

Article

# Content-Based Color Image Retrieval Using Block Truncation Coding Based on Binary Ant Colony Optimization

Yan-Hong Chen <sup>1</sup>, Chin-Chen Chang <sup>2</sup>, Chia-Chen Lin <sup>3,\*</sup>  and Cheng-Yi Hsu <sup>2</sup>

<sup>1</sup> School of Information, Zhejiang University of Finance & Economics, Zhejiang 310018, China; zufe\_graphics@163.com

<sup>2</sup> Department of Computer Science and Information Engineering, Feng Chia University, Taichung 40724, Taiwan; alan3c@gmail.com (C.-C.C.); pcs1596321@yahoo.com.tw (C.-Y.H.)

<sup>3</sup> Department of Information Engineering and Computer Science, Providence University, Taichung 43301, Taiwan

\* Correspondence: mhlin3@pu.edu.tw

Received: 2 December 2018; Accepted: 24 December 2018; Published: 27 December 2018



**Abstract:** In this paper, we propose a content-based image retrieval (CBIR) approach using color and texture features extracted from block truncation coding based on binary ant colony optimization (BACOBTC). First, we present a near-optimized common bitmap scheme for BTC. Then, we convert the image to two color quantizers and a bitmap image-utilizing BACOBTC. Subsequently, the color and texture features, i.e., the color histogram feature (CHF) and the bit pattern histogram feature (BHF) are extracted to measure the similarity between a query image and the target image in the database and retrieve the desired image. The performance of the proposed approach was compared with several former image-retrieval schemes. The results were evaluated in terms of Precision-Recall and Average Retrieval Rate, and they showed that our approach outperformed the referenced approaches.

**Keywords:** contest-based image retrieval; block truncation coding; ant colony optimization; vector quantization; feature descriptor

## 1. Introduction

Block Truncation Coding (BTC) is a classic technique for image compression [1]. It divides the original images into small, non-overlapped image blocks, and each block is simply represented by two mean values and a bitmap image. Because of the limitations of the traditional BTC method, some improved BTC methods have been proposed to improve the image quality and the compression ratio [2]. Yang et al. [3] proposed a near-optimum, accelerated BTC algorithm based on a truncated K-means algorithm that drastically reduced the computational complexity by utilizing the image inter-block correlation. An improved BTC scheme based on a set of pre-defined bit planes has also been proposed [4]. In this method, a Huffman compress algorithm is used to reduce the bit rate. Guo et al. proposed the Ordered Dither Block Truncation Coding [5], Error Diffusion Block Truncation Coding [6], dot-diffused BTC [7], and Direct Binary Search BTC [8] to improve the quality of images.

Traditional BTC and improved BTC have achieved outstanding performances in various fields, including data hiding and content-based image retrieval (CBIR), especially in CBIR. In recent years, with the rapid development of social media and the explosive growth of image scale, the image retrieval scheme based on a compressed domain has been paid more and more attention; it extracts features directly from the compressed stream, rather than extracting features from the original image as in the classical method. This type of retrieval method reduces the time of feature extraction. As a classic technique for image compression, many researchers have presented numerous new CBIR-based

BTC approaches to improve the performance of image retrieval. Guo et al. [9–12] presented several approaches to retrieve color images utilizing the features extracted from improved block truncation coding. Wang et al. [13] described the multi-factor correlation of images through gradient value, structural correlation and gradient direction, utilizing block truncation coding. Wang et al. [14] focused on a novel image retrieval algorithm combined with a global statistical feature and a local binary bitmap feature. Keke et al. [15] discussed a novel image retrieval technique, which is based on the slope magnitude method with block truncation coding and the shape features.

In this paper, we introduce an approach for retrieving images that enhances the performance of the CBIR using color and texture features extracted based on binary ant colony optimization (BACOBTC). First, an RGB color space image is converted to two quantizers and a bitmap image utilizing BACOBTC. Subsequently, two image features, i.e., the color histogram feature and the bit pattern histogram feature, are extracted from the quantizers and bitmap, respectively. Then, the similarity measure is performed between each target image vector and the query vector. The experimental results showed that the proposed scheme has a higher recall ratio and retrieval precision.

The remainder of this paper is organized as follows. In Section 2, the CBIR framework is introduced briefly, and the proposed BACOBTC algorithm is presented in Section 3. Feature extraction is performed in Section 4, and extensive experimental results are reported in Section 5. Our conclusions are presented in Section 5.6.

## 2. Content-Based Image Retrieval Framework

At present, most popular search engines use traditional text-based methods to provide image-retrieval services, which depend mainly on the keyword associated with the images, and these keywords require manual annotation [16]. When a query is submitted to the search engine, it could be matched with indexed terms associated with images in the database. The process of manual image annotation is very time-consuming and imprecise. If the tagged keyword is inappropriate, it is difficult for this method to obtain accurate search results. Therefore, content-based image retrieval is proposed. In CBIR, images can be retrieved by visual features, which can be extracted directly from image databases such as color, texture and shape. Many image retrieval schemes based on visual features have been proposed in the last few years [17–25]. Literatures [17–25] are discussed in Table 1.

**Table 1.** Introduction of Literatures [17–25].

Reference	Feature and Technique
S. Silakari et al. [17]	Block truncation coding, using K-Means clustering algorithm
F. X. Yu et al. [18]	Block truncation coding, vector quantization, colour pattern co-occurrence matrix.
C. H. Lin [19]	Color co-occurrence matrix.
P. Poursistani et al. [20]	Vector quantization, K-means clustering, create an effective histogram from DCT coefficients.
X. Wang and Z. Wang [21]	Image structure elements' histogram, HSV color space, integrates the advantages of both statistical and structural texture description methods.
G. H. Liu et al. [22]	The correlation between each pair of database images based on a query log and inferring the relevance.
B. S. Manjunath et al. [23]	Bayesian classifier, Feature extracted in DCT, Image semantic classification. Based on feature extracted in DCT and semantic classification by Bayes.
G. H. Liu et al. [24]	The micro-structure descriptor, HSV color space.
C. H. Lin et al. [25]	K-means algorithm, color spatial distribution features, clustering-based filter.
Biasotti S. et al. [26]	geometric shape or texture
Masoumi M. et al. [27]	The spectral graph wavelet approach for 3D shape retrieval using the bag-of-features paradigm.

In the CBIR system, a query image will be submitted to the system by a user, where the query image and all other images in the database will be processed and represented in the same manner so that relevant images can be retrieved. Then, the available visual features will be extracted and characterized in a certain manner into the data space. Finally, the distance between the query image's transformed features and those of all other images in the database will be computed by the system so that the most relevant images according to some distance measures will be returned [16].

### 3. BACOBTC for Color Image

The proposed BACOBTC [28] method introduces a BTC technique based on binary ant colony optimization (BACO) in order to get images that have good visual quality. The ant colony optimization (ACO) algorithm is a probabilistic technique for solving computational problems that can be reduced to finding good paths through graphs [29]. Following the idea of ACO, the BACO algorithm performs competitively in solving the discrete-domain problems because of its unique space structure of random binary search. The BACO's solution is represented by a binary bit string with each node selected from two possible values, i.e., 0 or 1 [29]. Considering that each position of the common bitmap has an independent effect on the distortion of a block, instead of assigning an ant to search all the positions, we assigned an ant for each position to reduce the time required to determine the proper common bitmap. For a common bitmap, each position has only two choices, 0 or 1. Each ant is initialized to 0 or 1 randomly and an initial common bitmap is generated when all the ants have been initialized. Then select an ant each time and change its value to the opposite. The pheromone of this ant for choosing 0 or 1 is changed by computing the evaluation value for the corresponding block. By comparing the updated pheromone value of this ant, the choice of 0 or 1 will be determined. When the pheromone of all the ants are updated, the near-optimal solution will be generated according to the updated pheromone matrices. Although the initialization of the binary matrix is random, in each loop we only reverse the value of one single position, thus, it will not affect the final optimal results. This process is detailed as follows:

Step 1. Input a color image that is to be divided into non-overlapping  $m \times n$  blocks, and the color image generates a bitmap  $m_{ij}$  and two mean values,  $x_H$  and  $x_L$ , which are defined in Equation (1):

$$\begin{cases} u = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n x_{ij}, \\ x_H = \frac{1}{n_H} \sum x_{ij}, \text{ if } x_{ij} \geq u, \\ x_L = \frac{1}{n_L} \sum x_{ij}, \text{ otherwise,} \end{cases} \quad (1)$$

where  $u$  is the block's mean value and  $x_{ij}$  is the pixel value located at the position  $(i, j)$  in each block.

Step 2. Generate three pairs of quantization values, i.e.,  $(x_{RH}, x_{RL})$ ,  $(x_{GH}, x_{GL})$ ,  $(x_{BH}, x_{BL})$  and a random binary matrix  $C = \{C_{ij} | C_{ij} \in \{0, 1\}, 1 \leq i \leq m, 1 \leq j \leq n\}$  as an initial common bitmap. Calculate the initial MSE using Equation (2):

$$MSE = \sum_{c_{ij}=0} (x_{ij} - x_L)^2 + \sum_{c_{ij}=1} (x_{ij} - x_H)^2, \quad (2)$$

where  $x_{ij} = (R_{ij}, G_{ij}, B_{ij})$  are the original values from the three channels at the same position.

Step 3. Obtain the image  $Z$  by replacing the pixel tagged as 1 with  $x_H$  and 0 with  $x_L$ . Then, select a value of matrix  $C$  and change it into the opposite value to generate a temporary matrix. Reconstruct image  $Z$  by replacing the pixel tagged as 1 with  $x'_H$  and 0 with  $x'_L$  and get the image  $Z'$ .

Step 4. Compute the corresponding  $newMSE_{ij}$  for this matrix by using Equation (2). Generate two pheromone matrices,  $\tau_{ij}(0)$  and  $\tau_{ij}(1)$ , and each component in these matrices is selected as  $\tau_{ij}(k) = 0.5 (k \in \{0, 1\})$  because each ant located in matrix  $C$  has the same probability of selecting path "0" or path "1" in the initial state.

Step 5. Compute the incremental pheromone  $\Delta\tau_{ij}(k)$  according to Equation (3), and update the initial pheromone matrices  $\tau_{ij}(0)$  and  $\tau_{ij}(1)$  using Equation (4):

$$\Delta\tau_{ij}(k) = \begin{cases} \frac{1}{\text{MSE}} & \text{if } C_{ij} = k, \\ \frac{1}{\text{newMSE}_{ij}} & \text{otherwise,} \end{cases} \quad (3)$$

where  $k \in \{0, 1\}, i \in [1, m], j \in [1, n]$ .

$$\tau_{ij}(k) = \tau_{ij}(k) + \Delta\tau_{ij}(k) \quad (4)$$

where  $k \in \{0, 1\}, i \in [1, m], j \in [1, n]$ .

Step 6. Update the common bitmap according to the following rule:

$$C_{ij}^* = \begin{cases} 1 & \text{if } \tau_{ij}(1) > \tau_{ij}(0) \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where  $i \in [1, m], j \in [1, n]$ .

Step 7. Repeat Steps 3 through 6 until all of the values in matrix C are dealt with. Similarly, all of the image blocks are dealt with in the same way. To this point, the near-optimal common bitmap  $C^*$  for the entire color image has been generated.

An example for BACOBTC algorithm procedures is shown in Figure 1.

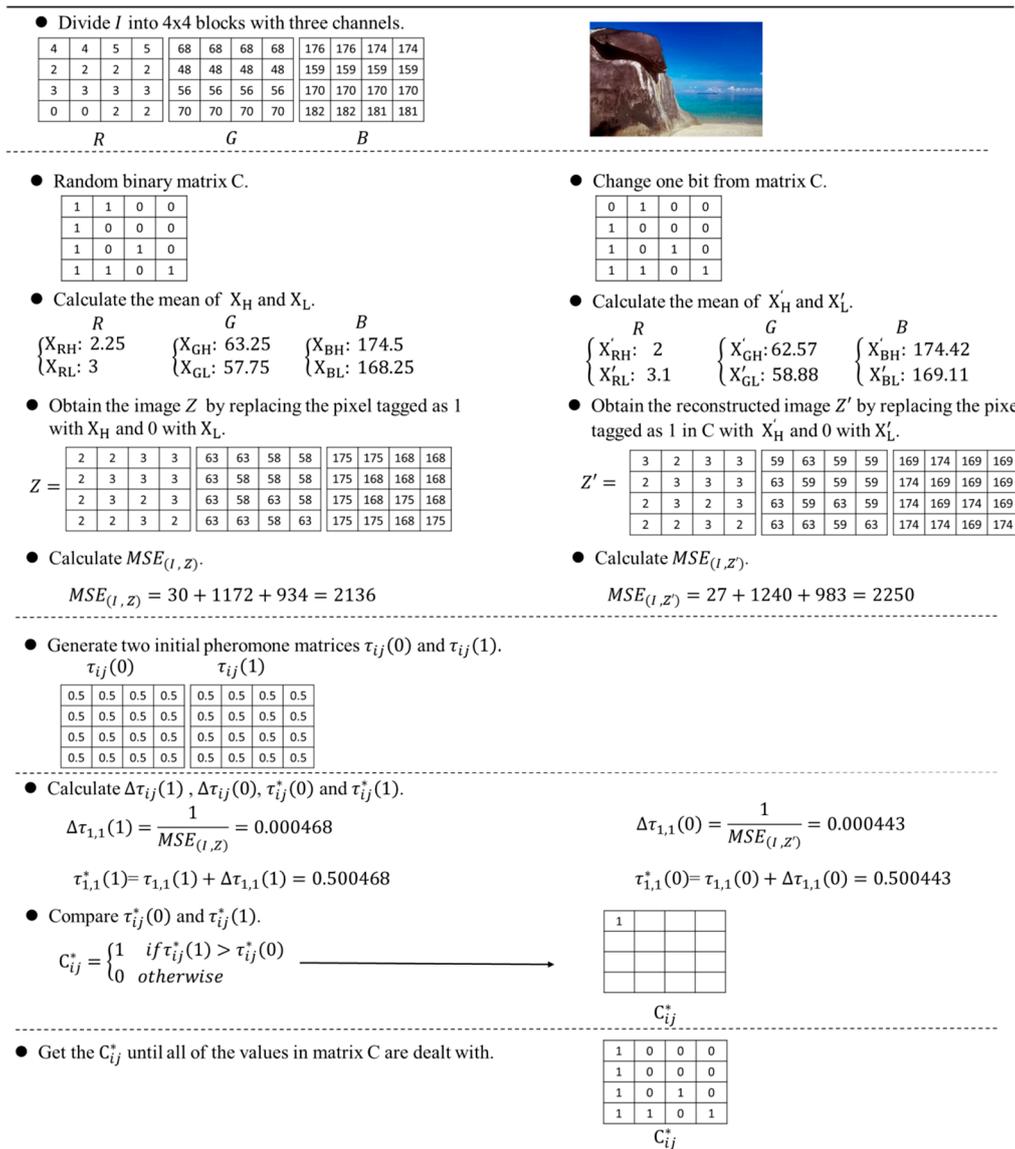


Figure 1. An example of BACOBTC algorithm procedures.

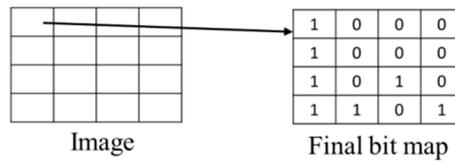
### 4. Feature Representation

In this paper, color histogram feature (CHF) and pattern histogram feature (BHF) are used to represent the content of the image. The CHF is derived from the two color quantizers (max quantizer and min quantizer), and the BHF is simply constructed from the BACOBTC bitmap image [9]. A detailed description of how to acquire these two features is introduced below.

#### 4.1. Color Histogram Feature

1. Generate BACOBTC Max Quantizer And Min Quantizer. For each color block, the quantization values, including  $x_{RH}$ ,  $x_{RL}$ ,  $x_{GH}$ ,  $x_{GL}$ ,  $x_{BH}$ , and  $x_{BL}$  are computed. Then,  $x_{RH}$  is mapped to the red max quantizer table, and  $x_{RL}$  is mapped to the red min quantizer table. Blue and green channels could do the same operations. The analyses of the min quantizer and the max quantizer are shown in Figure 2.

- Calculate the  $X_H$  and  $X_L$ .



$$\begin{matrix} R & G & B \\ \{X_{RH}: 2 & \{X_{GH}: 63 & \{X_{BH}: 174 \\ \{X_{RL}: 3 & \{X_{GL}: 59 & \{X_{BL}: 169 \end{matrix}$$

- Collect the  $X_{RH}, X_{GH}, X_{BH}$  in Max\_table and  $X_{RL}, X_{GL}, X_{BL}$  in Min\_table.

Max\_table

R				G				B			
2	7	31	56	63	72	71	51	174	182	192	187
42	99	84	109	96	131	77	31	213	183	172	47
45	53	76	97	75	79	78	77	31	90	14	113
182	142	56	42	82	67	61	64	113	114	104	103

Min\_table

R				G				B			
3	2	2	7	59	72	75	79	169	181	186	168
76	56	22	28	96	131	77	37	143	131	75	36
42	67	83	68	11	10	16	21	11	10	20	73
78	71	75	68	75	79	77	82	78	77	79	83

Figure 2. Example of min quantizer and max quantizer calculation.

2. The color codebook  $C$  is generated by vector quantization (VQ). In this study, the Linde–Buzo–Gary vector quantization (LBGVQ) was utilized to perform color clustering for a given training set to obtain the color codebook,  $C = \{c_1, c_2, \dots, c_{N_c}\}$ , where  $c_{N_c}$  denotes the codeword from the quantizer.

3. Obtain the color index table as shown in Equation (6):

$$\begin{cases} y_{\max}(i, j) = \operatorname{argmin} \|q_{\max}(i, j), c^{\max}\|_2^2 \\ y_{\min}(i, j) = \operatorname{argmin} \|q_{\min}(i, j), c^{\min}\|_2^2 \end{cases} \quad (6)$$

For  $k = 1, 2, \dots, N_c, i = 1, 2, \dots, M/m$ , and  $j = 1, 2, \dots, N/n$  where  $y_{\max}(i, j)$  denotes the optimal matching distance between  $q_{\max}(i, j)$  and  $c^{\max}$ , and the symbols  $i$  and  $j$  represent the indices of the image block. Herein,  $q_{\max}(i, j)$  denotes the R, G and B values of the max quantizer of each image block, and  $c^{\max}$  denotes the codeword. Similarly,  $y_{\min}(i, j)$  denotes the optimal matching distance between  $q_{\min}(i, j)$  and  $c^{\min}$  in the min quantizer.

4. Compute  $CHF_{\max}$  and  $CHF_{\min}$  using Equation (7):

$$\begin{cases} CHF_{\max}(k) = \Pr\left\{y_{\max}(i, j) = k \mid i = 1, 2, \dots, \frac{M}{m}\right\} \\ CHF_{\min}(k) = \Pr\left\{y_{\min}(i, j) = k \mid i = 1, 2, \dots, \frac{M}{m}\right\} \end{cases} \quad (7)$$

$\Pr\{\cdot\}$  denotes the probability of the occurrence of certain color codewords' appearing in an image. An example of the CHF calculation is shown in Figure 3a.

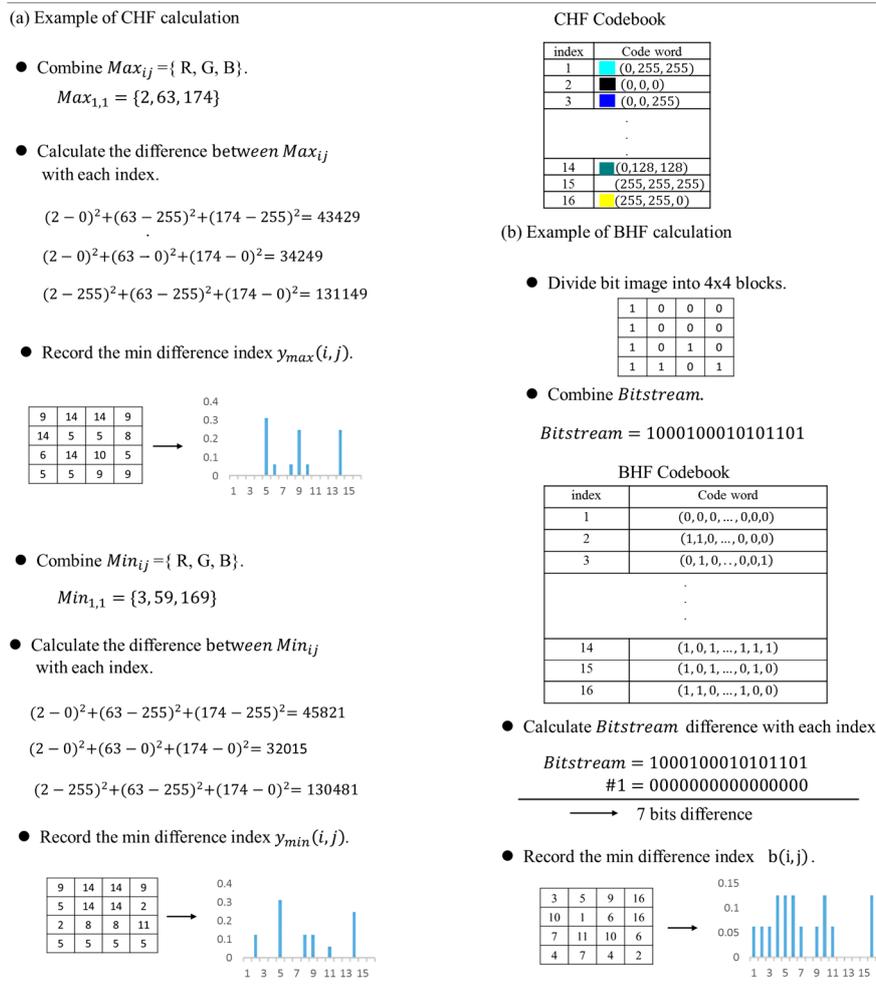


Figure 3. Example of CHF and BHF calculation.

#### 4.2. Bit Pattern Histogram Feature

The bit pattern histogram feature can identify the texture and brightness of the image effectively. Figure 3b shows an example of the BHF computation. Similar to the CHF calculation, the procedures for computing BHF are described below.

1. The bit pattern histogram feature can identify the texture and brightness of the image effectively. Figure 2 shows an example of the BHF computation. Similar to the CHF calculation, the procedures for computing BHF are described below:

2. The bit image is generated from the BACOBTC encoding process. Then, the bit pattern codebook  $B = \{b_1, b_2, \dots, b_{N_b}\}$  is defined as the bit pattern codebook comprised of  $N_b$  bit codewords.

3. Obtain the bitmap image index table as shown in Equation (8):

$$b(i, j) = \arg \min \varphi_H \{bm(i, j), B_q\} \tag{8}$$

where  $q = 1, 2, \dots, N_b$ ,  $\varphi_H\{.,.\}$  denotes the Hamming distance between the bitmap image  $bm(i, j)$ , and  $B_q$  denotes the bit pattern's codewords.

4. Compute BFH using Equation (9):

$$BHF(k) = pr \left\{ b(i, j) = k \mid i = 1, 2, \dots, \frac{M}{m}; j = 1, 2, \dots, \frac{N}{n} \right\} \tag{9}$$

## 5. Experiments and Results

This section demonstrates the performance of the proposed image retrieval scheme using two Corel datasets and one Caltech101 dataset. In order to evaluate the performance of the retrieval module, some experiments were conducted to compare the proposed algorithm with several image-retrieval approaches.

### 5.1. Image Databases

Caltech 101 is a digital image data set created in September 2003, and it contains 9146 images in total, which are divided into 101 distinct object categories (for instance, faces, watches, ants, pianos, etc.) and a background category [30]. The Corel image database is the most commonly used database to test the performance of image retrieval. In our experiments, we used two Corel subsets, i.e., Corel 1000 and Corel 10000 [31]. The Corel 1000 database consisted of over 1000 images in 10 categories, with each category containing 100 images. Every image in the database had the sizes of  $256 \times 384$  or  $384 \times 256$ . The Corel 10000 database consisted of 100 categories including diverse content such as beach, car, door, sunset fish, and so on.

### 5.2. Distance Metric

The similarity distance computation is one of the important steps in retrieving images. We can judge whether the two images are similar by the similarity distance. If the three feature vectors of the two images are approximate, the compared images are similar. By the analysis in Section 3, the image is converted to three feature vectors, i.e.,  $\text{CHF}_{\max}$ ,  $\text{CHF}_{\min}$ , and BHF. Assuming that the size of the bit pattern codebook B is 64 and that the size of the color codebook C is 64, then  $\text{CHF}_{\max} = \{a_1, a_2, \dots, a_{64}\}$ ,  $\text{CHF}_{\min} = \{b_1, b_2, \dots, b_{64}\}$ , and  $\text{BHF} = \{c_1, c_2, \dots, c_{64}\}$ . Let the query image's feature  $V = \{\text{CHF}_{\max}, \text{CHF}_{\min}, \text{BHF}\}$  and the target image's feature in database  $V' = \{\text{CHF}'_{\max}, \text{CHF}'_{\min}, \text{BHF}'\}$ . The similarity distance of the query image and the target image is described in Equation (10):

$$\text{Dis}(query, target) = \rho_1 \sum_{k=1}^{N_c} \frac{|\text{CHF}_{\min}^q(k) - \text{CHF}_{\min}^t(k)|}{\text{CHF}_{\min}^q(k) + \text{CHF}_{\min}^t(k) + \epsilon} + \rho_2 \sum_{k=1}^{N_c} \frac{|\text{CHF}_{\max}^q(k) - \text{CHF}_{\max}^t(k)|}{\text{CHF}_{\max}^q(k) + \text{CHF}_{\max}^t(k) + \epsilon} + \rho_3 \sum_{k=1}^{N_b} \frac{|\text{BHF}^q(k) - \text{BHF}^t(k)|}{\text{BHF}^q(k) + \text{BHF}^t(k) + \epsilon} \quad (10)$$

where  $\rho_1$ ,  $\rho_2$  and  $\rho_3$  denote the similarity weighting constants, and  $\rho_1 + \rho_2 + \rho_3 = 1$ , which represents the percentage contributions of  $\text{CHF}_{\max}$ ,  $\text{CHF}_{\min}$ , and BHF in the proposed image-retrieval process. To avoid mathematical errors in division, a small number,  $\epsilon$ , is added to the denominator.  $\text{CHF}^q$  and  $\text{CHF}^t$  represent the query and target images of the color feature descriptors, and  $\text{BHF}^q$  and  $\text{BHF}^t$  denote the query and target images of the descriptors of the bit pattern features.

### 5.3. Performance Metrics

Precision-Recall curves, as the commonly used performance measure, were used to measure the effectiveness of the proposed method. Precision P and recall R are defined below:

$$P(q) = I_q / N, \quad (11)$$

$$R(q) = I_q / M, \quad (12)$$

where M, N and  $I_q$  denote the number of similar images in the entire database that are similar to the query, the total number of images retrieved and the number of retrieved relevant images. In addition to the Precision-Recall curves, the average precision rate, APR, was used to measure the retrieval performance. The definition of the average precision rate is shown in Equation (13):

$$\text{APR} = \frac{1}{|N_t|} \sum_1^{|N_t|} P(q), \quad (13)$$

where  $N_t$  is the total number of images in the database.

5.4. Retrieval Performance

Figure 4 shows the precision recall values of the proposed BACOBTC feature descriptor under Corel 1000 database. In this experiment, in order to obtain feature descriptors, all images in the database are encoded into various image block sizes, i.e.,  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ , and  $32 \times 32$ . In addition, the color and bit pattern codebook sizes were determined as  $N_c = N_b = 64$ . We compared the influence of different values on  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$  in Figure 1 ( $\rho_i = \{0, 1\}, i = 1, 2, 3$ ). The red, blue and green lines represent different values of  $\rho_1$ ,  $\rho_2$ , and  $\rho_3$ , respectively, i.e., different combinations of the three characteristic vectors,  $CHF_{max}$ ,  $CHF_{min}$ , and BHF. Obviously, the red line achieved the best result over all image block sizes, and this was due to the effect of the combined action of the three characteristic vectors. The common use of the three feature vectors produced the highest precision recall rate, because the color distribution and the edge information can be determined. Figure 1 shows that the precision became lower as the block size increased. This was because the extracted reference factors became less when the block size increased.

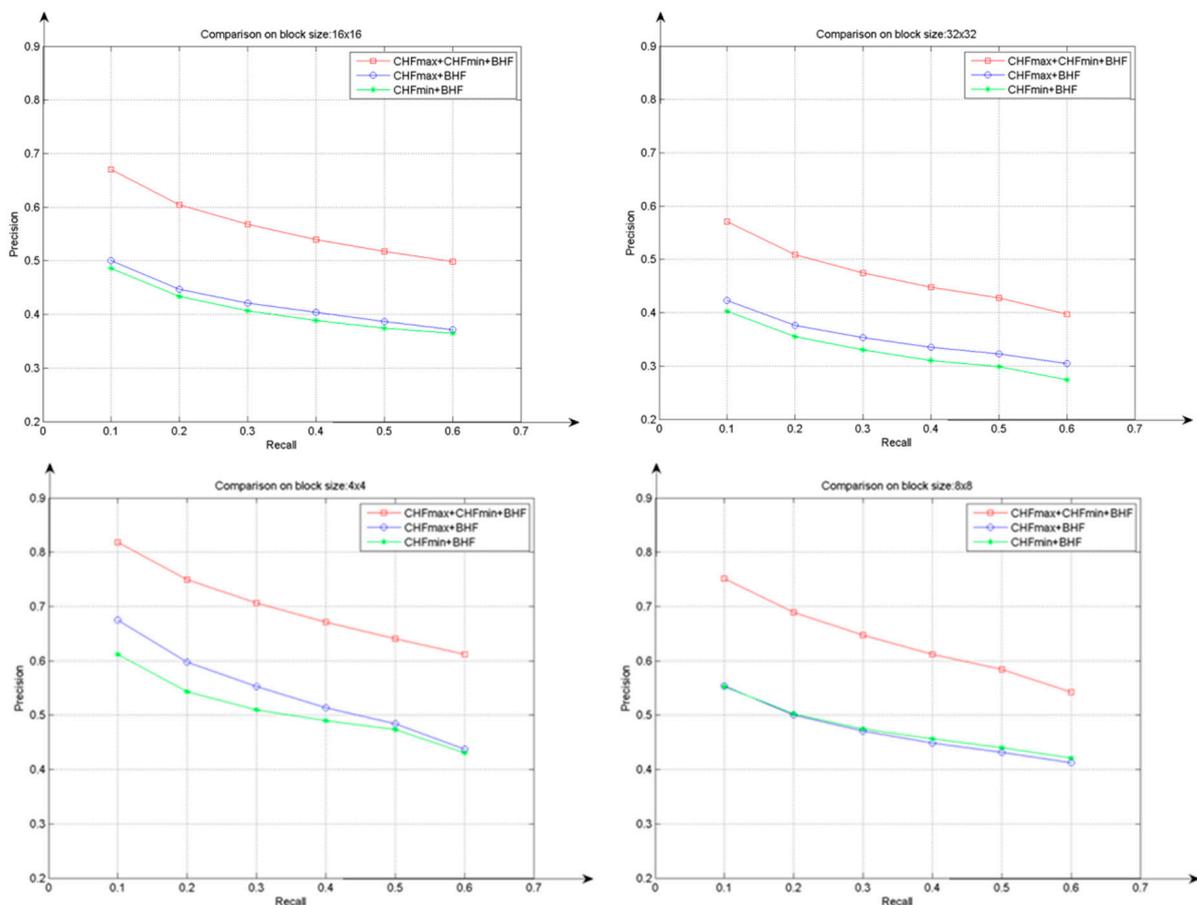


Figure 4. Effectiveness of the proposed feature under Corel 1000 database.

Figure 5 shows the 10 examples of image-retrieval results. The image at the top of the left-top corner is the query image, and the rest are the 10 most-similar images. In this experiment, the average precision of our scheme is 0.7962. In the proposed scheme, the color probability is used as an important reference for the retrieval feature. There are some soccer images in the retrieval result of the Tai Chi image.



Figure 5. Image retrieval for Caltech 101 dataset.

Figures 6–8 show the comparison of the experimental results between the proposed scheme and other schemes. In this experiment, the number of retrieved images,  $L$ , was 20. Figure 6 shows that the proposed scheme cannot consistently obtain the best average precision in every image category. For example, in the “People” category, Yu’s scheme [18] produced the best results and Silakari’s scheme [17] provided the best results in the “Building” category. Despite this, our scheme achieved a stable average precision rate. In addition, the time complexity of our scheme is  $O(n^2)$  while the proposed scheme by literature [11] is also  $O(n^2)$ . Although the proposed scheme used the same feature descriptor as Guo’s scheme [11], it provided better retrieval results because the BACOBTC had better image quality than ODBTC. Figure 8 shows that the proposed scheme achieved better performance than the other schemes in the Corel 10000 image database. Therefore, we can conclude that the performance of our proposed scheme is superior to the other schemes in color image retrieval.

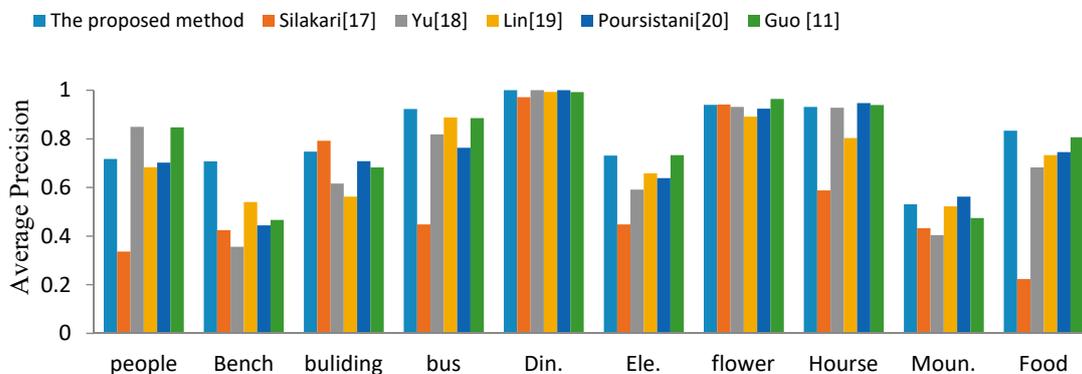


Figure 6. Contrasts of the proposed scheme and former schemes based on corel1000 image database precision rate.

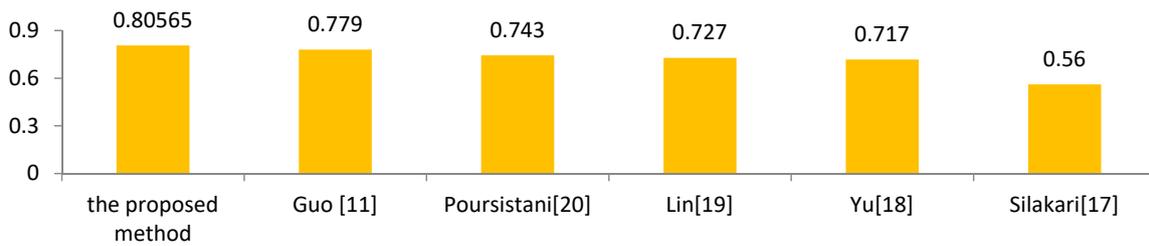


Figure 7. The average precision rate in Corel1000.

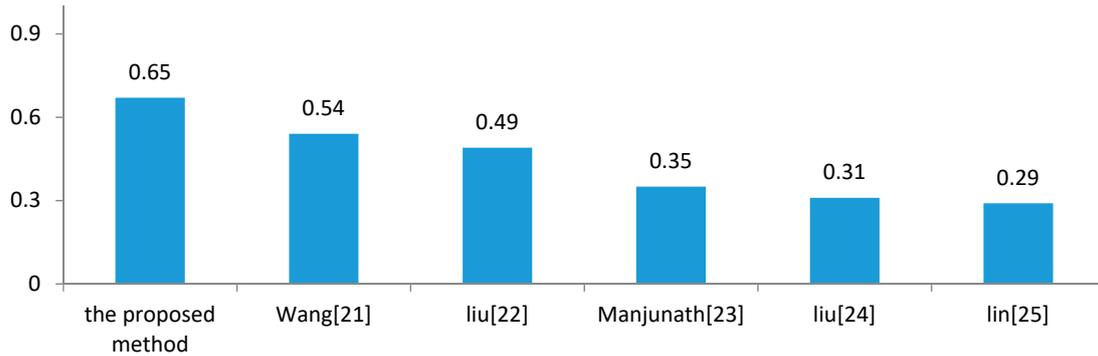


Figure 8. The average precision rate in Corel10000.

Similar to the experiment of [13], Figure 9 show four examples of image retrieval results for the Corel 1000 image database. The image at the top of the left-top corner is the query image and the rest are the 11 most-similar images. In this experiment, the proposed scheme had higher precision and better performance than the scheme in Reference [13]. In the literature [13], there are color mismatched images in the “bus” and “flower” categories and irrelevant images in the “food” category. However, as shown in Figure 9, the retrieved images are food images in the “food” category and their colours perfectly matched the query image in the “bus” and “flower” categories. The reason the precision of the proposed scheme was higher than that in the literature [13] was that  $CHF_{max}$  and  $CHF_{min}$  were used to describe the color features, obtaining the main color characteristics of the original image. In addition, we used a probabilistic approach to describe the images’ textural information in the computation of BHF, so the images’ right-and-left symmetry did not affect the results when the images were retrieved. As shown in Figure 9, the most-similar dragon images were retrieved irrespective of the direction of the dragon.



Figure 9. Cont.



Figure 9. Image retrieval for bus, flower, food and dragons.

### 5.5. Application Simulation Analysis

CBIR is widely used in many fields such as e-commerce [32], medicine [33], military, security technology [34], multimedia digital library, Geographic information System [35], architecture and engineering design [36], intellectual property [37], crime and security prevention etc. In this paper, the effectiveness of our approach is verified in e-commerce. In order to verify the feasibility and effectiveness of the retrieval scheme, this paper makes a simulation study on the online commodity retrieval system. The experiment uses the image database consisting of 1000 images in four groups (shoe, hat, watch and bag), in which each group contains 250 images. All images are from Taobao and other shopping sites. As shown in Figure 10, Our scheme also obtains good performance.

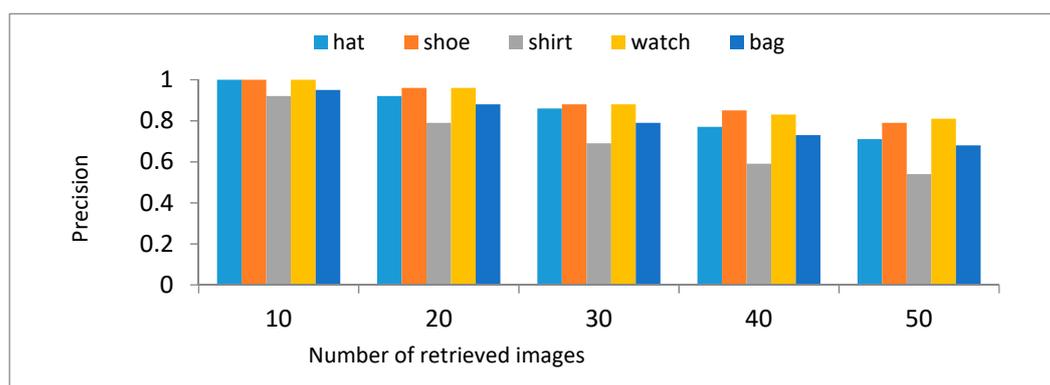


Figure 10. Precision comparison of different commodities.

Although the precision of our proposed scheme is high in the color image database, there are some shortcomings. As shown in Figure 3, the precision is lower in “people” and “mountain” categories. This is because the contents of these two groups of images are complex and the image color boundaries are not prominent. In addition, because of the lack of shape and spatial information, this scheme is not suitable for image retrieval in medicine, Geographic Information System, etc. Retrieval experiments are done after image rotation and scale changes, but the experimental results are not satisfactory.

### 5.6. Conclusions and Future Work

In content-based image retrieval, color and texture are the two most frequently extracted features. Obviously, the combination of these two features could achieve better retrieval performance. In this paper, we proposed a novel approach for the retrieval of color images using the features extracted from block truncation coding based on binary ant colony optimization. BACOBTC compresses a color image into two color quantizers and a bitmap image, and the reconstructed images have higher visual quality. Two image features, i.e., color histogram feature and bit pattern histogram feature, were used to measure the similarity between a query image and the target image in the database. Experiments are performed on two extensively-used color image databases to study the influences of different blocks and different parameter settings on Precision-Recall and to compare the results with several

other image-retrieval schemes. The results showed that the proposed scheme had a higher retrieval precision and recall ratio.

For future work, we intend to study the application of the content-based image retrieval approach. On the basis of the simulation network commodity retrieval, we will optimize the image retrieval algorithm for the application of large e-commerce image retrieval.

**Author Contributions:** Methodology, C.-C.C.; Software, Y.-H.C.; Writing—original draft, C.-Y.H.; Writing—review & editing, C.-C.L.

**Funding:** Ministry of Science and Technology, Taiwan: 105-2410-H-126-005-MY3.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Delp, E.J.; Mitchell, O.R. Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342. [[CrossRef](#)]
2. Mathews, J.; Nair, M.S. Adaptive block truncation coding technique using edge-based quantization approach. *Comput. Electr. Eng.* **2015**, *43*, 169–179. [[CrossRef](#)]
3. Yang, Y.; Chen, Q.; Wan, Y. A fast near-optimum block truncation coding scheme using a truncated K-means algorithm and interblock correlation. *AEU-Int. J. Electron. Commun.* **2011**, *65*, 576–581. [[CrossRef](#)]
4. Amarunnishad, T.M.; Govindan, V.K.; Mathew, A.T. Block truncation coding using a set of predefined bit planes. In Proceedings of the Conference on Computational Intelligence and Multimedia Applications, Sivakasi, India, 13–15 December 2007; pp. 73–78.
5. Guo, J.M.; Wu, M.F. Improved block truncation coding based on the void-and-cluster dithering approach. *IEEE Trans. Image Process.* **2009**, *18*, 211–213. [[PubMed](#)]
6. Guo, J.M. Improved block truncation coding using modified error diffusion. *Electron. Lett.* **2008**, *44*, 462–464. [[CrossRef](#)]
7. Guo, J.M.; Liu, Y.F. Improved block truncation coding using optimized dot diffusion. *IEEE Trans. Image Process.* **2014**, *23*, 1269–1275.
8. Guo, J.M.; Su, C.C. Improved Block Truncation Coding Using Extreme Mean Value Scaling and Block-Based High Speed Direct Binary Search. *IEEE Signal Proc. Lett.* **2011**, *18*, 694–697. [[CrossRef](#)]
9. Guo, J.M.; Prasetyo, H.J.; Chen, H. Content-based image retrieval using error diffusion block truncation coding features. *IEEE Trans. Circuits Syst. Video Technol.* **2015**, *25*, 466–481.
10. Guo, J.M.; Prasetyo, H. Content-based image retrieval with ordered dither block truncation coding features. In Proceedings of the IEEE International Conference on Image Processing, Melbourne, Australia, 15–18 September 2013; pp. 4006–4009.
11. Guo, J.M.; Prasetyo, H. Content-based image retrieval using features extracted from halftoning-based block truncation coding. *IEEE Trans. Image Process.* **2015**, *24*, 1010–1024.
12. Guo, J.M.; Prasetyo, H.; Wang, N.J. Effective Image Retrieval System Using Dot-Diffused Block Truncation Coding Features. *IEEE Trans. Multimed.* **2015**, *17*, 1576–1590. [[CrossRef](#)]
13. Wang, X.; Wang, Z. The scheme for image retrieval based on multi-factors correlation utilizing block truncation coding. *Pattern Recognit.* **2014**, *47*, 3293–3303. [[CrossRef](#)]
14. Wang, H.Z.; He, X.H.; Zai, W.J. Image retrieval based on combining local and global features. *Opt. Precis. Eng.* **2008**, *16*, 1098–1194.
15. Kekre, D.H.; Thepade, S.D.; Mukherjee, P.; Wadhwa, S.; Kakaiya, M.; Singh, S. Image retrieval with shape features extracted using gradient operators and slope magnitude technique with BTC. *Int. J. Comput. Appl.* **2010**, *6*, 28–33. [[CrossRef](#)]
16. Alzu'bi, A.; Amira, A.; Ramzan, N. Semantic content-based image retrieval: A comprehensive study. *J. Vis. Commun. Image Represent.* **2015**, *32*, 20–54. [[CrossRef](#)]
17. Silakari, S.; Motwani, M.; Maheshwari, M. Color image clustering using block truncation algorithm. *Int. J. Comput. Sci.* **2009**, *4*, 31–35.
18. Yu, F.X.; Luo, H.; Lu, Z.M. Colour image retrieval using pattern co-occurrence matrices based on BTC and VQ. *Electron. Lett.* **2011**, *47*, 100–101. [[CrossRef](#)]

19. Lin, C.H.; Chen, R.T.; Chan, Y.K. A smart content-based image retrieval system based on color and texture feature. *Image Vis. Comput.* **2009**, *27*, 658–665. [[CrossRef](#)]
20. Poursistani, P.; Nezamabadi-pour, H.; Moghadam, R.A.; Saeed, M. Image indexing and retrieval in JPEG compressed domain based on vector quantization. *Math. Comput. Model.* **2013**, *57*, 1005–1017. [[CrossRef](#)]
21. Wang, X.; Wang, Z. A novel scheme for image retrieval based on structure elements' descriptor. *J. Vis. Commun. Image Represent.* **2013**, *24*, 63–74. [[CrossRef](#)]
22. Wu, J.; Shen, H.; Xiao, Z.B.; Wu, Y.B.; Li, Y.D. Boosting Manifold Ranking for Image Retrieval by Mining Query Log Repeatedly. *J. Int. Technol.* **2014**, *15*, 135–143.
23. Gao, T.; Li, G.; Hou, J. Retrieving Image Resource Technique Based on Bayes Semantic Classification and Visual Feature Extraction. *J. Int. Technol.* **2012**, *14*, 929–934.
24. Liu, G.H.; Li, Z.Y.; Zhang, L.; Xu, Y. Image retrieval based on micro-structure descriptor. *Pattern Recognit.* **2011**, *44*, 2123–2133. [[CrossRef](#)]
25. Lin, C.H.; Huang, D.C.; Chan, Y.K.; Chen, K.H.; Chang, Y.J. Fast color-spatial feature based image retrieval schemes. *Expert Syst. Appl.* **2011**, *38*, 11412–11420. [[CrossRef](#)]
26. Biasotti, S.; Cerri, A.; Abdelrahman, M.; Aono, M.; Hamza, A.B.; El-Melegy, M.; Farag, A.; Garro, V.; Giachetti, A.; Giorgi, D.; et al. SHREC'14 track: Retrieval and classification on textured 3D models. In Proceedings of the Eurographics Workshop on 3D Object Retrieval, Strasbourg, France, 6 April 2014; pp. 111–120.
27. Masoumi, M.; Li, C.; Hamza, A.B. A spectral graph wavelet approach for nonrigid 3D shape retrieval. *Pattern Recognit. Lett.* **2016**, *83*, 339–348. [[CrossRef](#)]
28. Li, Z.; Jin, Q.; Chang, C.C.; Liu, L.; Wang, A. A common bitmap block truncation coding for color images based on binary ant colony optimization. *KSII Trans. Internet Inf. Syst.* **2016**, *10*, 2326–2345.
29. Kong, M.; Tian, P. A binary ant colony optimization for the unconstrained function optimization problem. In Proceedings of the International Conference on Computational Intelligence and Security, Xi'an, China, 15–19 December 2005; pp. 682–687.
30. Caltech 101 Photo Collection Image Database. 2015. Available online: [http://www.vision.caltech.edu/Image\\_Datasets/Caltech101/](http://www.vision.caltech.edu/Image_Datasets/Caltech101/) (accessed on 1 December 2018).
31. Corel Photo Collection Color Image Database. 2016. Available online: <http://wang.ist.psu.edu/docs/realtd/> (accessed on 1 December 2018).
32. Saraiva, P.C.; Cavalcanti, J.M.B.; Moura, E.S.D. A multimodal query expansion based on genetic programming for visually-oriented e-commerce applications. *Inform. Process. Manag.* **2016**, *52*, 783–800. [[CrossRef](#)]
33. Ma, L.; Liu, X.; Gao, Y. A new method of content based medical image retrieval and its applications to CT imaging sign retrieval. *J. Biomed. Inform.* **2017**, *66*, 148–158. [[CrossRef](#)]
34. Xu, Y.; Gong, J.; Xiong, L. A Privacy-Preserving Content-based Image Retrieval Method in Cloud Environment. *J. Vis. Commun. Image Represent.* **2017**, *43*, 164–172. [[CrossRef](#)]
35. Toumi, A.; Khenchaf, A.; Hoeltzener, B. A retrieval system from inverse synthetic aperture radar images: Application to radar target recognition. *Inform. Sci.* **2012**, *196*, 73–96. [[CrossRef](#)]
36. Fu, L.; Kara, L.B. From engineering diagrams to engineering models: Visual recognition and applications. *Comput. Aided Des.* **2011**, *43*, 278–292. [[CrossRef](#)]
37. Alaei, A.; Roy, P.P.; Pal, U. Logo and seal based administrative document image retrieval: A survey. *Comput. Sci. Rev.* **2016**, *22*, 47–63. [[CrossRef](#)]

