

# Content based image retrieval using visual geometric group19 with Jaccard similarity measure

Rajath Arakere Narayanaswamy<sup>1,2</sup>, Vidyalakshmi Krishne Gowda<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, GSSS Institute of Engineering and Technology for Women, Mysuru, India

<sup>2</sup>Department of Computer Science dan Engineering (Data Science), Sri Siddhartha Institute of Technology (A Constituent College of Sri Siddhartha Academy of Higher Education), Tumakuru, India

## Article Info

### Article history:

Received Oct 23, 2024

Revised Apr 6, 2025

Accepted Apr 23, 2025

### Keywords:

Content-based image retrieval

Higher-dimensional data

Jaccard

Normalization

Visual geometric group19

## ABSTRACT

Content-based image retrieval (CBIR) is an important research area that focuses on emerging techniques for handling large image collections and enabling efficient retrieval. The main challenge of image retrieval is to extract relevant feature vectors for image description. Therefore, visual geometric group 19 (VGG19) with Jaccard is proposed in this research for CBIR. The VGG19 allows to capture of hierarchical features, and it is appropriate for texture and fine detail characteristics. It enables to production of robust and discriminative feature representations because of numerous convolutional layers. The Jaccard is utilized as a similarity measure among feature vectors by comparing the union and intersection of feature sets. It is helpful to manage sparse and higher-dimensional data that provides a fast and accurate image retrieval process. The Caltech 256 and Corel 1K datasets are considered and it is preprocessed by image resizing and normalization. The raw images are resized to ensure dataset similarity and normalized into the range of 0 and 1. The metrics such as recall, f-measure, and average precision are used to calculate the VGG19-Jaccard performance. The VGG19-Jaccard achieves average precision of 99.0 and 99.8% for Caltech 256 and Corel 1K datasets compared to the two-stage CBIR technique.

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



## Corresponding Author:

Rajath Arakere Narayanaswamy

Department of Computer Science and Engineering

GSSS Institute of Engineering and Technology for Women

KRS Road, Metagalli, Mysuru-570 016, India

Email: rajath191985@gmail.com

## 1. INTRODUCTION

Content-based image retrieval (CBIR) is the process of retrieving images relevant to a query image from an extensive number of visual contents [1]. It is used to search for feature representation of an image to retrieve ranked image collection based on its similarity to the query instance [2]. Based on the visual content of a query image, CBIR searches for related images, with applications in areas such as online shopping, remote diagnostics, and facial recognition [3]. Due to the growing demand for image-based data retrieval systems, CBIR has become a leading technology in this field [4]. It takes most significant and broad applications in numerous areas such as architecture design, education, medical science, the Department of Justice, and Military affairs [5]. Image similarity plays a significant role in the CBIR system which ensures accurate matching and enhances the overall search experience [6]. The user submits the query, and the system retrieves images through the similarity criterion in descending way [7]. The dual fundamental steps of each CBIR are feature extraction and similarity measurements [8]. The CBIR has numerous upgrades which

are designed to enhance its efficiency and retrieval performance [9]. The CBIR schemes concentrate on prominent image features which enhances the retrieval accuracy and relevancy according to query search [10]. The major problem related to CBIR is to get significant data from raw data for removing the semantic gap [11]. It demonstrates the difference between high-level representations of images and low-level concepts. The visual feature represents the extracted images which are matched through others in data to identify similar images [12]. The image features represent the color or shape distribution of the image. The shape-based image retrieval utilizes moment or edges while color-based retrieval utilizes a histogram of image pixel values [13]. Recently various global, local, and deep learning (DL) features have been developed for CBIR by considering various mid-level presentations and visual properties like color, shape, and texture. The main problem in CBIR is to calculate the relevance of query images based on dataset images [14], [15]. Most of the existing research mostly focused on matching similar images through leveraging multi-visual features [16]. In this context, several approaches related to some image retrieval methods that have been used in recent years are discussed along with their limitations. Taheri *et al.* [17] suggested a combination of low-level features and a deep Boltzmann machine for CBIR (LB-CBIR). The feature vector integrated low-level and mid-level features. The low-level feature extraction contains shape, color, and texture which are accomplished through auto-correlogram, Gabor wavelet transform, and multi-level fractal dimension. The mid-level features are extracted by a deep Boltzmann machine. The LB-CBIR has better performance because of its low-level feature extraction. However, it does not concentrate and extracts the high-level features which affect the model's flexibility. Fadaei *et al.* [18] presented a multi-scale averaging local binary patterns for CBIR. Initially, the image was created in various scales, and texture features were extracted from the image scale. Lastly, extracted features of various ranges are integrated to create the last feature vector. Every local pattern suffers from major drawbacks in that it does not handle basic image data. It addressed overfitting and generalization issues, but the similarity was not measured thereby reducing the CBIR performance.

Wang *et al.* [19] introduced a two-stage CBIR by sparse representation and feature fusion. The generalized search tree (GIST) features are primarily applied to retrieve images with the same scene data by calculating Canberra distance. Then, sparse coding and pooling features are applied to attain a sparse presentation of local features from retrieval results. At last, the Euclidean distance is applied to calculate sparse feature vector similarity to obtain better retrieval results. Nevertheless, it fails to detect the irregular shape images and reduces the interpretability which affects the performance of CBIR. Geetha *et al.* [20] developed an image retrieval for extracting local features that depended on a combination of scale-invariant feature transform (SIFT) and KAZE. The strength of the local feature descriptor SIFT was complemented through the global feature descriptor KAZE. The SIFT focused on the whole image area by featuring fine points and ponders of KAZE on boundary details. The combination of local and global feature descriptors enhanced an image retrieval that has a diverse classification of semantics and supports to attain good results in huge-scale retrieval. However, the method required a high amount of objects to search. Rahbar and Taheri [21] implemented a triplet loss function-based binary cross-entropy for CBIR. It enables deep metric learning and creates discriminative feature space with the highest discriminative among classes and distance. The image features are extracted through pre-trained CNN. Then, the Siamese network was trained to build discriminative feature space by deep metric network. The model does not extract many informative and hierarchical patterns due to vanishing gradient issues. However, its generalization performance was reduced, and bridging the semantic gap problem occurred among low-level and high-level semantic features.

From the above analysis, the existing techniques have limitations such as not concentrating and extracting the high-level features which affect the model's flexibility. The similarity was not measured thereby reducing the CBIR performance. Struggles to detect irregular shape images reduce interpretability and require a large number of objects to search. The generalization performance was reduced and bridging the semantic gap problem occurred among low-level and high-level semantic features. To tackle these limitations, the visual geometric group 19 (VGG19) with Jaccard is proposed in this research. The key contributions are stated as follows: the VGG19 is selected for feature extraction which effectively captures a hierarchical feature with more information in the image because of numerous convolutional layers. It is useful in producing fine and layered vision characteristics which is helpful in matching images at CBIR. Jaccard is used to compare the similarity and image divergence of different samples due to its effectiveness. It enables better-discriminating images which have semantic content at a higher level. This research paper is systematized as: section 2 describes the details of the proposed methodology. Section 3 explains the results and discussion. The conclusion of this research paper is given in section 4.

## 2. RESEARCH METHOD

The VGG19 with Jaccard is proposed for CBIR system. Dataset such as Caltech 256 and Corel 1K are preprocessed by using image resizing and normalization. It is extracted by using VGG19 which generates

a feature vector that presents the key visual characteristics of images. Then, the similarity is calculated by Jaccard, and the images are retrieved. Figure 1 denotes the process of the proposed methodology.

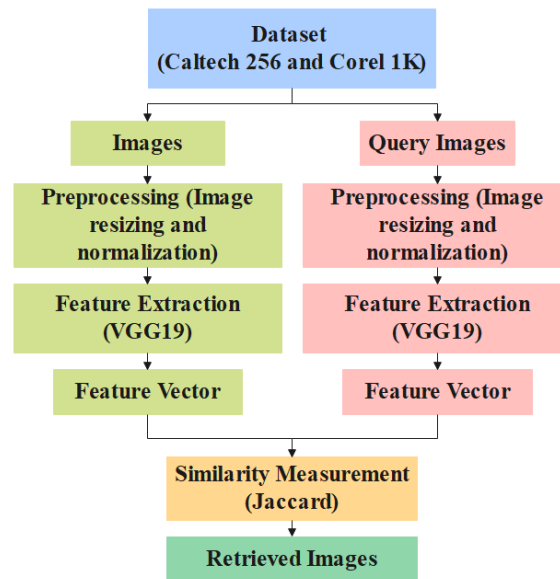


Figure 1. Process of the proposed methodology

## 2.1. Dataset

This research utilizes the Caltech 256 and Corel 1K datasets which are detailed with the total number of classes and images. The Caltech 256 dataset [22] has 30,607 images across 256 classes, every class has a minimum of 80 images. The dataset includes various classes such as rings, brains, and diamonds. The Corel 1K dataset [23] has 10,000 images across 1,000 classes, every class has 100 images. The dataset includes various classes such as cake, flower, and bus. The dataset is divided into training and testing ratio of 70:30. Table 1 represents a dataset description.

Table 1. Dataset description

Dataset	Number of classes	Number of images	Image size
Caltech 256	256	30,607	300×200 pixels
Corel 1K	1,000	10,000	128×128 pixels

## 2.2. Preprocessing

The Caltech 256 and Corel 1K datasets are preprocessed by using image resizing and normalization. Each dataset has a different size of images, so it is resized and normalized into the scale of 0 and 1. The explanation of image resizing and normalization is given as follows,

### 2.2.1. Image resizing

The DL algorithm quickly trains with small image sizes. As the raw input images vary in size various DL algorithms require the image to resize in the same dimension. It is required to standardize the image size before training. Therefore, raw images are resized into 256×256 dimensions to ensure dataset similarity. The pixel values are adjusted to small size and inappropriate regions are removed during the resizing process.

### 2.2.2. Normalization

The normalization is used to uniform the values from different query results which generate values in 0 to 1 ranges [24]. The normalized scores on every feature are multiplied through weights to achieve a total score. If  $x$  is an image value,  $x_{max}$  is a maximum score and the normalized score  $x'$  is estimated by (1).

$$x' = 1 - \frac{x}{x_{max}} \quad (1)$$

### 2.3. Feature extraction

The VGG19 has three fully connected (FC) and sixteen 2D convolution (conv) layers [25]. In the training phase, the conv layer is used for feature extraction, and the max-pooling layer with few conv layers is used to reduce the feature dimensionality. The VGG19 is selected for feature extraction because of its effectiveness in capturing features with much information. It is efficient in producing fine and layered vision characteristics which are useful in matching images at CBIR. Initially, the 2D conv layer is applied distinctly to every input image by the rectified linear unit (ReLU) activation function for spatial feature extraction. It has 64 filters through the ReLU function. To create conv results with less complexity, a max-pooling layer is applied with a  $2 \times 2$  matrix which performs a down-sampling procedure. Then, three dual conv layers have 128 filters with the matrix of  $3 \times 3$  which utilized the ReLU function. These included layers allow VGG19 to differentiate higher-level features that have been lost in past conv layers. Then, max-pooling by  $2 \times 2$  pooling is followed by four 2D conv layers which have the formation of 256 filters with a matrix of  $3 \times 3$  that is followed by max-pooling and the 2D conv layer has a formation of 512 filters with max-pooling. Then, four more 2D conv layers have a formation of 1,024 filters which is followed by max-pooling. Two fully connected (FC) layers are organized through 4,096 neurons and ReLU function which is followed by FC layer with neurons of 1,000. Lastly, the output is minimized to dual classes through the softmax activation function. The variance among true and predicted scores for VGG19 is attained by using binary cross-entropy loss function which is estimated by (2), where,  $y$  and  $\hat{y}$  are true and predicted labels.

$$loss = -[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})] \quad (2)$$

#### 2.3.1. Convolution layer

The convolution layer is a main element in VGG19 which utilizes a group of input images and filters to develop 2D layers. The VGG19 enables the model to record image function due to the weight distribution with less cost. In the CBIR, it has sixteen conv layers and a  $3 \times 3$  filter. Every conv layer is applied as a set of kernels to the input which captures spatial hierarchies such as textures, edges, and intricate structures. The VGG19 stacks numerous conv layers before max-pooling which enables deep feature extraction among various levels.

#### 2.3.2. Max polling layer

The max pooling is used for image feature extraction which is the most usual form of pooling layer. It used a  $2 \times 2$  filter for selecting the conv layer of the activation map. The pooling layer minimizes the size and execution parameters in the network that handle overfitting and enhances the network process. The VGG19 reduces the computational load and overcomes the overfitting issues by reducing the number of parameters. The max pooling is applied after each two or three conv layers which contribute to capturing hierarchical patterns in data.

#### 2.3.3. Rectified linear unit

The ReLU is a type of activation function that calculates the input values and it requires less computer resources compared to other activation functions. The neural networks are trained through VGG19 which has backpropagation stages and modified the weights to reduce losses for every epoch. If the number of layers is enhanced, the gradients are decreased linearly which reduces the gradient score. If the gradient is zero because of activation it pushes the score to 0 which is known as a vanishing gradient issue. The ReLU is given in (3).

$$ReLU(x) = \max(0, x) = \begin{cases} 0, & \text{if } x \leq 0 \\ x, & \text{if } x > 0 \end{cases} \quad (3)$$

The ReLU is used after every conv and FC layer which presents non-linearity through converting all negative scores in the feature map to 0. It assists in training deep networks by reducing vanishing gradient issues. This assists in boosting the training process due to efficiency which requires simple threshold operation. In VGG19, ReLU enhances the capability to learn intricate patterns which enables it to provide better performance in CBIR.

#### 2.3.4. Fully connected layer

In VGG19, the FC layer is used to transform higher-level features extracted by conv layers into final classifier outputs. After conv, pooling layers, the output is flattened and fed into the FC layer. The VGG19 has three FC layers such as the first two have 4096 units, and the last one has 1,000 units. These layers are followed by ReLU and the final FC layer generates class probabilities. Figure 2 displays the VGG19 architecture.

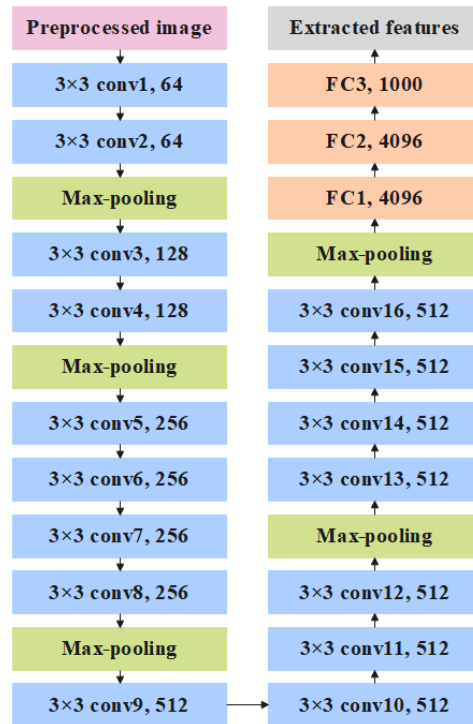


Figure 2. Architecture of VGG19

## 2.4. Similarity measure

The similarity is determined by comparing images and the similarity among image images is detected when its neighbors are similar. The Jaccard is used to compare the similarity and image element divergences in various samples due to its efficiency. It is helpful in CBIR because it reflects the number of intersection sets when comparing dissimilar and similar images. It enables better discriminative images that have semantic content at a higher level, and it is beneficial in focusing on the semantic gap. The similarity is calculated by using the Jaccard similarity measure as (4).

$$JS(x, y) = \frac{|x \cap y|}{|x \cup y|} = \frac{|x \cap y|}{|x| + |y| - |x \cap y|} \quad (4)$$

Where  $JS$  is Jaccard similarity,  $|x|$  and  $|y|$  are number of elements in  $x$  and  $y$ , the  $|x \cap y|$  is the size of the intersection of set  $x$  and  $y$ ,  $|x \cup y|$  is the size of the union of set  $x$  and  $y$ . Initially, the similarity score among image graphs is initialized to NULL which denotes no images. Then, the model calculates a similarity score for every pair among images  $i$  and  $j$  and includes the result in the similarity score. Lastly, it returns the final similarity score. Therefore, the similarity between the actual image and query image results in better retrieval images. The suitable images related to query images are accomplished through feature extraction and similarity calculation. Figure 3 shows an example of image retrieval from the Caltech 256 dataset of the query image (Figure 3(a)) and the retrieved image (Figure 3(b)). Figure 4 shows an example of image retrieval from the Corel 1K dataset of the query image (Figure 4(a)) and the retrieved image (Figure 4(b)).

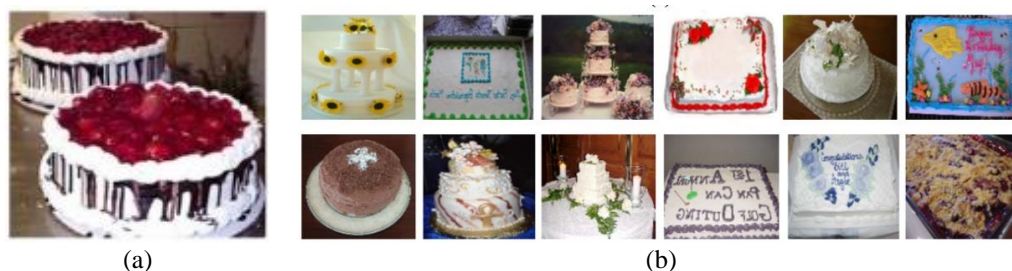


Figure 3. Sample of image retrieval from Caltech 256 dataset (a) query image and (b) retrieved images



Figure 4. Sample of image retrieval from Corel 1K dataset (a) query image and (b) retrieved images

### 3. RESULTS AND DISCUSSION

The proposed VGG19-Jaccard is implemented in the environment of Python 3.10 with system configuration of Windows 10 OS, Intel Core i5, and 8 GB RAM. The metrics such as recall, f-measure, and average precision are used to calculate the VGG19-Jaccard performance. These metrics are formulated in (5)-(7). Where  $TP$  and  $FN$  are true positive and false negative,  $R_n$  and  $P_n$  are recall and precision at  $n$ th threshold,  $R_{n-1}$  is a recall at  $n - 1$ th threshold.

$$Recall = \frac{TP}{TP+FN} \times 100 \quad (5)$$

$$F - measure = \frac{2 \times Average\ precision \times Recall}{Average\ precision + Recall} \times 100 \quad (6)$$

$$Average\ precision = \sum_n (R_n - R_{n-1}) P_n \times 100 \quad (7)$$

Table 2 displays the different feature extraction results with metrics of recall, f-measure, and average precision for Caltech 256 and Corel 1K datasets. The InceptionV1, MobileNetV3, and ResNet151 are taken as existing techniques to compare the result values of VGG19. The VGG19 achieves 99.5%, 99.3%, and 99.0% of recall, f-measure, and average precision for the Caltech 256 dataset. The VGG19 achieves 98.8, 98.8, and 99.8% of recall, f-measure, and average precision for the Corel 1K dataset.

Table 3 displays the different loss function results with metrics of recall, f-measure, and average precision for Caltech 256 and Corel 1K datasets. The mean square error (MSE), mean absolute error (MAE), and cross-entropy are taken as existing techniques to compare the result values of Jaccard. The Jaccard achieves 99.5, 99.3, and 99.0% of recall, f-measure, and average precision for the Caltech 256 dataset. The Jaccard achieves 98.8, 98.8, and 99.8% of recall, f-measure, and average precision for the Corel 1K dataset.

Table 4 displays the average precision by different distance measure results in terms of with and without VGG19 for Caltech 256 and Corel 1K datasets. Cosine and chi-square are taken as existing techniques to compare the result values of Jaccard. The Jaccard achieves 32.17 and 49.28 in terms of with and without VGG19 for the Caltech 256 dataset. The Jaccard achieves 31.57 and 50.70 in terms of with and without VGG19 for the Corel 1K dataset.

Table 5 displays the proposed methodology results with metrics of recall, f-measure, and average precision for Caltech 256 and Corel 1K datasets. The EfficientNet, DenseNet121, and CNN are taken as existing techniques to compare the result values of VGG19-Jaccard. The VGG19-Jaccard achieves 99.5, 99.3, and 99.0% of recall, f-measure, and average precision for the Caltech 256 dataset. The VGG19-Jaccard achieves 98.8, 98.8, and 99.8% of recall, f-measure, and average precision for the Corel 1K dataset.

Table 2. Results for different feature extraction

Dataset	Method	Recall (%)	F-measure (%)	Average precision (%)
Caltech 256	InceptionV1	96.5	96.7	96.8
	MobileNetV3	97.2	97.5	97.7
	Resnet151	98.3	98.4	98.6
	VGG19	99.5	99.3	99.0
Corel 1K	InceptionV1	97.3	97.0	97.6
	MobileNetV3	98.0	98.0	98.0
	Resnet151	95.8	96.0	96.1
	VGG19	98.8	98.8	99.8

Table 3. Result for different loss function

Dataset	Method	Recall (%)	F-measure (%)	Average precision (%)
Caltech 256	MSE	97.0	96.8	96.5
	MAE	96.5	96.6	96.8
	Cross entropy	96.8	96.7	96.6
	Jaccard	99.5	99.3	99.0
Corel 1K	MSE	97.5	97.3	97.1
	MAE	96.8	97.5	96.9
	Cross entropy	96.9	97.1	97.0
	Jaccard	98.8	98.8	99.8

Table 4. Result for average precision with different distance measure

Dataset	Method	With VGG19	Without VGG19
Caltech 256	Cosine	35.56	56.39
	Chi-square	35.25	57.63
	Jaccard	32.17	49.28
Corel 1K	Cosine	36.92	69.75
	Chi-square	40.17	68.08
	Jaccard	31.57	50.70

Table 5. Result of the proposed methodology

Dataset	Method	Recall (%)	F-measure (%)	Average precision (%)
Caltech 256	EfficientNet	98.6	98.6	98.8
	DenseNet121	98.4	98.4	98.6
	CNN	97.9	97.9	98.1
	VGG19-Jaccard	99.5	99.3	99.0
Corel 1K	EfficientNet	98.2	98.2	98.3
	DenseNet121	97.9	97.9	98.0
	CNN	97.5	97.5	97.7
	VGG19-Jaccard	98.8	98.8	99.8

### 3.1. Comparative analysis

In this context, the comparative analysis with the proposed VGG19-Jaccard is presented in terms of various metrics such as recall, f-measure, and average precision. The existing methods such as LB-CBIR [17] and Two-stage CBIR [19] are considered which is compared with the proposed VGG19-Jaccard for both datasets such as Caltech-256 and Corel 1K. The VGG19-Jaccard achieves 99.5, 99.3, and 99.0% of recall, f-measure, and average precision for the Caltech 256 dataset. The VGG19-Jaccard achieves 98.8, 98.8, and 99.8% of recall, f-measure, and average precision for the Corel 1K dataset. Table 6 displays the comparative analysis.

Table 6. Comparative analysis

Dataset	Method	Recall (%)	F-measure (%)	Average precision (%)
Caltech 256	LB-CBIR [17]	NA	NA	32.12
	Two-stage CBIR [19]	0.24	0.38	NA
	VGG19-Jaccard	99.5	99.3	99.0
Corel 1K	LB-CBIR [17]	NA	16.83	99.2
	VGG19-Jaccard	98.8	98.8	99.8

### 3.2. Discussion

The limitations of existing methods and advantages of the proposed method are discussed in this section. The existing techniques such as LB-CBIR [17] do not concentrate and extract the high-level features that affect the model's flexibility. Multi-scale averaging local binary patterns [18] does not concentrate and extracts the high-level features that affect the model's flexibility. Two-stage CBIR [19] fails to detect the irregular shape images and reduces the interpretability which affects the performance of CBIR. The SIFT-KAZE [20] required a high number of objects to search. In the triplet loss function [21], generalization performance was reduced, and bridging the semantic gap problem occurred among low-level and high-level semantic features. To overcome these limitations, this research proposes a VGG19-Jaccard for CBIR. The VGG19 enables to capture of hierarchical features and is suitable for characterizing texture and fine details. The pre-trained weights of VGG19 allow to generation of discriminative and robust feature representations thereby enhancing CBIR performance. The Jaccard is used as a similarity measure among feature vectors by comparing the intersection and union of feature sets. It helps manage high-dimensional and sparse data which provides quick and accurate image retrieval.



#### 4. CONCLUSION

This research proposes a VGG19 with Jaccard for CBIR. The VGG19 is used for feature extraction that efficiently captures a hierarchical feature with more information in the image due to numerous convolutional layers. It helps create fine and layered vision characteristics. The Jaccard is used to compare the similarity and image divergence of various samples because of its effectiveness. Moreover, it enables better discriminative images that have semantic content at a higher level. The raw dataset images are resized into particular dimensions to ensure dataset similarity. Then, the normalization is used to uniform the values from different query results which generates values in 0 to 1 ranges. The VGG19-Jaccard achieves an average precision of 99.0 and 99.8% for Caltech 256 and Corel 1K datasets. In the future, other neural networks will be discovered to further enhance the CBIR performance.

#### FUNDING INFORMATION

Authors state no funding involved.

#### AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rajath Arakere	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Narayanaswamy														
Vidyalakshmi Krishne		✓				✓		✓	✓	✓	✓	✓	✓	
Gowda														

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

#### CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

#### DATA AVAILABILITY

The data that support the findings of this study are openly available in [Kaggle] at “Caltech 256 dataset.” <https://www.kaggle.com/datasets/jessicali9530/caltech256>. and “Corel 1K dataset.” <https://www.kaggle.com/datasets/elkamel/corel-images>.

#### REFERENCES




- [1] P. Shamna, V. K. Govindan, and K. A. Abdul Nazeer, “Content-based medical image retrieval by spatial matching of visual words,” *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 58–71, 2022, doi: 10.1016/j.jksuci.2018.10.002.
- [2] G. K. Raju, P. Padmanabham, and A. Govardhan, “Enhanced content-based image retrieval with trio-deep feature extractors with multi-similarity function,” *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 6, pp. 511–525, 2022, doi: 10.22266/ijies2022.1231.46.
- [3] M. M. Monowar, M. A. Hamid, A. Q. Ohi, M. O. Alassafi, and M. F. Mridha, “AutoRet: a self-supervised spatial recurrent network for content-based image retrieval,” *Sensors*, vol. 22, no. 6, 2022, doi: 10.3390/s22062188.
- [4] M. Rashad, I. Afifi, and M. Abdelfatah, “RbQE: an efficient method for content-based medical image retrieval based on query expansion,” *Journal of Digital Imaging*, vol. 36, no. 3, pp. 1248–1261, 2023, doi: 10.1007/s10278-022-00769-7.
- [5] A. Rahman, E. Winarko, and K. Mustofa, “Content-based product image retrieval using squared-hinge loss trained convolutional neural networks,” *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5804–5812, 2023, doi: 10.11591/ijece.v13i5.pp5804-5812.
- [6] F. A. Alghamdi, “An effective hybrid framework based on combination of color and texture features for content-based image retrieval,” *Arabian Journal for Science and Engineering*, vol. 49, no. 3, pp. 3575–3591, 2024, doi: 10.1007/s13369-023-08087-y.
- [7] S. Fadaei, “New dominant color descriptor features based on weighting of more informative pixels using suitable masks for content-based image retrieval,” *International Journal of Engineering, Transactions B: Applications*, vol. 35, no. 8, 2022, doi: 10.5829/IJE.2022.35.08B.01.
- [8] S. Tena, R. Hartanto, and I. Ardiyanto, “Content-based image retrieval for traditional Indonesian woven fabric images using a modified convolutional neural network method,” *Journal of Imaging*, vol. 9, no. 8, 2023, doi: 10.3390/jimaging9080165.






- [9] S. Agrawal, A. Chowdhary, S. Agarwala, V. Mayya, and S. Kamath, "Content-based medical image retrieval system for lung diseases using deep CNNs," *International Journal of Information Technology (Singapore)*, vol. 14, no. 7, pp. 3619–3627, 2022, doi: 10.1007/s41870-022-01007-7.
- [10] J. A. G. Sweetta and B. Sivagami, "SOM based multi-stage CBIR method using the features of CPVTH, POPMV and MDLBP integrated with multi-query," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, pp. 622–633, 2024, doi: 10.22266/IJIES2024.0831.47.
- [11] F. Ahmad, "Deep image retrieval using artificial neural network interpolation and indexing based on similarity measurement," *CAAI Transactions on Intelligence Technology*, vol. 7, no. 2, pp. 200–218, 2022, doi: 10.1049/cit2.12083.
- [12] V. S. Mahalle, N. M. Kandoi, and S. B. Patil, "A powerful method for interactive content-based image retrieval by variable compressed convolutional info neural networks," *Visual Computer*, vol. 40, no. 8, pp. 5259–5285, 2024, doi: 10.1007/s00371-023-03104-5.
- [13] M. C. Chiu, Y. H. Lee, and T. M. Chen, "Integrating content-based image retrieval and deep learning to improve wafer bin map defect patterns classification," *Journal of Industrial and Production Engineering*, vol. 39, no. 8, pp. 614–628, 2022, doi: 10.1080/21681015.2022.2074155.
- [14] N. Keisham and A. Neelima, "Efficient content-based image retrieval using deep search and rescue algorithm," *Soft Computing*, vol. 26, no. 4, pp. 1597–1616, 2022, doi: 10.1007/s00500-021-06660-x.
- [15] E. Breznik, E. Wetzter, J. Lindblad, and N. Sladoje, "Cross-modality sub-image retrieval using contrastive multimodal image representations," *Scientific Reports*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-024-68800-1.
- [16] R. A. Narayanaswamy, V. K. Gowda, and G. N. K. Murthy, "Content-based image retrieval using the 2-dimensional convolutional neural network," *International Journal of Intelligent Engineering and Systems*, vol. 17, no. 4, pp. 67–76, 2024, doi: 10.22266/IJIES2024.0831.06.
- [17] F. Taheri, K. Rahbar, and P. Salimi, "Effective features in content-based image retrieval from a combination of low-level features and deep Boltzmann machine," *Multimedia Tools and Applications*, vol. 82, no. 24, pp. 37959–37982, 2023, doi: 10.1007/s11042-022-13670-w.
- [18] S. Fadaei, A. Dehghani, and B. Ravaei, "Content-based image retrieval using multi-scale averaging local binary patterns," *Digital Signal Processing: A Review Journal*, vol. 146, 2024, doi: 10.1016/j.dsp.2024.104391.
- [19] W. Wang, P. Jiao, H. Liu, X. Ma, and Z. Shang, "Two-stage content based image retrieval using sparse representation and feature fusion," *Multimedia Tools and Applications*, vol. 81, no. 12, pp. 16621–16644, 2022, doi: 10.1007/s11042-022-12348-7.
- [20] S. B. Geetha, R. Muthukumar, and V. Seenivasagam, "Enhancing scalability of image retrieval using visual fusion of feature descriptors," *Intelligent Automation and Soft Computing*, vol. 31, no. 3, pp. 1737–1752, 2022, doi: 10.32604/IASC.2022.018822.
- [21] K. Rahbar and F. Taheri, "Enhancing image retrieval through entropy-based deep metric learning," *Multimedia Tools and Applications*, 2024, doi: 10.1007/s11042-024-19296-4.
- [22] J. Li, "Caltech 256 dataset," Kaggle, 2018, [Online]. Available: <https://www.kaggle.com/datasets/jessicali9530/caltech256>.
- [23] A. Elkamel, "Corel 1K dataset," Kaggle, 2020, [Online]. Available: <https://www.kaggle.com/datasets/elkamel/corel-images>.
- [24] R. Hidayat, A. Harjoko, and A. Musdholifah, "A robust image retrieval method using multi-hierarchical agglomerative clustering and Davis-Bouldin index," *International Journal of Intelligent Engineering and Systems*, vol. 15, no. 2, pp. 441–453, 2022, doi: 10.22266/ijies2022.0430.40.
- [25] S. Mohsen, A. M. Ali, E. S. M. El-Rabaie, A. Elkaseer, S. G. Scholz, and A. M. A. Hassan, "Brain tumor classification using hybrid single image super-resolution technique with ResNext101\_32× 8d and VGG19 pre-trained models," *IEEE Access*, vol. 11, pp. 55582–55595, 2023, doi: 10.1109/ACCESS.2023.3281529.

## BIOGRAPHIES OF AUTHORS



**Rajath Arakere Narayanaswamy**    is an Assistant Professor of Computer Science and Engineering at GSSS Institute of Engineering and Technology for Women, Mysuru and has a teaching experience of 13 years. He pursuing is a Ph.D. in Information Retrieval, and Computer Vision at Sri Siddhartha Institute of Technology, SSAHE University, Tumakuru, specializing in computer vision, information retrieval, machine learning, deep learning, and image processing. He is passionate about teaching and is dedicated to mentoring undergraduate and graduate students in their research endeavors. He can be contacted at email: rajath191985@gmail.com.



**Vidyalakshmi Krishne Gowda**    received a B.E. degree in 2005 at Sri Siddhartha Institute of Technology, Tumakuru, Karnataka, an M.Tech degree in 2011 at the National Institute of Engineering, Mysuru, Karnataka, and a Ph.D in 2020 from VTU Belagavi, Karnataka. She is currently working as an Assistant Professor in the Department of Information Science and Engineering and has a teaching experience of 12 years. Her research interest includes Computer Networks, wireless communications, and Image processing. She has 4 papers and 6 conferences published in peer-reviewed International Journals in her credit and has received the best paper award (IEEE-2019) a research paper has been published as a chapter in a book called HCMC. As an extension of her work and knowledge, she also is guiding Ph.D. students in the same field. She can be contacted at email: vidyalakshmi@ssit.edu.in.