Jassim, "On Computational Aspects of Tchebichef Polynomials for Higher Polynomial Order," IEEE Access, vol. 5, no. March, pp. 2470–2478, 2017.

[24] S. H. Abdulhussain, A. R. Ramli, S. A. R. Al-Haddad, B. M. Mahmod, and W. A. Jassim, "Fast Recursive Computation of Krawtchouk Polynomials," J. Math. Imaging Vis., vol. 60, no. 3, pp. 285–303, 2018.

[25] Pew-Thian Yap, R. Paramesran, and Seng-Huat Ong, "Image analysis by krawtchouk moments," IEEE Trans. Image Process., vol. 12, no. 11, pp. 1367–1377, Nov. 2003.

[26] T. Ojala, M. Pietikäinen, and T. Maenpaa, “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.

[27] S. Kranazy, M. Ferreira, and M. Gabbouj, "Automatic Object Extraction Over Multiscale Edge Field for Multimedia Retrieval," IEEE Trans. Image Process., vol. 15, no. 12, pp. 3759–3772, Dec. 2006.

[28] A. Mojsilovic, H. Hu, and E. Soljanin, “Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis,” IEEE Trans. Image Process., vol. 11, no. 11, pp. 1238–1248, Nov. 2002.

[29] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: A comprehensive study," J. Vis. Commun. Image Represent., vol. 32, no. July, pp. 20–54, Oct. 2015.

[30] S.-H. Cha, "I. Comprehensive Survey on Distance Similarity," Int. J. Math. Model. Methods Appl. Sci., vol. 1, no. 4, 2007.

[31] G.-H. Liu and J.-Y. Yang, "Content-based image retrieval using color difference histogram," Pattern Recognit., vol. 46, no. 1, pp. 188–198, Jan. 2013.

[32] J. Z. Wang, J. Li, and G. Wiederhold, "SIMPLICity: Semantics-sensitive Integrated Matching for Picture Libraries," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1929, 2000, pp. 360–371.

[33] M. S. Haji, M. H. Alkawaz, A. Rehman, and T. Saba, "Content-based image retrieval: a deep
look at features prospectus,” *Int. J. Comput. Vis. Robot.*, vol. 9, no. 1, p. 14, 2019.

[34] A. ElAdel, R. Eibali, M. Zaied, and C. Ben Amar, “A hybrid approach for Content-Based Image Retrieval based on Fast Beta Wavelet network and fuzzy decision support system,” *Mach. Vis. Appl.*, vol. 27, no. 6, pp. 781–799, Aug. 2016.

[35] A. Bala and T. Kaur, “Local texton XOR patterns: A new feature descriptor for content-based image retrieval,” *Eng. Sci. Technol. an Int. J.*, vol. 19, no. 1, pp. 101–112, Mar. 2016.