

Content-based image retrieval based on corel dataset using deep learning

Rasha Qassim Hassan, Zainab N. Sultani, Ban N. Dhannoon

Department of Computer Science, College of Science, Al-Nahrain University, Baghdad, Iraq

Article Info

Article history:

Received Nov 25, 2022

Revised Jan 18, 2023

Accepted Jan 30, 2023

Keywords:

Content-based image retrieval
Convolution neural network
Deep learning

ABSTRACT

A popular technique for retrieving images from huge and unlabeled image databases are content-based-image-retrieval (CBIR). However, the traditional information retrieval techniques do not satisfy users in terms of time consumption and accuracy. Additionally, the number of images accessible to users are growing due to web development and transmission networks. As the result, huge digital image creation occurs in many places. Therefore, quick access to these huge image databases and retrieving images like a query image from these huge image collections provides significant challenges and the need for an effective technique. Feature extraction and similarity measurement are important for the performance of a CBIR technique. This work proposes a simple but efficient deep-learning framework based on convolutional-neural networks (CNN) for the feature extraction phase in CBIR. The proposed CNN aims to reduce the semantic gap between low-level and high-level features. The similarity measurements are used to compute the distance between the query and database image features. When retrieving the first 10 pictures, an experiment on the Corel-1K dataset showed that the average precision was 0.88 with Euclidean distance, which was a big step up from the state-of-the-art approaches.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Rasha Qassim Hassan
Department of Computer Science, Al-Nahrain University
Al Jadriyah Bridge, Baghdad, Iraq
Email: rasha.qassim.cs2020@ced.nahrainuniv.edu.iq

1. INTRODUCTION

Large repositories for multimedia and image content have developed due to the growing usage of digital computers, storage technologies, and digital multimedia in recent years. This enormous volume of multimedia data is utilized in various industries, including digital forensics, electronic games, archaeology, video, satellite data and still image repositories, and medical treatment. This rapid growth has generated a continuous need for image retrieval systems that operate on a large scale [1]. For large image databases, the traditional text-based image extraction approach seems ineffective. There are certain drawbacks to retrieving images based on text, such as the time-consuming task of adding labels to individual images in huge databases. That label text depends on language and is only appropriate for one language at a time. Another drawback is that multiple users can set different labels for the same image. When retrieving images from image content, these drawbacks can be avoided. This type of image retrieval is known as content-based image retrieval (CBIR) [2].

CBIR has been a popular technique of community multimedia research since the early 1990s [3]. The main block diagram is shown in Figure 1. CBIR is the most crucial technology for image processing and computer vision. CBIR applications have been developed for various uses, including object recognition, geographic information systems, architectural design, remote sensing [4], surveillance systems, and medical image retrieval [5]. CBIR is a well-defined image search and retrieval technique. It uses the visual content of

images to find and retrieve images from huge data collections [6]. CBIR uses the search method for low-level features such as texture, shape, and color [7]. This set of low-level features generates a feature vector, which describes the content of each image in the image database. Subsequently, image retrieval is based on similarities in their contents. The similarity between the query and the feature vector dataset is used to sort the list of matching images [8]. The features used by CBIR may be divided into two groups: global feature descriptors and local feature descriptors. Global features such as color [9], shape [10], and texture [11]. Local features such as local binary pattern (LBP) [12], oriented fast and rotated binary robust independent elementary features BRIEF (ORB) [13], speeded-up robust feature (SURF) [14], scale-invariant feature transform (SIFT) [15], and histogram of oriented gradient (HOG) [16].

Local feature descriptors define each image patch, whereas global feature descriptors describe the whole image. The advantage of global feature descriptors is their quick computation, but the disadvantage is their poor precision. Global feature descriptors frequently fall short in their attempts to extract significant visual characteristics from an image. Local feature descriptors are more accurate than global feature descriptors because they use features calculated from the image's patch to represent the image. The disadvantage of local features is that they will result in large feature space for large image databases [17]. The feature vectors and similarity measures have the biggest effects on the retrieval performance of the CBIR system. There is always a semantic gap between high-level human perception and the low-level image pixels that systems collect. Researchers decided to address this problem to enhance CBIR's performance in light of the recent success of deep learning techniques, particularly the performance of convolutional neural networks (CNN), in resolving the issue of computer vision applications [18].

Srivastava and Khare [19] presented a technique for CBIR that combines local and global features. Geometric moments extract global features, while SIFT descriptors extract local features. SIFT and moments are combined to find visually similar images. The Corel 1K has an average precision of 0.3981. Mehmood *et al.* [20] combine the SURF and HOG image features. After the final features were extracted using the bag-of-visual words (BoVW) model, they used Euclidean measurement to evaluate the similarity between the query image and the database images. The average precision is 0.8061 on the Corel 1K datasets. These methods have two main drawbacks: first, they require a lot of time, and second, finding the most silent places is not always easy.

Nazir *et al.* [21], suggest a new CBIR system that combines local and global features to handle low-level information. A color histogram (CH) is used to extract color information. Edge histogram descriptor and discrete wavelet transform (DWT) extract texture features (EDH). Based on the results of the experiments, the suggested method does better on the Corel 1K, with an average precision of 0.735.

Pardede *et al.* [22] suggested a CBIR method that employed deep CNN for feature extraction from fully connected FC1 and FC2 layers. The Feature Extractor utilizes the fully connected feature vectors FV.FC1 and FV.FC2 to extract image features from each image and compares the performance of deep CNN for CBIR tasks with three classifications: softmax, support vector machine (SVM), and extreme gradient boost (XGBoost). A deep CNN model was produced based on the suggested neural network structure. The results of the mathematical experiments suggest by utilizing the XGBoost classification, the extracted feature extractor from deep CNN can improve CBIR performance, and the best feature extractor is FV.FC2. The precision on the Wang dataset (Corel 1k) is 0.69.

Öztürk [23] proposes a useful CBIR framework. The dictionary learning method addresses the training issue with a small amount of labelled data. Dictionary learning (DL) cannot produce reliable features for the retrieval task, particularly when there is a complicated background. To address both the issue of identifying objects and dealing with complicated backgrounds, a DL technique utilizing CNN's (Resnet-50) feature representation capabilities is implemented in this system. When 10 images are retrieved, the mean average precision (mAP) for the modified Corel dataset is 0.855. Öztürk [24] provided a framework for content-based medical image retrieval (CBMIR) based on high-level deep features. The insufficient number of photos is the main problem here. To address this issue, a class-driven retrieval strategy is suggested. Different hash code lengths are produced using feature reduction methods, and their performances are evaluated. Experiments with the National Electrical Manufacturers Association Magnetic Resonance Imaging (NEMA MRI) and the National Electrical Manufacturers Association Computed Tomography (NEMA CT) datasets show that the framework given is better than the existing methods in the literature. Desai *et al.* [25] suggested an effective deep learning architecture for fast image retrieval based on convolution neural networks CNN and SVM. SVM is used for classification to reduce the time required to retrieve the results. VGG16 is used to extract features. It has 12 convolutional layers, 4 fully connected layers, and, as the last layer, a SoftMax classifier. The average precision for retrieving 10 images from the Corel dataset is 0.8361. The VGG16 has 138 million parameters, which is a drawback since it causes an explosion in the gradient issue. This paper's main contributions may be summarized:

- This paper employed CNN to extract deep features from photos to bridge the semantic gap between high-level human perception and the low-level picture pixels that computers gather to produce the most relevant images.
- The goal is to determine the appropriate CNN architecture while focusing on selecting the best hyperparameter values.
- Two experiments are used: the CNN model with max pooling and the CNN model with average pooling.
- In similarity measurement phase, two distance measurements were implemented: Euclidean and City Block (Manhattan) to find the best in terms of mAP.

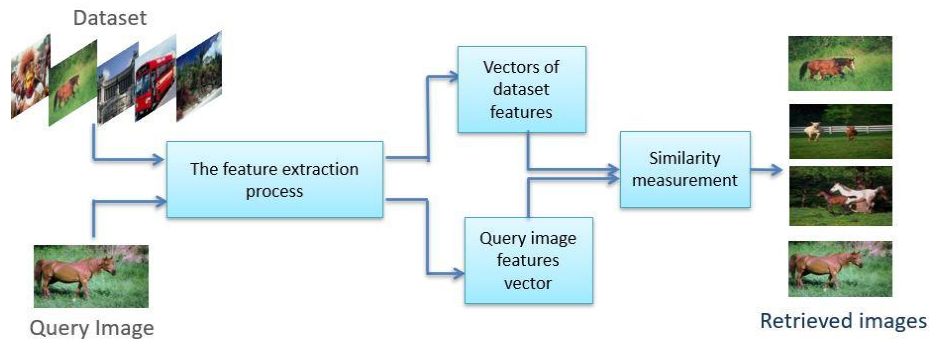


Figure 1. Block diagram of content-based image retrieval

The paper has been arranged in the following manner: Section 2 presents the proposed methodology, and Section 3 displays the similarity measurement. In addition, Section 4 presents the experimental results and discussion. This paper is concluded in Section 5.

2. METHOD

In this paper, CNN is used for feature extraction since it is efficient at closing the semantic gap between high-level human perception and low-level machine features, as well as at finding the most relevant images and improving retrieval performance. The Corel 1K [26] database was utilized to validate the results. It is a 1,000-image database collection. These images are organized into 10 categories, each of which has 100 images. A block diagram of the proposed method is illustrated in Figure 2.

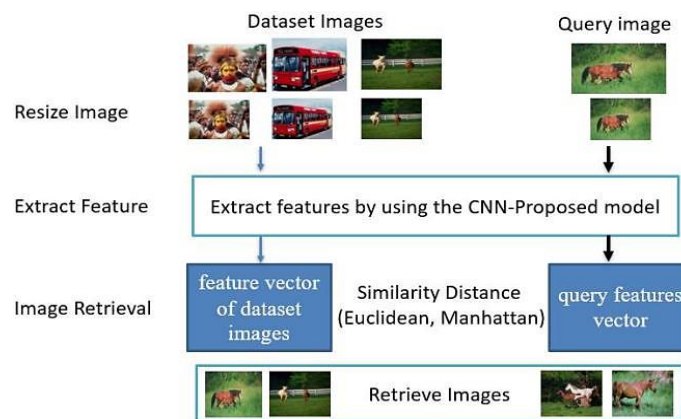


Figure 2. Block diagram of the proposed method

2.1. Feature extraction

CNN is a type of neural network model that uses several large network layers [27]. CNN has gained popularity in various image processing applications, including object recognition [28] and picture

classification [29], and has produced promising results. CNN's are increasingly used in various image-processing tasks, such as object classification, face recognition, and gesture identification. According to earlier research, it is possible to input an image directly into a CNN network and use features for image classification [30]. The basic CNN architecture is composed of convolutional layers, pooling layers, fully connected layers (FC), SoftMax layers, and non-linear activation functions like rectifier neural network (ReLU) [31], [32]. The forward pass stage includes a convolution layer, where an activation map is produced as the result of computing the dot product of the filter's input volume and the filter's dot product. Next, use the ReLU function to decrease negative values and pooling to downsample the feature maps before activating the value. Numerous iterations of this phase are carried out, with no restrictions on how often it is repeated. The final step of the forward pass enters the fully connected layer, where the output is created in vector form. To determine whether the output belongs to the model class, the SoftMax values and error values can be computed for the values in the training dataset after getting the output from the fully connected layer. CNN works on image volumes. The input volume can therefore be thought of as the input image. Width, height, and depth are the three dimensions that make up the volume. CNN's initial values for the input volume are W, H, and D [33]. The filter shifts from the top to the bottom of the input volume, beginning at the top left and moving to the top right. Every motion from left to right is performed as thoroughly as a stride. The number of steps convolutes its stride. Since ReLU transforms the negative pixel value to 0, it is a quick activation function. When the value is 0, the result is 0 [34], as seen in (1). The hidden layer's size is huge after the convolution procedure. It is typical to utilize a pooling or sub-sampling layer right after a convolutional layer to decrease computational complexity. Max and average pooling are two types of pooling that are commonly utilized [35]. Let $y=y_{ij}$ represent the matrix in a pool.

$$ReLU(Y) = \max(0, Y) \tag{1}$$

Using the maximum element in y as the output is known as max pooling, as seen in (2).

$$x = \max(y) \tag{2}$$

Taking the average of all the element values (y_{ij}) is known as average pooling [18], as seen in (3).

$$x = \frac{1}{N * M} \sum_{i=1}^M \sum_{j=1}^N y_{ij} \tag{3}$$

M and N represent the elements in the pooled matrix. This paper employed two experiments, the first with max pooling and the second with average pooling. The proposed CNN architecture model comprises 19 layers, as shown in Figure 3. A proposed CNN model is used to extract dataset feature vectors. The model contains six convolutional layers and six batch normalization layers to normalize the data. Each two-convolution layer is followed by a pooling layer, two dropout layers for regularities, and two fully connected layers. The original images, which are 384×256 or 256×384 pixels in size, will be resized to 64 x 64 pixels before they are fed into the CNN model. In the layers of the CNN model, the filter is scaled up and down so that features can be found. The internal architecture of the CNN model used to train the CBIR is shown in Table 1. The hyperparameters used in this paper are illustrated in Table 2. For the optimizer, Adam is used.

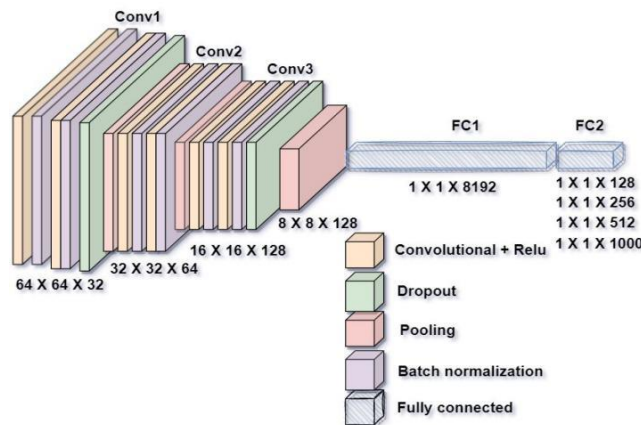


Figure 3. Proposed CNN-model architecture

Table 1. The internal architecture of the CNN model

Layers (type)	Input image shape	No. filter	Size of filter	Window size of the pooling	output	Para.
conv2d (Conv2D)	(64, 64, 3)	32	3 * 3		(64,64,32)	896
batch normalization	(64,64,32)				(64,64,32)	128
conv2d_1 (Conv2D)	(64,64,32)	32	3 * 3		(64,64,32)	9248
batch normalization_1	(64,64,32)				(64,64,32)	128
average_pooling2d	(64,64,32)			2 * 2	(32, 32, 32)	0
dropout				0.3		0
conv2d_2 (Conv2D)	(32, 32, 32)	64	3 * 3		(32, 32, 64)	18496
batch normalization_2	(32, 32, 64)				(32, 32, 64)	256
conv2d_3 (Conv2D)	(32, 32, 64)	64	3 * 3		(32, 32, 64)	36928
batch normalization_3	(32, 32, 64)				(32, 32, 64)	256
average_pooling2d_1	(32, 32, 64)			2 * 2	(16, 16, 64)	0
conv2d_4 (Conv2D)	(16, 16, 64)	128	3 * 3		(16, 16, 128)	73856
batch normalization_4	(16, 16, 128)				(16, 16, 128)	512
conv2d_5 (Conv2D)	(16, 16, 128)	128	3 * 3		(16, 16, 128)	147584
batch normalization_5	(16, 16, 128)				(16, 16, 128)	512
average_pooling2d_2	(16, 16, 128)			2 * 2	(8, 8, 128)	0
Dropout_1				0.2		0
flatten	(8, 8, 128)				8192	0

Table 2. Hyperparameters' value

Hyperparameters	Value
Split data	900 train, 100 queries
Dropout	0.3, 0.2
Batch size	128
Learning rate	0.001
Num. of epochs	500

2.2. Image retrieval

Determine the difference in similarity between the feature vector obtained from the query and the feature vector from the training dataset by using two similarity measurements: Euclidean and Manhattan. As indicated in Algorithm 1, the images with the shortest distance are returned. Evaluation matrices are used to assess the system's performance.

Algorithm 1: image retrieval

- Input: Feature vector of 128 X 1 X 1, Output: Similar image and average precision
- Label the feature vector of images for all classes.
 - Convert nominal classes to numeric values. Ex. the bus is 3, the flower is 6
 - Dataset is divided into 900 training and 100 testing
 - Compute the distance between feature vectors of all testing and training and retrieve the smallest distance.
 - Calculate the mean average precision for all testing (Queries)
 - Display the most similar images for the query image.

3. SIMILARITY MEASUREMENT

The Euclidean and Manhattan distances measure the relationship between the feature vectors of the query images and the feature vector of the training dataset images. If the distance between the two vectors is the smallest with reference to the other distances, the generated image and the query are similar, as seen in (4) and (5), respectively. Where m represents the size of the feature vector; Fv_{db} and Fv_q are the feature vectors for the dataset and query images, respectively.

$$D(q, db) = \sqrt{\sum_{i=1}^M (Fv_{db}(i) - Fv_q(i))^2} \quad (4)$$

$$D(q, db) = \sum_{i=1}^M |Fv_{db}(i) - Fv_q(i)| \quad (5)$$

4. RESULTS AND DISCUSSION

4.1. Dataset

The Corel 1K dataset [26] has 1,000 JPEG images with a 256 by 384 or 384 by 256-pixel size. 100 images in each of the 10 categories make up this collection. The categories include Africa, horses, flowers,

beaches, buses, buildings, mountains, dinosaurs, elephants, flowers, and food. Figure 4 displays an example of each category. Retrieval was made more challenging and robust by keeping all the training images in one folder and all the testing images in another. This means a test folder with 100 images, each of which is a query image, and a training folder with 900 images.

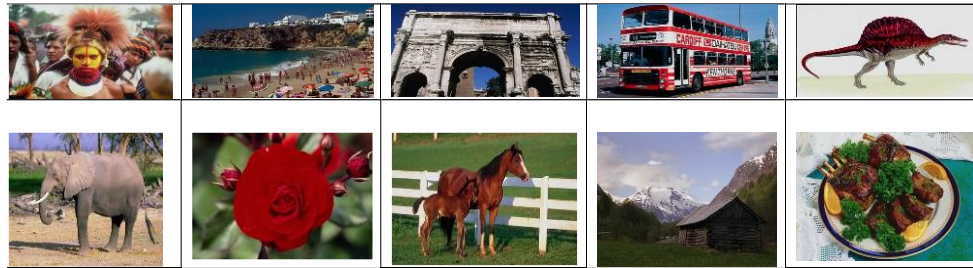


Figure 4. Examples of each category in the Corel 1K dataset

4.2. Evaluation measurement

Performance in the CBIR is assessed using precision and mean average precision (mAP). Divide the total number of images retrieved by the total number of relevant images to determine precision. It shows a system's ability to only return relevant images, as seen in (6). The mAP is the mean of the average precision for all classes. For average precision (AP), see (7), and mAP, see (8). Where P represents the precision, and n is the number of images, AP_k denotes the AP of class K, and m denotes the number of classes.

$$P = \frac{\text{Number of relevant images retrieved}}{\text{Total number of images retrieved}} \tag{6}$$

$$AP = \frac{1}{n} \sum_{i=1}^n P_i \tag{7}$$

$$mAP = \frac{1}{m} \sum_{k=1}^m AP_k \tag{8}$$

4.3. Experiment results

Retrieval performance has been assessed using precision. Higher precision means that it returns more relevant images than irrelevant ones. The precision of each query across all categories was calculated along with the average determined precision. The Corel 1K image dataset is utilized. The Corel 1K dataset contains diverse images ranging from natural scenes to outdoor activities to various animals, making it suitable for testing image retrieval systems. Two retrieval methods are used. The first method is based on CNN with max pooling, while the second is based on CNN with average pooling and various feature sizes. After several attempts to find the right hyperparameter value, shown in Tables 3 and 4, where the learning rate is 0.01, and the number of epochs is 100 and 500, respectively, the hyperparameter in Table 2 fits the architecture made in this paper.

Table 3. Average precision results when the learning rate is 0.01 and epochs are 100

Feature vector	128
Africa	0.52
Beaches	0.36
Buildings	0.47
Bus	0.84
Dinosaurs	0.97
Elephants	0.59
Flowers	0.93
Horses	0.62
Mountains	0.78
Foods	0.42
mAP	0.649

Table 4. Average precision results when the learning rate is 0.01 and epochs are 500

Feature vector	Flatten (8192)	128	256
Africa	0.5	0.66	0.64
Beaches	0.59	0.42	0.41
Buildings	0.25	0.66	0.68
Bus	0.46	0.95	0.96
Dinosaurs	0.98	0.98	0.98
Elephants	0.66	0.83	0.8
Flowers	1	0.98	0.97
Horses	0.83	0.56	0.59
Mountains	0.69	0.71	0.68
Foods	0.48	0.61	0.57
mAP	0.644	0.736	0.729

The results depend on the nature of the images. Some classes have simple and distinct colors that make distinguishing objects from the background easier. Other classes have similar colors, so it is difficult to distinguish. Also, max pooling selects the strong pixels and almost neglects the weak ones. It works as an edge

detector. As for average pooling, it takes a group of pixels, combines them, and divides them by a number according to the length of the matrix, working as an image smoother. Bus, Buildings, dinosaurs, flowers, horses, and mountains have the highest average precision of any other class in the average pooling. For max pooling, the classes bus, dinosaur, flower, horse, and mountain have the highest average precision. Table 5 shows the average precision for Euclidean distance. A feature size of 128 based on CNN with average pooling achieves higher average precision than max pooling when using Manhattan distance, as seen in Table 6. The best result, as shown in Tables 5 and 6, was achieved at Euclidean distance with feature sizes of 256 and 10 retrieve images. Figure 5 compares the average precision measured on the Corel 1K dataset to state-of-the-art traditional methods. The retrieved images, according to a query image using CNN, are illustrated in Figure 6.

Table 5. The average precision for each class with different feature vector sizes for average pooling and max pooling based on Euclidean distance

Feature size		Flatten (8,192)		128		256		512		1,000	
AVG-pooling	Number of retrieved images	10	20	10	20	10	20	10	20	10	20
	Africa	0.71	0.66	0.9	0.88	0.91	0.89	0.76	0.71	0.73	0.71
	Beaches	0.63	0.59	0.65	0.65	0.65	0.65	0.78	0.75	0.77	0.74
	Buildings	0.86	0.83	0.95	0.94	0.95	0.95	0.9	0.9	0.9	0.9
	Bus	0.96	0.92	0.9	0.9	0.9	0.89	0.98	0.96	0.94	0.93
	Dinosaurs	1	1	1	1	1	1	1	1	1	1
	Elephants	0.38	0.35	0.75	0.75	0.75	0.73	0.75	0.74	0.75	0.74
	Flowers	1	1	1	1	1	1	1	1	1	1
	Horses	1	1	0.98	0.98	0.99	0.98	0.97	0.97	0.98	0.97
	Mountains	0.84	0.79	1	1	1	1	1	0.99	1	0.99
	Foods	0.49	0.45	0.65	0.66	0.65	0.66	0.53	0.52	0.54	0.53
	mAP	0.787	0.759	0.878	0.876	0.88	0.876	0.866	0.855	0.86	0.852
Max-pooling	Africa	0.71	0.61	0.79	0.79	0.8	0.8	0.76	0.72	0.75	0.72
	Beaches	0.55	0.53	0.5	0.5	0.51	0.52	0.59	0.56	0.58	0.56
	Buildings	0.71	0.66	0.75	0.73	0.8	0.73	0.81	0.82	0.8	0.81
	Bus	0.65	0.62	0.94	0.92	0.93	0.91	0.88	0.86	0.84	0.84
	Dinosaurs	1	1	1	1	1	1	1	1	1	1
	Elephants	0.7	0.65	0.88	0.86	0.88	0.86	0.78	0.78	0.81	0.81
	Flowers	1	1	1	1	1	1	1	1	1	1
	Horses	1	0.97	0.92	0.92	0.92	0.91	0.9	0.9	0.9	0.9
	Mountains	0.9	0.8	0.9	0.86	0.88	0.85	0.81	0.83	0.81	0.82
	Foods	0.43	0.42	0.68	0.63	0.64	0.62	0.32	0.33	0.31	0.32
	mAP	0.765	0.726	0.835	0.821	0.835	0.821	0.783	0.78	0.781	0.778

Table 6. The average precision for each class with different feature vector sizes for average pooling and max pooling based on Manhattan distance

Feature Size		Flatten (8,192)		128		256		512		1,000	
AVG-pooling	Number of retrieved images	10	20	10	20	10	20	10	20	10	20
	Africa	0.58	0.56	0.93	0.9	0.9	0.89	0.76	0.74	0.79	0.78
	Beaches	0.64	0.62	0.63	0.62	0.62	0.65	0.78	0.76	0.78	0.76
	Buildings	0.84	0.81	0.93	0.93	0.96	0.95	0.9	0.9	0.9	0.9
	Bus	0.92	0.88	0.89	0.88	0.9	0.89	0.93	0.93	0.91	0.91
	Dinosaurs	1	1	1	1	1	1	1	1	1	1
	Elephants	0.31	0.28	0.75	0.74	0.75	0.73	0.75	0.76	0.72	0.73
	Flowers	1	1	1	1	1	1	1	1	1	1
	Horses	1	0.99	1	1	1	0.98	0.98	0.98	0.98	0.97
	Mountains	0.8	0.74	1	0.99	1	1	1	0.98	1	0.99
	Foods	0.48	0.46	0.65	0.66	0.66	0.66	0.53	0.55	0.55	0.56
	mAP	0.756	0.733	0.879	0.872	0.879	0.876	0.864	0.86	0.863	0.86
Max-pooling	Africa	0.62	0.54	0.82	0.8	0.79	0.8	0.77	0.74	0.76	0.74
	Beaches	0.59	0.55	0.57	0.56	0.51	0.52	0.65	0.63	0.62	0.58
	Buildings	0.77	0.71	0.73	0.72	0.76	0.73	0.8	0.8	0.81	0.82
	Bus	0.63	0.59	0.91	0.9	0.93	0.91	0.85	0.86	0.84	0.84
	Dinosaurs	1	1	1	1	1	1	1	1	1	1
	Elephants	0.67	0.62	0.86	0.85	0.88	0.86	0.76	0.76	0.77	0.79
	Flowers	1	1	1	1	1	1	1	1	1	1
	Horses	0.97	0.96	0.93	0.93	0.93	0.91	0.91	0.9	0.9	0.9
	Mountains	0.89	0.79	0.9	0.87	0.87	0.85	0.84	0.84	0.82	0.83
	Foods	0.5	0.44	0.63	0.61	0.62	0.62	0.34	0.34	0.33	0.3
	mAP	0.764	0.721	0.836	0.823	0.827	0.821	0.792	0.787	0.786	0.78

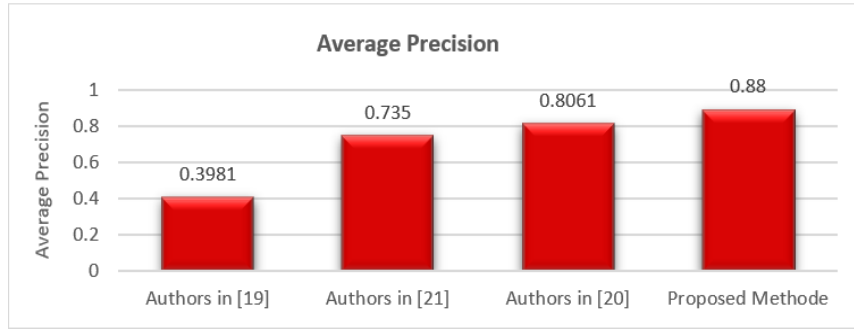


Figure 5. Comparison with state-of-the-art traditional methods

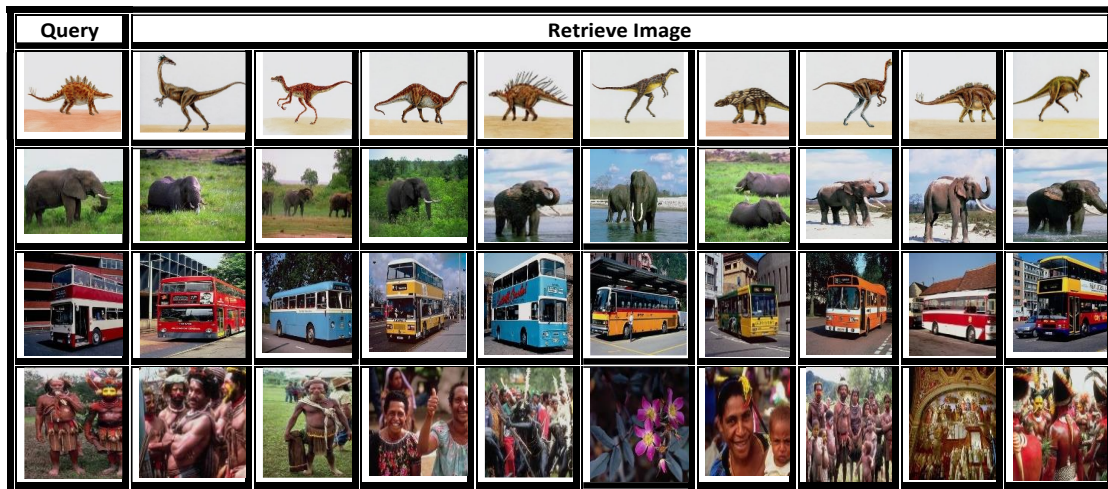


Figure 6. Query images are on the left, while their retrieved images are on the right

4.4. Comparing the results with other papers employing deep learning

This paper compares the results with two other papers using deep learning and the same dataset shown in Table 7. Pardede *et al.* [22] used deep CNN for feature extraction from FC1 and FC2 and SVM, softmax, and XGBoost for classifiers. Deep CNN architectures used 3 Conv layers+ReLU and 3 max-pooling layers, two FC layers to minimize overfitting before flattening, 3 dropouts, and 1 batch normalization. Desai *et al.* [25] extract features using a VGG16 layered CNN model. VGG16 consists of 12 Conv layers, 5 max-pooling layers, and 4 FC layers. The feature vector created from these layers is then fed to the SVM, which calculates the distance between each image in the dataset and the query image. The proposed CNN model used six Conv layers, six batch normalization layers, three average pooling layers, and two FC layers with hyperparameters, as shown in Table 2. As a result, compared to methods in related work for the same dataset, this paper's structure and parameters produced better results.

Table 7. Comparison with a state-of-the-art method

	Authors in [22] 2019	Authors in [25] 2021	Proposed method using Avg. pool
No. of the retrieved image		10	10
Africa	0.63	0.84	0.91
Beaches	0.53	0.8406	0.65
Buildings	0.36	0.8353	0.95
Bus	0.54	0.8273	0.9
Dinosaurs	0.82	0.832	1
Elephants	0.63	0.8386	0.75
Flowers	1	0.8308	1
Horses	0.93	0.8413	0.99
Mountains	0.60	0.838	1
Foods	0.88	0.8373	0.65
Avg.	0.69	0.8361	0.88

5. CONCLUSION

This paper proposed the CBIR technique using CNN for feature extraction. Two different pooling layers are used to extract features: max and average. The performance of the proposed method was assessed using precision and mAP. Euclidean and Manhattan similarity measurements are used to compute the distance between the query and database image features. The experiment results on the Corel 1K dataset with Euclidean showed a significant improvement in average precision of 0.88 when using average pooling with a feature size of 256 for retrieving the first 10 images when compared to other methods that had previously been proposed, such as CNN+SVM, SIFT, local and global CH for a color feature, DWT+EDH for a texture feature, and BoVW that used two feature extractions like HOG and SURF. The proposed method is more accurate than the existing state-of-the-art approaches, which are good and promising.

ACKNOWLEDGEMENTS

The author(s) received no financial support for the research, authorship, or publication of this article.




REFERENCES

- [1] S. Hamreras, R. Benítez-Rochel, B. Boucheham, M. A. Molina-Cabello, and E. López-Rubio, "Content based image retrieval by convolutional neural networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11487 LNCS, pp. 277–286, 2019, doi: 10.1007/978-3-030-19651-6_27.
- [2] K. Ramanjaneyulu, K. V. Swamy, and C. H. S. Rao, "Novel CBIR system using CNN architecture," in *Proceedings of the 3rd International Conference on Inventive Computation Technologies, ICICT 2018*, 2018, pp. 379–383, doi: 10.1109/ICICT43934.2018.9034389.
- [3] M. K. Alsmadi, "An efficient similarity measure for content based image retrieval using memetic algorithm," *Egyptian Journal of Basic and Applied Sciences*, vol. 4, no. 2, pp. 112–122, 2017, doi: 10.1016/j.ejbas.2017.02.004.
- [4] F. Ye, M. Dong, W. Luo, X. Chen, and W. Min, "A new re-ranking method based on convolutional neural network and two image-to-class distances for remote sensing image retrieval," *IEEE Access*, vol. 7, pp. 141498–141507, 2019, doi: 10.1109/ACCESS.2019.2944253.
- [5] U. A. Khan, A. Javed, and R. Ashraf, "An effective hybrid framework for content based image retrieval (CBIR)," *Multimedia Tools and Applications*, vol. 80, no. 17, pp. 26911–26937, 2021, doi: 10.1007/s11042-021-10530-x.
- [6] M. Garg and G. Dhiman, "A novel content-based image retrieval approach for classification using GLCM features and texture fused LBP variants," *Neural Computing and Applications*, vol. 33, no. 4, pp. 1311–1328, 2021, doi: 10.1007/s00521-020-05017-z.
- [7] I. M. Hameed, S. H. Abdhussain, and B. M. Mahmood, "Content-based image retrieval: a review of recent trends," *Cogent Engineering*, vol. 8, no. 1, 2021, doi: 10.1080/23311916.2021.1927469.
- [8] J. M. Patel and N. C. Gamit, "A review on feature extraction techniques in content based image retrieval," *Proceedings of the 2016 IEEE International Conference on Wireless Communications, Signal Processing and Networking, WiSPNET 2016*, pp. 2259–2263, 2016, doi: 10.1109/WiSPNET.2016.7566544.
- [9] U. A. Khan and A. Javed, "A hybrid CBIR system using novel local tetra angle patterns and color moment features," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 7856–7873, 2022, doi: 10.1016/j.jksuci.2022.07.005.
- [10] R. Ashraf *et al.*, "Content based image retrieval by using color descriptor and discrete wavelet transform," *Journal of Medical Systems*, vol. 42, no. 3, 2018, doi: 10.1007/s10916-017-0880-7.
- [11] S. P. Rana, M. Dey, and P. Siarry, "Boosting content based image retrieval performance through integration of parametric and nonparametric approaches," *Journal of Visual Communication and Image Representation*, vol. 58, pp. 205–219, 2019, doi: 10.1016/j.jvcir.2018.11.015.
- [12] P. Liu, J. M. Guo, K. Chamnongthai, and H. Prasetyo, "Fusion of color histogram and LBP-based features for texture image retrieval and classification," *Information Sciences*, vol. 390, pp. 95–111, 2017, doi: 10.1016/j.ins.2017.01.025.
- [13] P. Chhabra, N. K. Garg, and M. Kumar, "Content-based image retrieval system using ORB and SIFT features," *Neural Computing and Applications*, vol. 32, no. 7, pp. 2725–2733, 2020, doi: 10.1007/s00521-018-3677-9.
- [14] F. Baig *et al.*, "Boosting the performance of the BoVW Model Using SURF-CoHOG-based sparse features with relevance feedback for CBIR," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 1, pp. 99–118, Mar. 2020, doi: 10.1007/s40998-019-00237-z.
- [15] Z. N. Sultani and B. N. Dhannoon, "Modified bag of visual words model for image classification," *Al-Nahrain Journal of Science*, vol. 24, no. 2, pp. 78–86, Jun. 2021, doi: 10.22401/ANJS.24.2.11.
- [16] V. N. Degaonkar, P. Gadakh, P. Saha, and A. V. Kulkarni, "Retrieve content images using color histogram, LBP and HOG," in *Proceedings of the 4th International Conference on Electronics, Communication and Aerospace Technology, ICECA 2020*, 2020, pp. 896–899, doi: 10.1109/ICECA49313.2020.9297608.
- [17] Suharjito, Andy, and D. D. Santika, "Content based image retrieval using bag of visual words and multiclass support vector machine," *ICIC Express Letters*, vol. 11, no. 10, pp. 1479–1488, 2017.
- [18] O. Mohamed, K. El Asnaoui, O. Mohammed, and A. Brahim, "Content-based image retrieval using convolutional neural networks," *Advances in Intelligent Systems and Computing*, vol. 756, pp. 463–476, 2019, doi: 10.1007/978-3-319-91337-7_41.
- [19] P. Srivastava and A. Khare, "Content-based image retrieval using scale invariant feature transform and moments," in *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering, UPCON 2016*, 2017, pp. 162–166, doi: 10.1109/UPCON.2016.7894645.
- [20] Z. Mehmood, F. Abbas, T. Mahmood, M. A. Javid, A. Rehman, and T. Nawaz, "Content-based image retrieval based on visual words fusion versus features fusion of local and global features," *Arabian Journal for Science and Engineering*, vol. 43, no. 12, pp. 7265–7284, 2018, doi: 10.1007/s13369-018-3062-0.
- [21] 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: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings*, 2018, vol. 2018-Janua,




- pp. 1–6, doi: 10.1109/ICOMET.2018.8346343.
- [22] J. Pardede, B. Sitohang, S. Akbar, and M. L. Khodra, “Improving the performance of CBIR using XGBoost classifier with deep CNN-based feature extraction,” 2019, doi: 10.1109/ICoDSE48700.2019.9092754.
- [23] S. Öztürk, “Convolutional neural network based dictionary learning to create hash codes for content-based image retrieval,” *Procedia Computer Science*, vol. 183, pp. 624–629, 2021, doi: 10.1016/j.procs.2021.02.106.
- [24] Ş. Öztürk, “Class-driven content-based medical image retrieval using hash codes of deep features,” *Biomedical Signal Processing and Control*, vol. 68, 2021, doi: 10.1016/j.bspc.2021.102601.
- [25] P. Desai, J. Pujari, C. Sujatha, A. Kamble, and A. Kambli, “Hybrid approach for content-based image retrieval using VGG16 layered architecture and SVM: An application of deep learning,” *SN Computer Science*, vol. 2, no. 3, 2021, doi: 10.1007/s42979-021-00529-4.
- [26] J. Li and J. Z. Wang, “Automatic linguistic indexing of pictures by a statistical modeling approach,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1075–1088, 2003, doi: 10.1109/TPAMI.2003.1227984.
- [27] D. A. Jasm, M. M. Hamad, and A. T. H. Alrawi, “Deep image mining for convolution neural network,” *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 347–352, 2020, doi: 10.11591/ijeecs.v20.i1.pp347-352.
- [28] Z. Widyantoko, T. Purwati Widowati, I. Isnaini, and P. Trapsiladi, “Expert role in image classification using CNN for hard to identify object: distinguishing batik and its imitation,” *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 10, no. 1, p. 93, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp93-100.
- [29] M. Syarif and W. Setiawan, “Convolutional neural network for maize leaf disease image classification,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 3, pp. 1376–1381, 2020, doi: 10.12928/TELKOMNIKA.v18i3.14840.
- [30] V. Bhandi and K. A. Sumithra Devi, “Image retrieval by fusion of features from pre-trained deep convolution neural networks,” in *1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing and Communication Engineering, ICATIECE 2019*, 2019, pp. 35–40, doi: 10.1109/ICATIECE45860.2019.9063814.
- [31] R. Poojary, R. Raina, and A. K. Mondal, “Effect of data-augmentation on fine-tuned cnn model performance,” *IAES International Journal of Artificial Intelligence*, vol. 10, no. 1, pp. 84–92, 2021, doi: 10.11591/ijai.v10.i1.pp84-92.
- [32] N. A. Mohammed, M. H. Abed, and A. T. Albu-Salih, “Convolutional neural network for color images classification,” *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 3, pp. 1343–1349, 2022, doi: 10.11591/eei.v11i3.3730.
- [33] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” vol. 21, no. 1, pp. 1–9, Nov. 2015, doi: 10.48550/arXiv.1511.08458.
- [34] Z. Rian, V. Christanti, and J. Hendryli, “Content-based image retrieval using convolutional neural networks,” in *Proceedings - 2019 IEEE International Conference on Signals and Systems, ICSigSys 2019*, 2019, pp. 1–7, doi: 10.1109/ICSIGSYS.2019.8811089.
- [35] Fachrudin, Saparudin, E. Rasywir, and Y. Pratama, “Network and layer experiment using convolutional neural network for content based image retrieval work,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 20, no. 1, pp. 118–128, 2022, doi: 10.12928/TELKOMNIKA.v20i1.19759.

BIOGRAPHIES OF AUTHORS






Rasha Qassim Hassan    is a Master’s Student at Al-Nahrain University/College of Science/Computer science department. She received her BSc from Al-Nahrain University in 2014. Her special area is in Image Processing she can be contacted at email: rasha.qassim.cs2020@ced.nahrainuniv.edu.iq, rashaqassim92@gmail.com.



Dr. Zainab N. Sultani    is a Assistant Professor Dr. at Al-Nahrain University, College of Science/ Computer Science department. She received her BSc in Computer Engineering from Al-Balqa’a University in 2006, the MSc from Middle East University in 2012 and PhD from Computer Science Department-University of Technology in 2016. Her special area of research is in Machine Learning, Data Mining, and Image Processing. She can be contacted at email: zainab.namhabdula@nahrainuniv.edu.iq.



Dr. Ban N. Dhannoon    PhD holder since 2001, a professor in Computer Science Dept./College of Science/Al-Nahrain University, Iraq. My main research concerns are: Artificial Intelligent (natural language processing, machine learning and Deep Learning) and Digital Image Processing. She can be contacted at email: ban.n.dhannoon@nahrainuniv.edu.iq.