



A Robust Image Retrieval Method Using Multi-Hierarchical Agglomerative Clustering and Davis-Bouldin Index

Rahmad Hidayat^{1,2} Agus Harjoko^{3*} Aina Musdholifah³

¹*Doctoral Program Department of Computer Science and Electronics,
Faculty of Mathematics and Natural Science, Universitas Gadjah Mada, Yogyakarta, Indonesia*

²*Department of Information Technology and Computer, Politeknik Negeri Lhokseumawe, Lhokseumawe, Indonesia*

³*Department of Computer Science and Electronics, Faculty of Mathematics and Natural Science,
Universitas Gadjah Mada, Yogyakarta, Indonesia*

* Corresponding author's Email: aharjoko@ugm.ac.id

Abstract: An image retrieval system is required to provide high accuracy in a short time. Combining various features will usually increase accuracy but also increase retrieval time. This study developed a CBIR (Content-Based Image Retrieval) method based on hierarchical clustering on low-level features. Low-level features consisting of color, texture, and shape are extracted and then clustered hierarchically. The resulting clusters are then validated to obtain their optimal number. In the retrieval process, the query image features are extracted and compared with the cluster centroid on each feature. The scores of query results on each feature are normalized, and then the normalized scores are weighted to get the total score. The experiment was carried out using three datasets, namely DIKE20, Corel-1k, and Corel-10k. Based on the experimental result, the proposed method shows better performance compared to the existing state-of-the-art method. On the Corel-1k and Corel-10k datasets, the proposed method obtained precision scores of 0.81 and 0.62, respectively.

Keywords: Low-levels features, Hierarchical clustering, Cluster validation, Binary operations.

1. Introduction

Image is widely used on the internet. Sharing it on various social media platforms has produced a massive number of images. Various methods have been developed to display, store, and search images efficiently. Content-Based Image Retrieval (CBIR) is a method that is often used to search for images based on their content. Combining various features in a CBIR system causes problems, including increasing retrieval time [1]. An online-based CBIR system is required to provide retrieval results with high accuracy quickly [2].

Various low-level features are commonly used in CBIR, namely color, texture, and shape [3]. The combination of these features can provide high retrieval accuracy [4]. Color is a visual feature that is widely used in CBIR. It is because the color features can be easily and quickly computed [5]. Texture

features refer to visual patterns that have similarities that are not produced by a single color [6]. The important characteristic of texture is, it can distinguish two images with the same color and shape [7]. Shape features provide robust and efficient information in representing objects as well as distinguishing them [8]. Combining all these low-level features can improve accuracy but also increase retrieval time.

The clustering algorithm is a powerful processing method that can quickly classify massive resources in a short period of time [9]. CBIR studies that utilize the hierarchical clustering method were carried out among others by [10, 11]. There are several advantages of using hierarchical clustering in CBIR, including fast and flexible implementation. In addition, another advantage is no need for initialization of the number of clusters and a relatively faster retrieval [12].

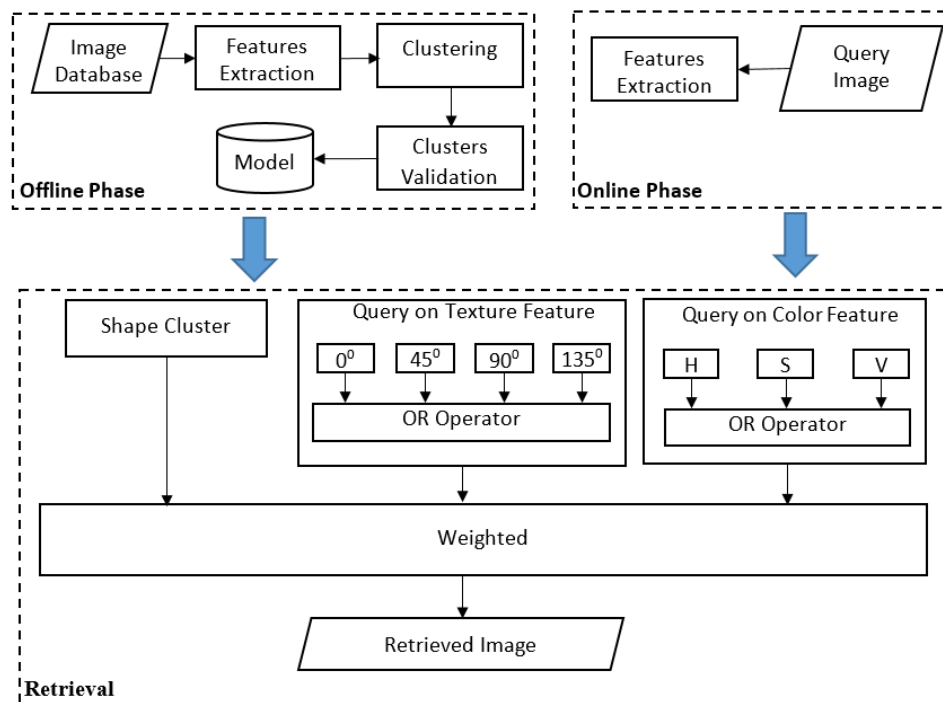


Figure. 1 Image retrieval method using multi-hierarchical agglomerative clustering and davies-bouldin index

In this paper, we propose a clustering-based CBIR method. Each low-level feature is extracted and then clustered hierarchically. Color features are extracted using the Color Moment (CM) method on each channel (HSV), and then each features vector are clustered separately. The Gray Level Co-occurrence Matrix (GLCM) method is used to extract texture features. The extraction process on the texture feature is carried out in four directions (0° , 45° , 90° , and 135°). The feature vector of each direction is then clustered separately. The shape feature is extracted using the Hu-moment method, where the resulting feature vectors, which consist of seven moments, are then clustered. To get the optimal number of clusters, each cluster that has been formed are then validated using the Davis-Bouldin Index (DBI). A small DBI shows the optimal number of clusters. The query process is carried out on each feature. The query image features are extracted and then compared with each cluster centroid. The query results on each feature are then combined using binary operations. The combined query results are then weighted to get the total scores.

The rest of this paper is organized as follows: Section 2 presents the literature review; Section 3 describes the proposed method; Section 4 presents the result and discussion; Section 5 presents the benchmarking results; Finally, Section 6 presents the conclusion of this paper.

2. Literature review

Many methods have been proposed to improve the performance of CBIR system based on low-level features. Research conducted by [1] developed an image retrieval framework that combines 8-Directional Gray Level Co-occurrence Matrix (8D-GLCM) and HSV Color Moments (HSVCM) features. GLCM constructed in 8 angles with pixel distance 1. Meanwhile, HSV color moments (HSVCM) features is extracted using two steps : Transformation of the input RGB image into HSV color space and Computation of mean (average color information), standard deviation (total number of pixels that differs from the mean) and Skewness from each of the HSV components. Although the required retrieval time is quite low, the precision scores obtained on large datasets still need to be improved.

A study by [13] proposed a correlated microstructure descriptor (CMSD) that represents high-level semantics by establishing and detecting correlations between texture orientation, color, and intensity features. Color features are extracted by quantizing each channel of the HSV color space. Edge detection is performed using a multi-directional sobel operator, while contrast information is obtained by quantizing the V of the HSV channel. Correlated microstructures are then identified and mapped to obtain micro-color, micro-orientation, and micro-intensity maps, respectively. Despite the

computational complexity of extracting and constructing correlations between features, the experimental results show insignificant increase in precision and recall scores compared to other state-of-art methods.

Research conducted by [14] proposed a CBIR method that utilizes texon templates to detect and extract consistent zones from images. The dominant color descriptor (DCD) is then extracted from the consistent zone. In addition, the translation and rotation invariance of the Hu moments feature is applied to extract the shape information in the same consistent zone of the image. At the experimental stage, this study carried out various levels of quantization in extracting CDCs for retrieval. While on the retrieval process of the shape features, this study performs a combination of different weights for each moment (seven moment). Although the feature vectors formed are quite long (2598), the precision and recall scores still need to be improved.

The CBIR method developed by [15] made improvements to the local binary pattern (LBP) method by combining an adaptive threshold and a directional local pattern. The standard deviation of the gray value of all pixels in the local neighborhood of the image and the gray value of the center pixel are used as comparison threshold for encoding. This threshold is adaptively changed following the grayscale changes in the neighborhood. The directional local pattern is obtained by comparing the grayscale changes of local neighborhood pixels in different directions, discrimination, and binary encoding are performed to indicate the local texture features directionality. The experiment shows that the best precision score obtained by this method is still quite low.

The study conducted by [16] developed an image retrieval method based on the hierarchical clustering of images that have been grouped semantically. In this study, color features are represented in a 68-dimensional color histogram. Meanwhile, the texture features consist of 48 dimensions with mean and standard deviation coefficients. The images are hierarchically clustered and then validated using the DBI. Although the precision score is good, the retrieval time is relatively high, and the use of semantic labels on images can cause different perceptions.

The study by [17] presents a deep convolutional neural network-based model called MaxNet for content-based image retrieval. The proposed MaxNet model contains a total of twenty-one convolution layers that are iterated in a structured way to extract maximum information from the images. After the MaxNet model trained, the dataset is fed into the

network, and the feature vectors are extracted from the last layer of the proposed MaxNet model. On the retrieval process, the query image is feed-forwarded through the network. The extracted features of the query image are compared with feature vectors of the entire dataset. The experiment results show that the precision and recall scores on the small dataset are good, but on the large dataset the scores are quite low.

Based on the observations of the CBIR methods above, this study develops a CBIR method based on hierarchical clustering on low-level features to obtain high precision and recall scores with fast retrieval time.

3. Methodology

The CBIR method developed in this study extracts low-level features consisting of color, texture, and shape. On the color feature extraction, the CM method generated three moments: mean, standard deviation, and skewness, from each channel. The texture features are extracted using the GLCM method in four directions (0° , 45° , 90° , and 135°). The shape features are extracted using Hu-moment method which consists of seven moments.

The feature vectors are then clustered hierarchically. In the color feature, the clustering process is performed on each channel. In the texture feature, the clustering process is performed on four directions, while on the shape feature the clustering process is performed based on seven moments. The clusters in each feature are then validated using the DBI to get their optimal number.

On the retrieval process, the query image features are extracted and then the query process is carried out on each feature. The binary operator OR is used to combine the query result between clusters in each feature. The scores on each feature are normalized and then the total score is calculated using weighting operations. Fig. (1) shows the proposed CBIR method developed in this study.

3.1 Feature extraction

3.1.1. Color feature extraction

Color Moment (CM) is a color feature extraction method that assumes that the color distribution in the image can be interpreted as a probability distribution. The study by Stricker and Orengo (1995) utilizes three center moments of the color distribution in the image, namely the mean, standard deviation, and skewness. An image can have several color channels, so the moments are generated from each channel. If an image has three color channels, then it will have nine moments (three moments for each channel). [18]

The first moment in CM is mean. Mean can be interpreted as the average value of the pixels in the image. Mean is defined in Eq. (1).

$$E_i = \sum_{j=1}^N \frac{1}{N} p_{ij} \tag{1}$$

Where N is the number of pixels, p_{ij} is the pixel value at the j_{th} index, and i is the color channel.

The standard deviation is the root of the variance distribution of the pixels in the image. The standard deviation is defined in Eq. (2).

$$\sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^2\right)} \tag{2}$$

Where σ_i is a standard deviation, and E_i is the mean. Skewness is asymmetry levels of the distribution pixels in the image. Skewness is defined in Eq. (3).

$$S_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^N (p_{ij} - E_i)^3\right)} \tag{3}$$

Where S_i is the skewness.

3.1.2. Texture features extraction

The Gray Level Co-Occurrence Matrix (GLCM) method was developed by Haralick et al., (1973). This method is widely used in various applications, including breast cancer detection [19]. GLCM is obtained by accumulating the number of gray pairs of the pixels in the image. GLCM is computed based on spatial distance (D) and orientation ($0^\circ, 45^\circ, 90^\circ$, or 135°) at certain gray levels (G). After the GLCM matrix is generated, the combined probability function is used to normalize GLCM by dividing each pixel pair set with the total pixel pair number. Probability function which is notated as $p(r, n)$ shown in Eq. (4).

$$p(r, n) = \frac{1}{\sum_{r=0}^{G-1} \sum_{n=0}^{G-1} C_d(r, n)} C_d(r, n) \tag{4}$$

Where $p(r, n)$ is a probability function, G is a gray level, and $C_d(r, n)$ represents the total pair of pixels.

Many statistical features can be derived from the GLCM matrix, including energy, entropy, homogeneity, contrast, correlation, local homogeneity, variance, sum of average, sum entropy, difference entropy, inertia, cluster shade, and cluster prominence [20]. This study uses only five features that are considered significant in representing object textures, namely energy, entropy, homogeneity, contrast, and correlation. Energy calculates the sum

of squared elements in the GLCM. Energy is computed using Eq. (5).

$$energy = \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} p(r, n)^2 \tag{5}$$

Entropy returns a value that measures the irregularity of the gray level in the image. Entropy is computed as follows:

$$entropy = \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} p(r, n) \log p(r, n) \tag{6}$$

Homogeneity measures the proximity in the distribution of elements in GLCM to GLCM diagonally. Homogeneity is calculated using Eq. (7).

$$homogeneity = \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} \frac{p(r, n)}{(1+|r-n|)} \tag{7}$$

Contrast is a measure of the presence of variations in the gray level of pixels in the image. Contrast is calculated using Eq. (8).

$$contras = \frac{1}{(G-1)^2} \sum_{r=0}^{G-1} \sum_{n=0}^{G-1} (r - n)^2 p(r, n) \tag{8}$$

Correlation measures the degree of correlation between neighboring pixels in the image. Correlation is calculated using Eq. (9).

Table 1. The seven moment in Hu-moment invariants

Moment	Equation
φ_1	$(\eta_{20} + \eta_{02})$
φ_2	$(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$
φ_3	$(\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$
φ_4	$(\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$
φ_5	$(\eta_{30} - 3\eta_{12}) (\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03}) (\eta_{21} + \eta_{03}) - [3(\eta_{30} + \eta_{12})^2] - (\eta_{21} + \eta_{03})^2$
φ_6	$(\eta_{20} - \eta_{02}) + [(3\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$
φ_7	$(3\eta_{21} - \eta_{03}) (\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} + 3\eta_{12}) (\eta_{21} + \eta_{03}) [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$

$$correlation = \sum_{i_1}^{G-1} \sum_{i_2}^{G-1} P_{i_1, i_2} \frac{(i_1 - \mu_{i_1})(i_2 - \mu_{i_2})}{\sqrt{(\sigma_{i_1}^2)(\sigma_{i_2}^2)}} \quad (9)$$

3.1.3. Shape feature extraction

Hu-moment has been used to extract shape features in various images, including CT scan images with satisfactory results [20]. In his study, Hu (1962) derived a collection of invariants using invariant algebra. The moment of a two-dimensional image with size $M \times M$ has a gray function $f(x,y)$, ($x, y=0, \dots, M-1$) defined as in Eq. (10).

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x)^p \cdot (y)^q f(x,y) \quad (10)$$

where $p, q = 0, 1, 2, 3$

Specifically, Hu defined seven moments using order up to 3 that is invariant to scale, position, and orientation. These seven moments are shown in Table 1.

3.2 Clustering using hierarchical agglomerative clustering method (HACM)

The HACM method performs bottom-up clustering, where initially, each data is represented as a single cluster (the number of initial clusters is equal to the number of objects). The clusters are then combined based on their distance, where clusters that are close to each other will be merged. This process is repeated until all clusters are merged into a single cluster. The following is the HACM clustering algorithm:

Algorithm 1: HACM

- 1:** $F_i[]$; (Image Features)
 - 2: repeat**
 - 3:** Generate distance matrix between image/cluster (ward)
 - 4:** Merge the pair of clusters with the smallest distance
 - 5:** Update distance matrix
 - 6: until** all clusters merged into a single cluster
-

Distance measurement in data processing measures whether a data has similarities with other data [12]. We use Euclidean Distance to generate a distance matrix between clusters, where based on the resulting distance matrix, the merge process is executed. Let point p have Cartesian coordinates (x_1, y_1) and point q have coordinates (x_2, y_2) . Then the distance between point p and q is given by:

$$d_{p,q} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (11)$$

3.3 Cluster validation

Cluster validation is a technique to find the cluster set with the best natural partition. In general, there are two types of cluster validation, namely external and internal cluster validation methods. External validation methods use external information related to data, such as F-Measure purity, entropy, NMI-Measure, and others. The internal cluster validation method uses the intrinsic information contained in the data [21]. Davies-Bouldin Index (DBI) is an internal cluster validation based on the estimated distance and dispersion of the cluster to get the final result that represents the optimal partition [22]. The DBI can be calculated using the following formula:

$$DBI = \frac{1}{c} \sum_{i=1}^c \text{Max}_{j=1, 2, \dots, c, j \neq i} \left\{ \frac{d(X_i) + d(X_j)}{d(c_i, c_j)} \right\} \quad (12)$$

3.4 The merging of the multiple queries results

In combining multi-query results, binary operations are the most commonly used operators. Besides being easy to implement, binary operators can also increase the confidence factor in retrieval results [23]. The OR operator is a binary operator which generally used to combine query results with equal degrees. On the color feature query process, if the retrieval result on channel H is denoted Rw_h , the retrieval result on channel S is denoted as Rw_s , and the retrieval result for channel V is denoted as Rw_v , then the final query result for the color features is computed as follows:

$$Rw = Rw_h \cup Rw_s \cup Rw_v \quad (13)$$

On the texture feature query process, if the retrieval result at direction 0° is denoted Rt_0 , the retrieval result at direction 45° is denoted as Rt_1 , the retrieval result at direction 90° is denoted as Rt_2 , and the retrieval result at direction 135° is denoted as Rt_3 . The final query result for texture feature (Rt) computed as follows:

$$Rt = Rt_0 \cup Rt_1 \cup Rt_2 \cup Rt_3 \quad (14)$$

The normalization process aims to uniform the scores from various query results so that they are comparable. The Normalization process generates the scores to the 0-1 range. If x is a score of an image and x_{max} is the maximum score in the data, then the normalized score x' computed as follows:

$$x' = 1 - \frac{x}{x_{max}} \tag{15}$$

The normalized scores on each feature are then multiplied by its respective weight to get the total score. Weighting is a process that involves various factors by giving weight to each of these factors to achieve the desired ranking of alternatives [24]. If there are n retrieval systems ir_1, ir_2, \dots, ir_n and for each query q , returns the result r_i . Each r_i is a ranking sequence of images with their respective scores. The weighting method uses Eq. (16) to calculate the total score for image d [25].

$$M(x, q) = \sum_{i=1}^n \beta_i s_i(x', q) \tag{16}$$

Where $S_i(x', q)$ is the normalized score for image x , i is the weight for the enviable system, and $M(x, q)$ is the total score for image x . The experiments conducted by [26] assign weights 0.3, 0.3, and 0.4 for color, texture, and shape features, respectively. The experiment result shows satisfactory results with an average score of accuracy is 0.95.

3.5 Performance measurement

In measuring performance using the confusion matrix, there are four variables that represent the results of the classification process. True Negative (TN) is the number of negative data detected correctly, while False Positive (FP) is negative data but detected as positive. Meanwhile, True Positive (TP) is positive data that is detected correctly as positive. False Negative (FN) is the opposite of True Positive, where the positive data is detected as negative data [27]. Based on TN, FP, FN, and TP, the precision and recall score can be derived. The precision describes the number of correctly classified positive category data divided by the total number of positive classified data. The precision score is calculated using Eq. (17). Meanwhile, the recall score shows the percentage of positive category data which are correctly classified by the system. Eq. (18) is used to calculate the recall score [28].

$$Precision = \frac{TP}{FP+TP} \tag{17}$$

$$Recall = \frac{TP}{FN+TP} \tag{18}$$

In addition, performance measurements of the methods are also carried out using retrieval time. Retrieval time is measured from the time the query image is entered into the system until the entire retrieval process is completed.



Figure. 2 Backpack localization using superpixel segmentation



Figure. 3 The example of images: (a) Corel-1k and (b) Corel-10k

3.6 Datasets

The datasets used to develop the retrieval method in this study consist of the DIKE20, Corel-1k, and Corel-10k. The detail of each dataset is described as follows:

3.6.1. DIKE20 dataset

We built the DIKE20 dataset by collecting pedestrian video from the video surveillance system of the Faculty of Mathematics and Natural Sciences Universitas Gadjah Mada. Using frame different method, we separate the foreground and the background area of the video. The foregrounds area are then segmented using the superpixel method. Fig. 2 shows the process of localizing the backpack area using the superpixels segmentation method.

Based on the process described above, there were 327 backpack images with various orientations, colors, and sizes collected. The examples of images in the DIKE20 dataset are presented in Fig. 3.

3.6.2. Corel-1k and Corel-10k datasets

The Corel-1k dataset consists of 10 categories of images: African, beach, building, bus, dinosaur, flower, elephant, horse, mountain, and food. Each category has 100 images (total of 1,000 images). Fig. 4 displays the example of images in the Corel-1k dataset

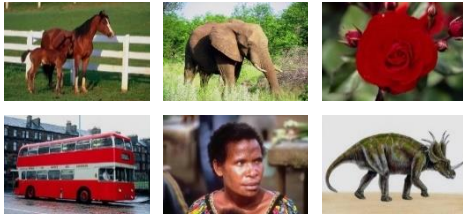


Figure. 4 The example of images in Corel-1k dataset

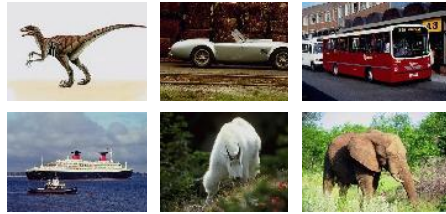


Figure. 5 The example of images in Corel-10k dataset

The Corel-10k dataset consists of 100 categories of images: art, aviation, cat, dog, owl, tiger, lion, etc. Each category has 100 images, a total of 10,000 images. Fig. 5 displays the example of images in the Corel-10k dataset.

4. Result and discussion

We carried out the experiment on our method using three datasets: DIKE20, Corel-1k, and Corel-10k. The low-level features of each image in the datasets are extracted and clustered hierarchically. Furthermore, the generated clusters were then validated. On the retrieval process, measurement of precision and recall scores was carried out on the number of retrieval images of 10, 20, 30, 40, 50, and 100.

4.1 Feature extraction

The HSV color space consists of 3 channels, namely hue, saturation, and value. Moments consisting of mean (E), standard deviation (σ), and skewness (S) were extracted from each channel using the CM method. The extraction result consists of 3 feature vectors where one feature vector for each channel. The texture feature extraction using the GLCM method is carried out in four directions, namely 0^0 , 45^0 , 90^0 , and 135^0 , while the pixel distance is 2. Various features can be derived from the GLCM matrix, including dissimilarity, correlation, homogeneity, contrast, ASM, energy, etc. There are four feature vectors formed as the result of extraction, where one feature vector for each direction. On the shape feature, seven moments ($\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6$, and φ_7) are extracted using Hu-moment method. Fig. 6 shows the result of low-level features extraction on an image in the DIKE20 dataset.

4.2 The clustering process

The first step in the clustering process is to form a distance matrix between images based on their features. The distance between images is measured using Euclidean Distance. At the initialization stage of the HACM clustering, each image is represented as a single cluster. Based on the generated distance matrix, the clusters that are close to each other merged. This step is repeated until all images are merged into a single cluster.

The clustering process on the color feature is carried out on each channel. The moment features (mean, standard deviation, and skewness) of each channel in the images are used to form the distance matrix. On the texture features, the clustering process is carried out in four directions (0^0 , 45^0 , 90^0 , and 135^0). The distance matrix between images is formed by performing calculations based on statistical features derived from the GLCM matrix (dissimilarity, correlation, homogeneity, contrast, and energy). On the shape feature, the distance matrix between each image is calculated based on the seven moments ($\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6$, and φ_7). Fig. 7 shows the truncated dendrogram of the color feature (H channel) on the DIKE20 dataset.

4.3 Cluster validation

The validation of the clusters on DIKE20 dataset is carried out by calculating the DBI values. The optimal number of clusters is indicated by the smallest DBI value. The clusters validation on the color feature is carried out on each channel. In the cluster of the H channel, the smallest DBI value is obtained when the number of clusters is 4. In the cluster of the S channel, the smallest DBI value is obtained when the number of clusters is 15. In the cluster of the V channel, the smallest DBI value is obtained when the number of clusters is 2 clusters. Fig. 8 presents the calculation results of the DBI value on the color feature.

The clusters validation on texture feature which was extracted using the GLCM method was carried out in 4 directions. On the 0^0 direction, the smallest DBI value is obtained when the number of clusters is five, while on the 45^0 direction, the smallest DBI value is obtained when the number of clusters is two. On the 90^0 direction, the smallest DBI value is obtained when the number of clusters is three, and on the direction of 135^0 , the smallest DBI value is obtained when the number of clusters is eight. The results of the cluster validation on the texture feature in all directions are presented in Fig. 9. On the shape

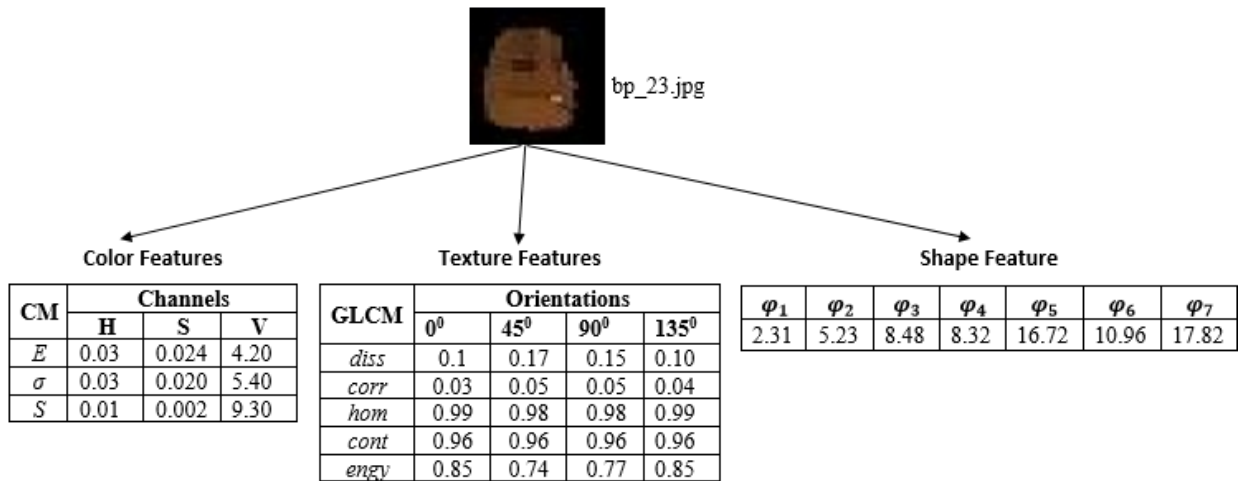


Figure. 6 The process of low-level features extraction

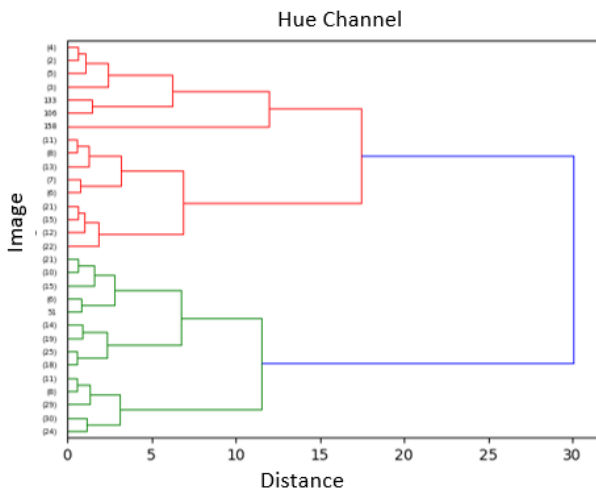


Figure. 7 The truncated dendrogram of the color feature (Hue channel)

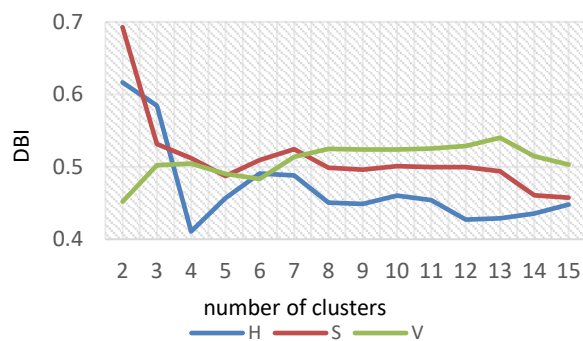


Figure. 8 Clusters validation on color feature

feature, there is only 1 cluster which consists of seven moments which extracted using the Hu-moment method.

The smallest DBI value of 0.73 is obtained when the number of clusters is 3. The DBI value of the shape features for the number of clusters 2 to 15 is shown in Fig. 10.

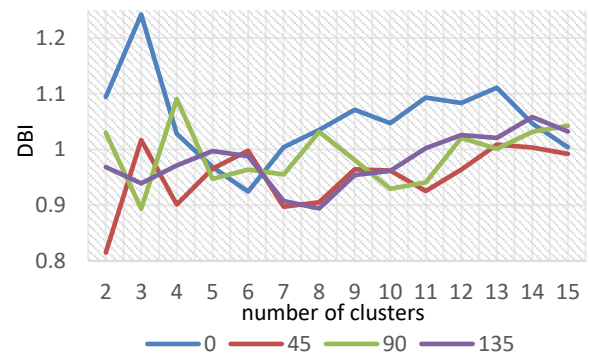


Figure. 9 Clusters validation on texture feature

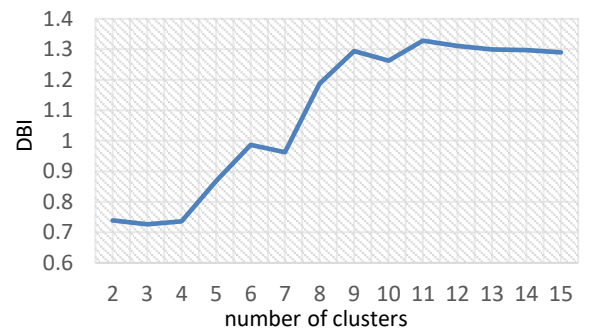


Figure. 10 Clusters validation on shape feature

4.4 The experiment on DIKE20 dataset

The DIKE20 dataset consists of backpack images acquired from various angles. In the experiment using the DIKE20 dataset, the query image (backpack) feature is extracted and then the query process is carried out on each features cluster. The query results in each feature are then normalized and weighted. The retrieval results in the form of backpack images are sorted based on the percentage of similarity with



Figure. 11 Query image Q_{m_1}

Table 2. The combined query results on color feature

No	Image	Distance
1	bp_52.jpg	2.02
2	bp_12.jpg	2.38
3	bp_82.jpg	2.58
4	bp_363.jpg	2.61
5	bp_26.jpg	2.70
6	bp_54.jpg	2.78
7	bp_86.jpg	2.79
8	bp_27.jpg	2.92
9	bp_57.jpg	2.92
10	bp_43.jpg	3.01

Table 3. The combined query results on texture feature

No	Image	Distance
1	bp_79.jpg	1.86
2	bp_56.jpg	1.86
3	bp_51.jpg	1.89
4	bp_154.jpg	1.91
5	bp_100.jpg	1.91
6	bp_11.jpg	1.92
7	bp_309.jpg	1.94
8	bp_2.jpg	1.95
9	bp_82.jpg	1.95
10	bp_86.jpg	1.952

the query image. Fig.11 displays an example of a query image Q_{m_1} used in the experiment. On the color feature, the query results on each channel are then combined using binary OR and then re-sorted.

The distances of combined query results (the first ten images) on color features are presented in Table 2. The clustering on the texture feature is carried out in 4 directions. The query results in each direction are then combined using binary OR and then re-sorted.

Table 3 presents the distance of combined query results (the first ten images) on the texture feature.

The retrieval process on the shape features is done by comparing the shape features of the query image with the centroid of the shape cluster that has been formed. The query results are then sorted in ascending order, where the image with the smallest distance from the query image is at the top position.

Table 4 displays the first ten images of the query results on the shape features.

To make the distance from various query results comparable, then the distance must be normalized in the range of 0-1. Eq. (15) is used to normalize the distance of each image. After the distance of images are normalized, the following process is weighting. The image score on each feature is multiplied by its respective weight and then summed to get the total score.

Table 4. The query result on shape feature

No	Image	Distance
1	bp_257.jpg	0.55
2	bp_67.jpg	0.73
3	bp_119.jpg	0.78
4	bp_55.jpg	0.78
5	bp_19.jpg	0.82
6	bp_223.jpg	0.98
7	bp_161.jpg	0.99
8	bp_305.jpg	1.06
9	bp_1.jpg	1.06
10	bp_22.jpg	1.07

Table 5. The weighted and total score of the images

No	Image	Weighted Score			Total
		Color	Texture	Shape	
1	bp_56.jpg	0.23	0.19	0.32	0.74
2	bp_43.jpg	0.23	0.18	0.32	0.73
3	bp_19.jpg	0.2	0.18	0.35	0.73
4	bp_67.jpg	0.19	0.18	0.36	0.73
5	bp_91.jpg	0.22	0.18	0.33	0.73
6	bp_305.jpg	0.21	0.17	0.34	0.72
7	bp_22.jpg	0.21	0.17	0.34	0.72
8	bp_53.jpg	0.22	0.18	0.32	0.72
9	bp_157.jpg	0.21	0.17	0.33	0.71
10	bp_98.jpg	0.23	0.19	0.32	0.71

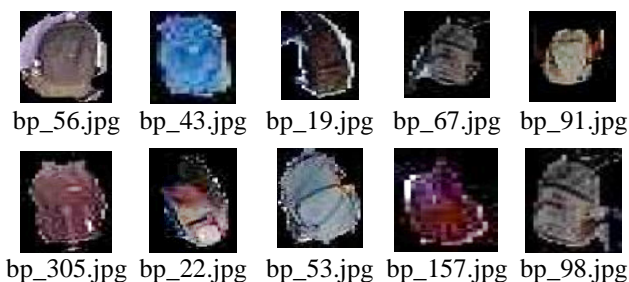


Figure. 12 Image retrieval results for top 10 images retrieved of DIKE20 dataset

The weights for the color, texture, and shape features are 0.3, 0.3, and 0.4, respectively. Table 5 shows the weighted and total scores for each image.

After the weighting process is complete, then the images are sorted based on the total score in descending order. The image scores are then converted into similarity percentages. The top ten retrieval result for the query image Q_{m1} is shown in Fig. 12.

4.5 The experiment on the Corel-1k dataset

We started our experiment on the Corel-1k dataset by selecting an image as the query image from each category. The query image is then inputted into the system to measure precision scores, recall, and retrieval time. Table 6 displays the query image and the top ten most relevant images for the dinosaur, horse, and building categories.

The precision and recall scores measurements were carried out on the number of retrieval images (L) from 10 to 100.

Based on the experimental results, the average precision obtained by the proposed method are 0.88, 0.81, 0.76, 0.71, 0.68, and 0.56 for L of 10, 20, 30, 40, 50, and 100, respectively. The precision scores for each category on the Corel-1k dataset are presented in Table 7.

Table 6. Top 10 image retrieval results on Corel-1k dataset for category dinosaur, horse, and building







No	Query Image	Result
1		
2		
3		

Table 7. The precision score of the proposed method on the Corel-1k dataset

Category	L					
	10	20	30	40	50	100
africa	0.80	0.75	0.70	0.58	0.6	0.36
beach	0.80	0.80	0.77	0.65	0.64	0.51
building	0.80	0.65	0.67	0.68	0.72	0.51
bus	0.70	0.80	0.70	0.68	0.64	0.51
dinosaur	1	0.95	0.87	0.78	0.66	0.68
flower	1	1	1	1	1	0.92
food	0.80	0.70	0.70	0.78	0.78	0.68
elephant	0.90	0.75	0.67	0.60	0.56	0.42
horse	1	0.95	0.90	0.85	0.76	0.77
mountain	1	0.70	0.57	0.48	0.40	0.26
Average	0.88	0.81	0.76	0.71	0.68	0.56

Table 8. The recall score of the proposed method on Corel-1k dataset

Category	L					
	10	20	30	40	50	100
africa	0.08	0.15	0.21	0.23	0.3	0.36
beach	0.08	0.16	0.23	0.26	0.32	0.51
building	0.08	0.13	0.2	0.27	0.36	0.51
bus	0.07	0.16	0.21	0.27	0.32	0.51
dinosaur	0.10	0.19	0.26	0.31	0.33	0.68
flower	0.10	0.20	0.30	0.40	0.50	0.92
food	0.08	0.14	0.21	0.31	0.39	0.68
elephant	0.09	0.15	0.2	0.24	0.28	0.42
horse	0.10	0.19	0.27	0.34	0.38	0.77
mountain	0.10	0.14	0.17	0.19	0.20	0.26
Average	0.09	0.16	0.23	0.29	0.35	0.60

The average recall score obtained by the proposed method in the Corel-1k dataset are 0.09, 0.16, 0.23, 0.29, 0.35, and 0.60 for L of 10, 20, 30, 40, 50, and 100 respectively. Recall scores for each category on the Corel-1k dataset are presented in Table 8.

In addition, the average retrieval time obtained by the proposed method on the Corel-1k dataset is 0.036s.

4.6 The experiment on the Corel-10k dataset

On the experiment on the Corel-10k dataset, the average precision and recall scores were also measured on the number of retrieval L images of 10, 20, 30, 40, 50, and 100. The average precision score obtained by the proposed method was 0.64, 0.62, 0.57, 0.53, 0.50, and 0.40 for L of 10, 20, 30, 40, 50, and 100, respectively. Furthermore, the average recall score are 0.06, 0.11, 0.15, 0.19, 0.22, and 0.35 for L of 10, 20, 30, 40, 50, and 100, respectively. Table 9

Table 9 The average precision and recall score on Corel-10k dataset

Performance	<i>L</i>					
	10	20	30	40	50	100
Precision	0.64	0.62	0.57	0.53	0.50	0.40
Recall	0.06	0.11	0.15	0.19	0.22	0.35

Table 10. Performance comparison on Corel-1k dataset

No	Method	Prec	Rec
1	Dawood et al., [13]	0.79	0.14
2	Xie et al., [14]	0.69	0.14
3	Xiaobo et al., [15]	0.68	-
4	<i>Proposed Method</i>	0.81	0.16

Table 11. Performance comparison on Corel-10k dataset

No	Method	Prec	Rec
1	Thusnavis and Vasuki, [1]	0.56	-
2	Pradhan et al., [16]	0.57	0.11
3	Hussain et al., [17]	0.46	0.09
4	<i>Proposed Method</i>	0.62	0.11

presents the average precision and recall scores obtained by the proposed method on the Corel-10k dataset.

The average retrieval time obtained by the proposed method on the Corel-10k dataset is 0.09s.

5. Benchmarking

The comparison of the average precision and recall scores obtained by the proposed method with the state-of-art method was carried out at $L=20$. On the Corel-1k dataset, the proposed method obtained the best precision score of 0.81 when compared with [13-15] obtained precision scores of 0.79, 0.69, and 0.68, respectively. Meanwhile, the recall score obtained by the proposed increased by 2% compared to the recall score obtained by [13, 14]. Table 10 shows the performance comparison between the proposed CBIR method and other state-of-the-art methods on the Corel-1k dataset.

On the Corel-10k dataset, the proposed method increased the precision score by 5% compared to the score obtained by [16]. Meanwhile, when compared with [1, 17], the precision scores obtained by the proposed method increased by 6% and 16%, respectively. The recall score obtained by proposed is 0.11, the same score also obtained by [16]. Table 11 compares each method's precision and recall scores on the Corel-10k dataset.

6. Conclusion and future work

The proposed method shows better performance compared to the state-of-art methods based on the experiment on the Corel-1k and Corel-1k datasets. On the Corel-1k dataset, the average precision score is 0.81, increased by 2% compared with the precision score obtained by [13]. Meanwhile, the proposed method also obtained the best recall score, which is 0.16. On the Corel-10k dataset, the average precision score is 0.62, increased by 5% compared to the precision score obtained by [16]. In addition, the recall score obtained by the proposed method is 0.11, where the same score was also obtained by [16].

Future work is possible by selecting and combining appropriate features to improve score precision and recall. In addition, on images with complex and varied backgrounds, feature extraction can be performed only on salient objects.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Conceptualization and methodology, Rahmad Hidayat, Agus Harjoko, and Aina Musdholifah; software, Rahmad Hidayat; validation, resources, and investigation, Rahmad Hidayat and Agus Harjoko; formal analysis, Rahmad Hidayat, Agus Harjoko, and Aina Musdholifah; writing—original draft preparation, Rahmad Hidayat; writing—review and editing, Agus Harjoko and Aina Musdholifah; visualization, Rahmad Hidayat, Agus Harjoko, and Aina Musdholifah.

Acknowledgments

This research was supported by the Doctoral grant of the Indonesia endowment fund for education (LPDP) ministry of Finance Republic of Indonesia (No. PRJ-4809/LPDP.3/2016).

References

- [1] M. I. Thusnavis and A. Vasuki, "An efficient image retrieval framework using fused information feature R", *Comput. Electr. Eng.*, Vol. 75, pp. 46-60, 2019.
- [2] M. Jain and S. K. Singh, "An Efficient Content Based Image Retrieval Algorithm Using Clustering Techniques For Large Dataset", *Int. Conf. Comput. Commun. Autom.*, Vol. 4, pp. 1-5, 2018.
- [3] D. Saravanan, S. V. Lakshmi, and D. Joseph, "Image retrieval by image feature using data

- mining technique”, *Proc. Int. Conf. Inven. Syst. Control. ICISC 2017*, pp. 4-7, 2017.
- [4] X. Y. Wang, Y. J. Yu, and H. Y. Yang, “An effective image retrieval scheme using color, texture and shape features”, *Comput. Stand. Interfaces*, Vol. 33, No. 1, pp. 59-68, 2011.
- [5] J. Pradhan, S. Kumar, A. K. Pal, and H. Banka, “A hierarchical CBIR framework using adaptive tetrolet transform and novel histograms from color and shape features”, *Digit. Signal Process.*, Vol. 82, pp. 258-281, 2018.
- [6] D. Zhang and G. Lu, “Review of shape representation and description techniques”, *Pattern Recognit.*, Vol. 37, No. 1, pp. 1-19, 2004.
- [7] J. Yue, Z. Li, L. Liu, and Z. Fu, “Content-based image retrieval using color and texture fused features”, *Math. Comput. Model.*, Vol. 54, No. 3-4, pp. 1121-1127, 2011.
- [8] M. K. Alsmadi, “Content-Based Image Retrieval Using Color, Shape and Texture Descriptors and Features”, *Arab. J. Sci. Eng.*, Vol. 45, No. 4, pp. 3317-3330, 2020.
- [9] W. Dai, Z. Zhu, and F. Wu, “Image Clustering Algorithm and Its Application in Human Resources Management in Colleges”, *IEEE Access*, Vol. XX, pp. 1-8, 2020.
- [10] R. A. Stefan, I. Szöke, and S. Holban, “Hierarchical clustering techniques and classification applied in Content Based Image Retrieval (CBIR)”, *Int. Symp. Appl. Comput. Intell. Informatics*, Vol. 10, No. 1, pp. 147-152, 2015.
- [11] R. A. Stefan, I. Szöke, and S. Holban, “Hierarchical clustering techniques and classification applied in Content Based Image Retrieval (CBIR)”, *Int. Symp. Appl. Comput. Intell. Informatics*, Vol. 10, pp. 147-152, 2015.
- [12] M. A. Mercioni and S. Holban, “A study on Hierarchical Clustering and the Distance metrics for Identifying Architectural Styles”, In: *Proc. of 2019 Int. Conf. ENERGY Environ.*, pp. 49-53, 2019.
- [13] H. Dawood, M. H. Alkinani, A. Raza, H. Dawood, R. Mehboob, and S. Shabbir, “Correlated microstructure descriptor for image retrieval”, *IEEE Access*, Vol. 7, pp. 55206-55228, 2019.
- [14] G. Xie, B. Guo, Z. Huang, Y. Zheng, and Y. Yan, “Combination of Dominant Color Descriptor and Hu Moments in Consistent Zone for Content Based Image Retrieval”, *IEEE Access*, Vol. 8, pp. 146284-146299, 2020.
- [15] Z. Xiaobo, P. Jinye, L. Tian, and A. Zhigang, “Image Retrieval Method Based on Improved Local Binary Pattern”, *2021 IEEE 3rd Int. Conf. Commun. Inf. Syst. Comput. Eng. CISCE 2021*, pp. 256-260, 2021.
- [16] J. Pradhan, A. K. Pal, H. Banka, and P. Dansena, “Fusion of region based extracted features for instance- and class-based CBIR applications”, *Appl. Soft Comput. J.*, Vol. 102, p. 107063, 2021.
- [17] S. Hussain, M. Ahmad, and W. Arshad, “Additive deep feature optimization for semantic image retrieval”, *Expert Syst. Appl.*, Vol. 170, No. January, p. 114545, 2021.
- [18] R. Haralick, K. Shanmugan, and I. Dinstein, “Textural features for image classification”, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 3, pp. 610-621, 1973.
- [19] N. Pullaiah, D. Venkatasekhar, P. Venkatramana, and B. Sudhakar, “Detection of Breast Cancer on Magnetic Resonance Imaging Using Hybrid Feature Extraction and Deep Neural Network Techniques”, *Int. J. Intell. Eng. Syst.*, Vol. 13, No. 6, pp. 229-240, 2020, doi: 10.22266/ijies2020.1231.21.
- [20] S. Sun, Y. Sun, G. Xu, L. Zhang, Y. Hu, and P. Liu, “Partial Discharge Pattern Recognition of Transformers Based on the Gray-Level Co-Occurrence Matrix of Optimal Parameters”, *IEEE Access*, Vol. 9, pp. 102422-102432, 2021.
- [21] M. B. Agbaje, A. E. Ezugwu, and R. Els, “Automatic data clustering using hybrid firefly particle swarm optimization algorithm”, *IEEE Access*, Vol. 7, pp. 184963-184984, 2019.
- [22] D. L. Davies and D. W. Bouldin, “A Cluster Separation Measure”, *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. PAMI-1, No. 2, pp. 224-227, 1979.
- [23] B. Verma and S. Kulkarni, “A fuzzy-neural approach for interpretation and fusion of colour and texture features for CBIR systems”, *Appl. Soft Comput. J.*, Vol. 5, No. 1, pp. 119-130, 2004.
- [24] Y. Liu, Y. Dong, H. Liang, F. Chiclana, and E. H. Viedma, “Multiple Attribute Strategic Weight Manipulation with Minimum Cost in a Group Decision Making Context with Interval Attribute Weights Information”, *IEEE Trans. Syst. Man, Cybern. Syst.*, Vol. 49, No. 10, pp. 1981-1992, 2019.
- [25] S. Wu, “Linear combination of component results in information retrieval”, *Data Knowl. Eng.*, Vol. 71, No. 1, pp. 114-126, 2012.
- [26] Jumi, A. Harjoko, and A. Ashari, “Content-Based Image Retrieval For Asset Management Based On Weighted Feature And K-Means Clustering”, *J. Theor. Appl. Inf. Technol.*, Vol. 77, No. 1, pp. 116-124, 2015.
- [27] D. M. W. Powers, “Evaluation: From Precision, Recall and F-Factor to ROC, Informedness,

Markedness & Correlation David”, *J. Mach. Learn. Technol.*, Vol. 2, No. December, pp. 37-63, 2007.

- [28] A. Ahmed, “Implementing Relevance Feedback for Content-Based Medical Image Retrieval”, *IEEE Access*, Vol. 8, pp. 79969-79976, 2020.