Complex & Intelligent Systems (2023) 9:1729–1751
https://doi.org/10.1007/s40747-022-00866-8

ORIGINAL ARTICLE

Deep learned vectors' formation using auto-correlation, scaling, and derivations with CNN for complex and huge image retrieval

Ahmad Naeem[1] · Tayyaba Anees[2] · Khawaja Tehseen Ahmed[3] · Rizwan Ali Naqvi[4] · Shabir Ahmad[5] · Taegkeun Whangbo[5]

Received: 20 February 2022 / Accepted: 21 August 2022 / Published online: 8 October 2022

Abstract

Deep learning for image retrieval has been used in this era, but image retrieval with the highest accuracy is the biggest challenge, which still lacks auto-correlation for feature extraction and description. In this paper, a novel deep learning technique for achieving highly accurate results for image retrieval is proposed, which implements a convolutional neural network with auto-correlation, gradient computation, scaling, filter, and localization coupled with state-of-the-art content-based image retrieval methods. For this purpose, novel image features are fused with signatures produced by the VGG-16. In the initial step, images from rectangular neighboring key points are auto-correlated. The image smoothing is achieved by computing intensities according to the local gradient. The result of Gaussian approximation with the lowest scale and suppression is adjusted by the by-box filter with the standard deviation adjusted to the lowest scale. The parameterized images are smoothed at different scales at various levels to achieve high accuracy. The principal component analysis has been used to reduce feature vectors and combine them with the VGG features. These features are integrated with the spatial color coordinates to represent color channels. This experimentation has been performed on Cifar-100, Cifar-10, Tropical fruits, 17 Flowers, Oxford, and Corel-1000 datasets. This study has achieved an extraordinary result for the Cifar-10 and Cifar-100 datasets. Similarly, the results of the study have shown efficient results for texture datasets of 17 Flowers and Tropical fruits. Moreover, when compared to state-of-the-art approaches, this research produced outstanding results for the Corel-1000 dataset.

Keywords CBIR · BOW · Deep learning · Image retrieval

✉ Rizwan Ali Naqvi
rizwanali@sejong.ac.kr

✉ Taegkeun Whangbo
tkwhangbo@gachon.ac.kr

Ahmad Naeem
f2019288007@umt.edu.pk

Tayyaba Anees
tayyaba.anees@umt.edu.pk

Khawaja Tehseen Ahmed
tehseen@bzu.edu.pk

Shabir Ahmad
shabir@gachon.ac.kr

[1] Department of Computer Science, University of Management and Technology, Lahore 54000, Pakistan

[2] Department of Software Engineering, University of Management and Technology, Lahore 54000, Pakistan

[3] Department of Computer Science, Bahauddin Zakariya University, Multan 60800, Pakistan

[4] Department of Unmanned Vehicle Engineering, Sejong University, Seoul 05006, Korea

[5] Department of IT Convergence Engineering, Gachon University, Gyeonggi-do, Seongnam-si 11320, Korea

Introduction

In traditional CBIR techniques, color, shape and texture-based features are used to search relevant images [1, 2]. Furthermore, it uses hand-crafted local-level feature descriptors such as speed-up robust features (SURF) [1] and scale-invariant feature transform (SIFT) [2]. In the recent era, the exponential growth of images occurs and there is a dire need for an image retrieval system that can work on a large-scale database [3]. Deep learning for image retrieval has been used in this era and retrieving images with the highest accuracy is the biggest challenge, which still lacks auto-correlation for feature description and extraction [4]. Convolution neural network (CNN) works efficiently on gigantic datasets due to this; it becomes the first choice of the computer vision community due to its high performance [5, 6]. Effective image analysis is based on correct image feature extraction and description [7]. In recent years, humans were surpassed by the deep learning algorithms for object recognition tasks by reducing the semantic gap. Therefore, the best image semantics are defined by implementing content base image retrieval with deep learning applications [4].

The accuracy of image retrieval is directly influenced by image semantics and perception. Deep learning performance is compromised when the transition occurs between image classification and image retrieval [8]. Content-based image retrieval with a convolutional neural network focuses on testing and training sets for the establishment of the image signature. To create an effective image representation, deep learning trains the model and then tests the model using training and testing sets. Hence, training of the model has significant importance, and proper availability of patterns for the training set is a significant problem that affects the accuracy of test results. Therefore, the next step is object and shape identification after dealing effectively with the test model problem [9]. Moreover, proper efforts in image processing and statistical pattern recognition are required to perform efficient image analysis. Furthermore, it is not necessary to put focus on a small area of an image instead of this fetching result of different object types is an essential requirement. To detect deep image features, primitive image components with global features are determined. Feature representation and similarity measure are the two key factors on which the retrieval efficiency of CBIR is formed, for the decades these factors have been extensively studied by researchers [10, 11].

The semantic gap remains one of the most difficult issues of CBIR research, because there is a difference between high-level semantics perceived by humans and low-level pixels captured by the machine. The promising result can only be obtained when the semantic gap between high-level perception and low-level image pixels perception is reduced [12]. Therefore, in the field of computer vision, research has been focused on the issues of image recognition and detection of image content. For the identification of complex and cluttered objects, deep learning is utilized which work on the principles of biological neural network which gain ample power from fully connected layers and through multiple stages of transformation, it processes the information [13]. Several recent studies show promising results of deep learning techniques in several applications, such as object detection [14], speech recognition [15], human pose estimation [16], and natural language processing [17]. The image retrieval from large datasets critically depends on deep learning features [18] as they have characteristics of spatial and texture detection attributes used for the classification and indexing of a large number of images [19, 20]. These attributes have a direct impact on precision [21, 65].

Inspired by the achievements of the deep learning model, this paper incorporates the proposed feature extraction technique with VGG-16 architecture. This research presents a new approach that uses auto-correlation, corner response computation, gradient computation, scale selection, Gaussian second-order derivative, box filtering, and localization. It utilized auto-correlation to analyze the composition of an image; it compares all potential pixel intensities and records their probabilities. Furthermore, corner/edge response is applied to detect the corner and edges of a possible object in an image. The gradient computation has been used to smoothen the detected edges and retrieve strong features and texture matching from the image. Whereas, scale selection is incorporated for scale-invariant representation of the image features. The suppression achieved by the Gaussian second-order derivative is fed into the box filter which recognizes the content of the image and establishes the foundation of the image's salient objects. The Gaussian smoothing has been used to smoothen the detected edges. Box filtering is applied to recognize the content of the image which establishes the foundation of the image's salient features. Moreover, the object and its boundary in an image are identified using localization. Furthermore, the information about the background and foreground objects is obtained through various levels of scaling. The spatial color coordinates are determined for all channels at the second stage of normalization to efficiently present the large feature vector present in the created signature. Principal component analysis (PCA) is applied to reduce these feature vectors. The VGG-16-based feature vectors are fused with the prominent features retrieved by the selected algorithms. These algorithms can easily fetch the features from challenging image datasets with complex foreground and background textures and they also merge the object of image with these strong features. For effective image indexing and retrieval, a bag of a word gets these image features as an input. Benchmark datasets with complex images are selected for result evaluation. The included datasets are Cifar-100 (100), Cifar-10 (10), Tropical fruits (15), 17 Flowers (17), Oxford (17), and Corel-1000 (10), and these datasets comprise millions of images collectively. The proposed method surpassed the existing methods and shows impressive results that demonstrated the effectiveness of the proposed method. The results are presented in both graphical and tabular formats. The main contribution of the research is discussed as follows.

- The proposed method fused the color, texture, and shape-based feature with VGG-16 generated features, which creates a better image signature.
- The proposed method improves VGG-16 architecture functionality through its internal coupling with primitive features to produce better performance.
- The feature detection criteria follow the simple step which includes auto-correlation, corner, and gradient response for the detection of feature elements.
- The effective feature extracted from the image contents is based on strength of scale selection, Gaussian second-order derivatives, filtering, and localization.
- In our proposed method, deep learning is used with both color and gray-scale features for image retrieval.
- The proposed method presented a novel technique for image revival, indexing and classification by combining the primitive features and fused deep features with bag of words (BoW) the assembly of the fused.

The remainder of the article is written like this: “Literature review” provides an overview of the literature review; “Proposed method” illustrates about methodology and the findings of the experiment and the results are presented in “Experimentation”. In “Conclusion”, the conclusion is discussed.

Literature review

A noticeable amount of research was performed in the field of content base image retrieval (CBIR) using deep learning. The interest points are detected through the detection of corners, edges, shapes, contours, ridges, and semantic and visual content interpretation. The performance of current CBIR systems suffers due to variation in the image datasets. Furthermore, a lot of effort is needed to build a dataset for training purposes and clear knowledge of the formula used for image retrieval is needed which refined the classification errors [3]. Two pictures taken apart from one another are not equivalent. For this purpose, a method is proposed which utilizes CNN to determine similarities in identical images [4]. Moreover, the geo-location is applied for the fine-tuning of images which led to an unequaled separation of images from each other [5]. Moreover, a small size codebook was the core issue of VLAD a method introduced in [22] which uses original clusters to generate primary and secondary residual codebooks. VLAD gains the same dimension by applying two-step aggregations. In comparison to this, [23] developed an approach that constructs a hidden layer of a visual vocabulary for VLAD, which consists of subwords for every word in the visual vocabulary. Image descriptor makes the same dimension as original VLAD when an aggregate of coarse layer and hidden layer vocabulary are compiled. However, a different idea was proposed by [24] which implements two compact bilinear representation approaches, that were based on bilinear pooling and end-to-end optimization for image recognition. In addition to this, a deep learning technique with aggregation and a cross-dimensional weighting was suggested in [25]. However, [26] utilizes AlexNet, and bag for feature extractor while SVM and Quadratic SVM classifiers. A novel technique generates durable feature vectors using a convolution layer for to detection of multiple image regions [27]. An image retrieval technique is proposed in [28] which implemented a simple similarity measure (SSM) that gets combined information from all relevant partition levels using Voronoi-based VLAD (VVLAD) and Voronoi-based CNN (VDCNN). In [29], a method has been proposed which concatenates multiple layers of CNN to enhance the performance of image retrieval. For the effective image retrieval, [30] proposed a patch-level descriptor was used to adopt a convolution kernel network, which performs better than a supervised convolutional network and time training is also fast. Whereas, a new technique is introduced [31] which first refines noisy dataset then built a differentiable deep architecture based on R-MAC descriptor and lastly trains the R-MAC based model on Siamese architecture for retrieval task. The new CNN techniques used in the CBIR sector showed incredible performance using BOW analysis. The interpretation of concatenated image features was obtained from various levels to ensure that no image features from the lower CNN layers were explicitly extracted for the work of CBIR [32]. While in [33], two parallel CNN were used with CBIR for extracting features from images that were trained on benchmark datasets. Layers of pre-trained CNN models collect both relevant and irrelevant features. Whereas, a weakly supervised metric learning algorithm is proposed in [34] for images associated with user-provided tags which were used for the social image retrieval. Information from society is used for retrieving images by employing user provide tags based on a weakly supervised learning algorithm. Furthermore, a method has been proposed in [35] which enhances the retrieval performance by utilizing the generalized mean pooling layer and feeding it to the VGG to attain a state-of-the-art performance. A novel idea was proposed in [36] which utilizes several different image retrieval methods for CNN activations. The achieved quality of image retrieval is an indication of the generalization of CNN which uses the ImageNet data set to classify images by reducing error. CNN was widely used due to its performance achieved for image recovery. Whereas, in [37], a method was proposed which uses CNN to achieve the goal of image recognition by utilizing the more central dataset. This work teaches several objects to boost the image retrieval efficiency for specific landmarks and buildings using deep learning. An end-to-end BOWs model with deep learning is introduced in [38]. This model identifies and separates the object from an input image; moreover, it has a high power of semantic discrimination for visual words. This model also generates feature maps for different object categories by using convolution layers and then makes a visual word for every feature map. In comparison to this, [39] proposed a technique that measures similarity between classes by calculating pair-wise dot product and mapping images into relevant classes. Furthermore, a deterministic algorithm trained on wordnet was used for calculating class centroids. However, Deep Linear Discriminant Analysis Hashing was introduced in [40], which utilizes hash labels and hash models. Hash labels were used to extract features from images, whereas reduction of image features can be done using an objective function based on linear discriminant analysis. Similarly, hash model deep learning models were trained using hash labels. Hash model trains the deep learning network using generated hash labels and getting discriminative hash codes for image training. For the fast image retrieval, deep hash model is used to map features to hash code. In comparison to this, [41] proposed a method for image retrieval that constructs candidate regions by combining color and texture information and feeding them to the deep neural network. Furthermore, for calculating the distance between different images, a similarity function is utilized. Whereas, the deep belief network method of deep learning is utilized in [41] for image extraction and classification. A novel idea was proposed by [42] in which object recognition and spatial color feature with shape features are extracted through the fusion between them. The combination of color with shape can more accurately recognize the object. Both RGB and gray-level images are used to extract color features and extract object edges and corners. CNN architectures are the most commonly used for training and classification tasks, in which descriptions are taken from upper layers of CNN networks which have been used to detect semantic features for classification at the class level. The transfer learning is used to fetch generic features of CNN which shows a significant performance for image retrieval [43]. An image retrieval technique is suggested in [44] in which a transfer learning has been applied to a dissimilar dataset with proper fine-tuning, the dataset consider at the time of the test gives improved results. Principal Component Analysis (PCA) is the common technique for dimension reduction which was formed from local feature-based methods. Some deep learning approaches perform CBIR-related tasks by learning appropriate features of images, the use of CNN for feature extraction can improve the efficiency of image retrieval and outperform many hand-crafted features. Moreover, some researcher creates image vectors using the combination of CNN, bag of words, and LBP [45].

The proposed method expresses the best image retrieval method. The Gaussian smoothing is applied after auto-correlation. Corner/edge response, image gradients computation, and box filtering are applied to extract the features of an image. Feature reduction is achieved by utilizing CNN, whereas VGG is incorporated for image classification. The spatial mapping was applied to normalized images for indexing and searching purposes. The present technique extracts the deep features and fused them with the deep feature vectors. The combination of these strong features is used to retrieve objects from diverse and cluttered datasets. Moreover, BoW architecture was used for efficient image retrieval and indexing.

Proposed method

Autocorrelation

The auto-correlation technique works in a user-independent manner which analyses the composition and detected interest points of an image. Initially, auto-correlation compares all potential pixel intensities and records their probabilities of pixel intensities in an image; if the variation in image pixels intensities are all low, then the auto-correlation is deliberate as flat. Edge is detected when some pixel intensities are high and some are low; similarly, auto-correlation showed a corner when all pixel intensities are high [46]. Our technique integrates the strength of SURF for feature detection and description. The SURF [47] feature descriptor considers neighboring rectangular areas in a detection frame at a particular position, summarizes each region's pixel intensities, and measures the difference around the point of interest. Autocorrelation in a more abstract sense is logically equivalent to the convolution of a function to itself. In this work, it is assumed that the image in the spatial domain is digital which is defined as a discrete entity having 2 to 4 dimensions with a limited degree, having genuine, confined, and discrete values, and auto-correlation can be characterized by Eq. (1) [48]

$$G(ai, bi) = \sum_{x}^{M} \sum_{y}^{N} p(xi, yi) * p(xi - ai, yi - bi), \quad (1)$$

where G (ai, bi) denoted the auto-correlation function, whereas p (xi, yi) shows intensities of an image at a place (xi, yi), wherever ai and bi are the lag from the close positions of xi and yi. In a normalized auto-correlation, the value and scope of images are focused on the retrieval method and parameters rather than the fundamental structure, it is represented by *Gp*, which is utilized for examining image correlation spectroscopy. The value of *gp* is calculated by the following equations:

$$gp(a,b) = \frac{\sum_x^M \sum_y^N p(xi, yi) * p(xi - ai, yi - bi)}{\sum_x^M \sum_y^N p(xi, yi) * p(xi, yi)} - 1 = \frac{F^{-1}\{F[p(xi, yi)]^2\}}{NM(i)^2} - 1, \tag{2}$$

where *F* (*xi*, *yi*) is the Fourier transform of *p* (*xi*, *yi*). Practically, Eqs. (1) and (2) have become obsolete due to extensive research in this field, and through the use of the Weiner–Khinchin theorem (as shown in Eq. 3), autocorrelations can be calculated much more effectively using Fast Fourier transforms than Eqs. (1) & (2)

$$F^{-1}[Gp(a,b)] = S(p) = |F[p(x,y)]|^2, \tag{3}$$

where *S(p)* represents the power spectrum of the image. The inverse Fourier transform of auto-correlation function is equal to *S(p).* The spectrum strength has been determined by the squared size of (*p*'s) Fourier transform, which is analogous to the multiplication of the Fourier transform by its opposite for real-valued functions. There are two reasons why this theorem is significant. The processing time reduces [from $O = n^4$ to *n*log(*n*)], and second, the auto-correlation operator connects to the Fourier transform that is more frequently accepted.

The preceding terminology we are using for the analysis of the auto-correlation operator is based on the empirical formulation of two points which have been decided on the vector among them

$$Gp(ai, bi) = P(xi, yi) * P(xi + ai, yi + bi) \tag{4}$$

$$gp(ai, bi) = P(xi, yi) * P(xi + ai, yi + bi) - 1. \tag{5}$$

We concentrate on the predicted autocorrelations for statistically defined image classes (i.e., ambient noises with certain intensities). It can be defined for image classes as the sum of the expected pixel intensity distribution *P* (*xi*, *yi*) and *P* (*xi* + *a*, *yi* + *b*) describing the predicted pixel intensity distribution in the image. This provides projected auto-correlation of the image, *Gp* in Eq. (4) and expected standardized auto-correlation *gp* in Eq. (5) [48]. To differentiate between preprocessing and auto-correlation analysis of images, it is important to explain the results of a variety of common processes for greater consistency and precision (Fig. 1).

Corner/edge response

To achieve the maximum accuracy for image classification, the quality of detected corner and edge has a high significance. For this purpose, we incorporated an edge and corner response which judges the quality of the corners and edges by picking up the pixels in the corner and thin edges [46]. First, consider the corner response Cr because of the variance in rotation, the function of α and β is need only. The use of *Ti*(*N*) and *Di*(*N*) in the formulation is desirable, as it prevents the decomposition of *N*'s value [46]

$$Ti(N) = \alpha + \beta = \mathring{A} + \mathcal{B} \tag{6}$$

$$Di(N) = \alpha\beta = \mathring{A}\mathcal{B} - \mathbb{C}^2. \tag{7}$$

Consider the following inspired formulation for the corner response:

$$\mathrm{Cr} = \mathrm{Det} - kTr^2. \tag{8}$$

Cr is positive in the edge areas and it is negative in the smooth region. The enhancement in the contrast increases the intensity of the response in every case. The area is considered as flat if *Tr* drops below some defined threshold value. The corner is identified in an image if its response is an 8-way local limit then. In addition, edge region pixels are classified as edges if their feedback is negative and it depends on how the first gradient amplitude is greater in the *x*- or *y*-direction [46].

Gradient computation

To extract information from the images, we introduced gradient computation. It creates gradient images by utilizing the original image. Each pixel of a gradient image is calculated by measuring the alteration in the intensity of a similar point in the actual image. Gradient images are computed in the *x*- and *y*-directions and they are commonly used for edge-identification [49]. Pixels with large gradient values become possible edge pixels and edges can be identified in the direction vertical to the direction of the gradient. The Canny edge detector algorithm works on the principles of image gradients. Our proposed method uses image gradients for strong features and matching texture. Moreover, various lighting effects or camera effects may result in extremely diverse pixel distributions in two images of an identical scene, which can cause a failure for algorithms that match identical image features. The matching errors can be reduced if the matching algorithm uses gradient images, whose estimating patterns were derived from the original images. Such patterns are less sensitive to lighting and camera adjustments, thereby reduc-

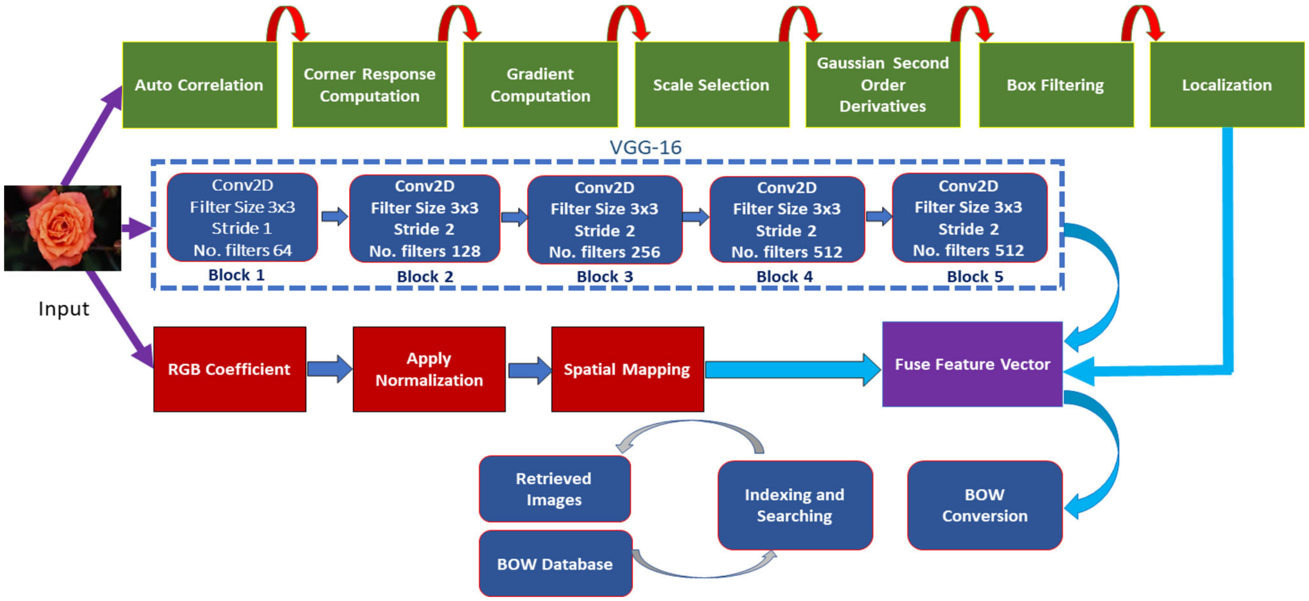

Auto Correlation
Corner Response Computation
Gradient Computation
Scale Selection
Gaussian Second Order Derivatives
Box Filtering
Localization
VGG-16
Conv2D Filter Size 3x3 Stride 1 No. filters 64 Block 1
Conv2D Filter Size 3x3 Stride 2 No. filters 128 Block 2
Conv2D Filter Size 3x3 Stride 2 No. filters 256 Block 3
Conv2D Filter Size 3x3 Stride 2 No. filters 512 Block 4
Conv2D Filter Size 3x3 Stride 2 No. filters 512 Block 5
Input
RGB Coefficient
Apply Normalization
Spatial Mapping
Fuse Feature Vector
Retrieved Images
BOW Database
Indexing and Searching
BOW Conversion

Fig. 1 Proposed methodology

ing matching errors [50]. An image's gradient is a vector of its matrices: as shown in Eq. (9)

$$\nabla\mathcal{F} = \left(\frac{\partial M}{\partial x}, \frac{\partial M}{\partial y}\right). \tag{9}$$

To describe this, we have to analyze partial derivatives. The image intensity changes with the change of x when the partial derivative of M *is* taken to x. Whereas, I (x, y) can be written as in Eq. 10

$$\frac{\partial M(x,y)}{\partial x} = \lim_{\Delta x \to 0} \frac{M(x+\Delta x, y) - M(x,y)}{\Delta x}. \tag{10}$$

In the distinct scenario, a one-pixel difference can be taken. Therefore, we can differentiate between M (x, y) and the pixels present before or after it. Moreover, by taking derivative with correlation the pixels before and after M (x, y) can be treated symmetrically, which can express in Eq. (11) [50] as

$$\frac{\partial M(x,y)}{\partial x} \approx \frac{M(x+1,y) - M(x-1,y)}{2}. \tag{11}$$

Similarly, we also compute

$$\frac{\partial M(x,y)}{\partial y} \approx \frac{M(x,y+1) - M(x,y+1)}{2}. \tag{12}$$

Scale selection

Different scales have been used by interest points. To handle the image structure at different scales, we have introduced scale selection, which provides a well-established architecture for multi-scale image structures. Image pyramids are normally made using scale-spaces and it is split into octaves. Gaussian was used repeatedly for image smoothing and to achieve a higher degree pyramid subsampling was performed. The octave layers of image pyramids enhance the computing efficiency for key point detection. The computation efficiency is enhanced by incorporating key points detected in the octave layers of the image pyramid [47]. The scale-space representation has an extremely useful feature of making image representation invariant to scale; for this purpose, it executes automatic selection of local scales depending upon local extrema over scales of the γ-normalized derivatives as expressed in Eq. (13)

$$\partial\varepsilon = t^{\gamma/2}\partial x,$$

and

$$\partial\eta = t^{\gamma/2}\partial y. \tag{13}$$

A scale selection module operating according to this principle can achieve the corresponding scale covariance property. A local maximum is assumed on a certain scale t_0 in an image for feature extraction, after that image is rescaled by a factor s, whereas local maxima were rescaled and transformed to scale level $s^2 t_0$ [51].

Gaussian second-order derivative

To achieve an enhanced image, we have introduced a second-order derivative. For the automatic edge detection, the second-order derivative is a more sophisticated approach, but is often prone to noise. Edge is detected by combing smoothing and gradient operations to suppress the noise. The Laplacian of Gaussian (LOG) is applied to the image which first smooths the image using a Gaussian smoothing, which reduces its noise sensitivity. The LOG creates a single grayscale image as an input and a binary image as an output. The zero-crossing detector finds positions in the Laplacian of an image from which the Laplacian's value passes through zero. These points also take place on the edges of images, i.e., points where the image amplitude varies in positions that are not associated with the edges [52]. The following equation shows the derivative of Laplacian:

$$\Delta^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}. \tag{14}$$

Box filter

Box filter refers to calculating the average-of-surrounding pixels in an image. It precisely recognizes the image content at an intermediate step which is a novelty of the proposed method. Box filter is essentially a convolution filter that is a widely used computational process for filtering images. Box filter gives us a way to multiply two arrays and created a third one. The image sample and the filter kernel are multiplied to obtain the filtering result in the box filter. The filter kernel defines the filtering type. The box filtering has the capability of sharpening, embossing, edge-identification, and smoothing the image. Box filter gives a small window with a fixed size which is generally larger than the pixel. This window dips through all positions over the image and measures the sum which it sees in its frame. Using a box filter, there has been no need for the same filter to be used iteratively on the output of previous filtered layers, instead of a box filter of any size that can directly apply to the original image. Hence, without reducing the size of an image, scale-space is evaluated by updating the filter size. A box filter of 9 × 9 is applied to estimate the Gaussian smoothing with a standard deviation $\theta_i = 1.2$ and it is the lowest scale for measuring the response. The box filter is denoted byB_{xx}, B_{yy}, B_{xy}. When weights are applied, the rectangular window becomes efficient to measure. The hessian determinant can be calculated as follows:

$$|\text{Hessian}| = B_{xx} B_{yy} - (d B_{xy})^2. \tag{15}$$

In Eq. (15), d represented the proportional weight to balance the Hessian determinant, which preserves the energy between Gaussian kernels using this technique. The hessian determinant is also used to describe image response at a specific location. Moreover, the filter responses are normalized using an appropriate scale [47].

Localization

In this approach, localization is introduced to detect the object and its boundaries in an image. The interest point localization has been achieved by applying a non-maximum suppression to 3 × 3 × 3 neighborhood. Moreover, over-scale interest points in the image have been localized [53]. The image space and scale are interpolated by the maximum determinant of the Hessian matrix [54]. In our case, scale-space exclamation is mainly significant as the difference in scale between the first layers of each octave is comparatively large.

VGG 16

To achieve maximum performance, the proposed VGG 16 architecture technology is paired with the presented feature detection and extraction technique. We have used the amazing and efficient image classification ability of VGG 16. The main contribution of VGG 16 is to use architecture with a small convolution filter of 3 × 3 for the detailed evaluation of the increasing depth network. The discriminative power of the model enhances by increasing the depth and this growing depth enables the utilization of more nonlinear activation. Therefore, a pooling size of 2 × 2 and 3 × 3 filter with spatial; footprints are always employed by VGG. At stride 1, the padding of 1 has been used for convolution. Whereas, max-pooling is done at stride 2. The spatial footprint of the output volume is preserved with the 3 × 3 filter and a padding of 1, although the combination also reduces the spatial footprint. Thus, the pooling was carried out in spatially overlapping areas, and the spatial footprint was therefore reduced by a factor of 2. In VGG-16 architecture after each max-pooling, the number of filters was always increased by a factor of 2. A significant improvement can be attained by increasing the depth from 11 (3 FC layers and 8 convolution layers) weighted layers to 19 (3 FC layers and 16 convolution layers) weighted layers. The architecture of VGG is very uniform and consists of 16 convolution layers and 138 million parameters. However, these parameters can handle through transfer learning. The width of the channel in the convolutions layer is small and it starts at 64; after every max-pooling layer, it increased by a factor of 2 and finished at 512. The fixed-size RGB image of 224 × 224 is used as an input. The filters of 3 × 3 were used to process the image which passes through the stack of the convolution layer. For the input channels of a linear transformation, a filter of 1 × 1 is also utilized. The stack of convolution layers was followed by the three fully connected (FC) layers: there were 4096 channels each in the first two layers, whereas 1000 channels have been present in the third layer. Across all networks, the configuration of the fully connected layers is the same. Rectification (Relu) non-linearity is fitted with all hidden layers. The spatial resolution after convolution is conserved by applying spatial

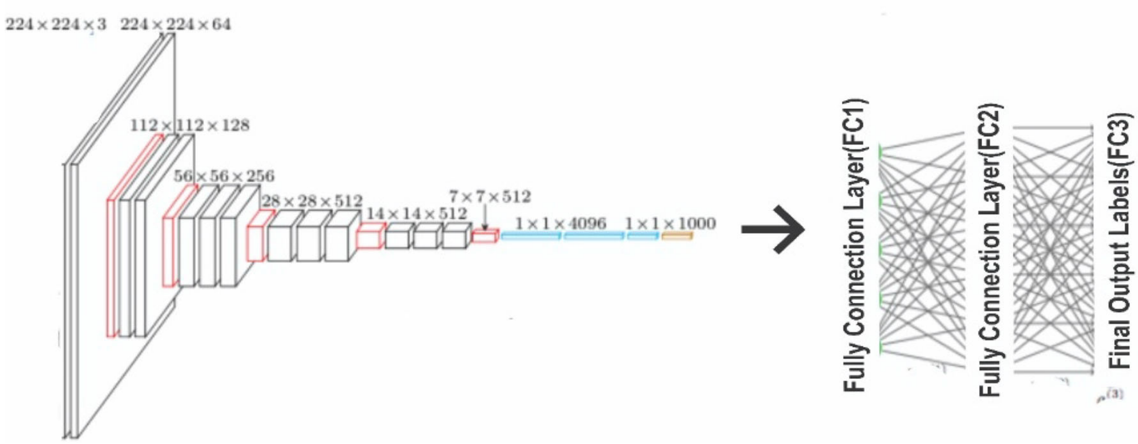


Fig. 2 VGG 16-Layers architecture

padding, whereas 1 pixel is utilized to convolution stride, i.e., for the 3 × 3 convolution layers one-pixel padding is applied. Five max pool layers follow some convolution layers which conduct spatial pooling. Using stride 2, max-pooling is accomplished through a window of 2 × 2 pixels. For the training process, an input image is cropped from the smallest side S of the rescaled training image. The value of S has not been less than 224. When S is equal to 224, the crop captures the statistics of the whole image whereas, if S is greater than 224, crop captures the small part which contains an object. Single and multi-scale approaches are used for training scale S. For single-scale training S is fixed, at two scales 256 and 384 with a smaller learning rate of 10^{-3}. Whereas, in multi-scale training, each image is randomly sampled from the particular range [$S_{\min} = 256$, $S_{\max} = 512$]. To identify objects on a wide range of scales, this type of training has been implemented. In the testing phase, the trained model and input images are classified by applying a rescaled smallest image side, which is known as the test scale and denoted as Q. Convolutional layers are made by utilizing the fully connected layers, which were applied to the whole image. The outcomes are class score maps based on the size of an input image. The class score is sum pooled to obtain the fixed-sized vector of class scores for an image. Initially, the learning rate started from 0.1, and when the error rose, it was divided by 10 [55]. The efficiency of a variety of computer vision applications such as face recognition and object identification has been improved with the use of VGG-16 (Fig. 2).

The proposed feature vectors create an influential image signature when they are fused with VGG-16 generated feature vectors that represent the shape and object characteristics. Therefore, the deep image features are demonstrated by the proposed algorithm. Bag of word (BoW) accepts these features and displays the resulting images utilizing k-nearest neighbors (KNN). For effective image retrieval from large datasets, the proposed method utilized principal component analysis (PCA) to reduce the image feature generated by bag of words (BoW).

Experimentation

The query image is a color image that is applied as an input to the proposed system. The convolution neural network takes a color image as input, whereas the color image is converted into greyscale for the proposed system. The images from benchmark datasets are used as input images. Image datasets are very important and selecting the correct data sets improves the efficiency as well as the accuracy of an image retrieval system. Most of the datasets are customized for a specific task, depending upon the nature of the task [65]. Some researchers use particular image categories, dependent on the domain. The reliability of the results is directly influenced by image characteristics such as color, the position of objects, consistency, scale, overlap, occlusion, and cluttering. The widely used datasets, which have diversified categories of images, spatial color and object information, object occlusion, and generic CBIR usage, are chosen for our experimentation process. The experiments were performed on six benchmark datasets included Cifar-100 [56], Cifar-10 (10) [56], Corel-1000 (10) [57], 17-Flowers [58], Tropical Fruits (FTVL) [59], and Oxford building [60]. The accuracy of the result is affected by the image characteristics, such as the location of the object, color, occlusion, size, and quality [61]. Both the proposed algorithm and the convolution neural network used a color image as a query image. For the proposed algorithm, color image is converted to gray scale. The image is picked as an input image from selected benchmark datasets. The training and testing ratios are set at 70% and 30%, respectively. The training time of VGG-16 is varied for each selected dataset, and it normally depends upon size, categories, and number of images. Whereas, the hardware also plays a vital role in training time. The training time varies to ~ 3–450 m when the research experiment was executed on a windows-based operating system with core i7 8th generation, 8 GB GPU NVIDIA GeForce GTX and 32 GB RAM with 15–75 epochs.

Assessment of precision and recall The accuracy of performance is evaluated on two metrics, precision and recall. The precision depends upon the positive predicted values, while recall measures the true-positive ratio. Equation (16) is used to calculate the precision for every category, whereas Eq. (17) [42] is used to calculate the recall for each category

$$\text{precision} = \frac{H_{w(n)}}{H_{u(n)}} \tag{16}$$

$$\text{recall} = \frac{H_{w(n)}}{H_o}, \tag{17}$$

where $H_{w(n)}$ describes the related images, which match with the query image, $H_{u(n)}$ describes the image retrieval against the query image, and H_o represents the total relevant images in a database.

Assessment of average retrieval precision (ARP) The graphical representation shows the Average Retrieval Precision of the proposed method. The graphs of ARP showed the performance of the proposed method on a variety of datasets. Equation (18) is utilized for computing ARP

$$\text{ARP} = \sum_{j=1}^{t} \frac{\text{AP}_j}{t}. \tag{18}$$

The average precision is represented by the AP and the total number of categories are represented by t. ARP is calculated for every category of the selected dataset. Irrespective of the category, the correct number of retrieved images is represented by the data bar in the ARP graph. When the number of categories is raised, the average precision is steadily lowered. The proposed method reports highest ARP for the Cifar-100 [56], Cifar-10 (10) [56], Corel-1000 (10) [57], 17-Flowers [58], Tropical Fruits [59], and Oxford building [60] datasets.

Assessment of average retrieval recall (ARR) The Average Retrieval Recall of the proposed method has been represented in graphical form. The graphs of ARR showed the performance of a proposed method for several datasets. Equation (19) is utilized for computing ARR

$$\text{ARR} = \sum_{j=1}^{t} \frac{\text{AR}_j}{t}. \tag{19}$$

The average precision is represented by the AR and the total number of categories are represented by t. ARR is calculated for every category of the selected dataset. The calculation of ARR values is ordered in ascending order to draw the ARR graph. The remarkable ARR rates are displayed in several datasets by the proposed method.

Experimentation on large benchmark Cifar 10 and Cifar 100 datasets

We test our model with large benchmark datasets of cifar. There are two separate datasets of cifar that exist Cifar 10 and Cifar 100. Cifar 100 is based on hundred different classes and each class contains 500 training images and 100 testing images. On the other hand, cifar 10 consists of ten classes and each class contains 5000 training images and 1000 testing images. Both datasets contain 50,000 images for training purposes and 10,000 images for testing. The resolution of cifar datasets has been resized to "tiny" and their resolution is 32 × 32 pixel. The word "tiny" refers only to resolution. A sampling of these datasets has been done from the same source but no data and class of these two datasets have overlapped. Cifar datasets have been ranked as the most popular benchmark datasets, which allow CNN to train fast, due to their low image resolution and manageable size. One of the limitations is duplicate images for testing sets [56]. Cifar 10 and cifar 100 are widely used by researchers for image retrieval. Liu et al. [5], proposed Deep Linear Discriminant Analysis Hashing (DLDAH) uses cifar 100 and cifar 10 to validate its results.

The Cifar-10 dataset consists of ten different semantic groups which are categorized as birds, frogs, ships, dogs, cars, cats, aero planes, horses, deer, and trucks. Each category contains 600 images. The results with the highest output are therefore inevitable. At this stage, the computational load is a major factor. Our technology adapted it in the following three stages: first, it accurately detects the interest point, and then, it performs the feature matching and reduction. The auto-correlation is used to detect the potential pixel intensities for calculating interest points, second, an appropriate scale is selected and subsampling is also performed, and in the third step, box filtering is applied to calculate the average-of-surrounding pixels in an image. Prominent features are extracted by applying three levels of work. Moreover, a low computational load provides a quick user response. Feature detection, extraction, fusion, CNN extraction, and BOW indexing of images took an average of 0.021 to 0.016 s.

The proposed method uses all ten categories of the Cifar-10 data set. Moreover, the highest average precision ratio is achieved by the proposed method. The use of deep learning features in the proposed method helps in the correct classification of images. Image auto-correlation, scaling, and filtering with deep features enabled the correct classification of the image from a wide range of image semantics on all categories of cifar 10, such as airplane, automobile, bird, cat, deer, frog, horse, ship, and truck, as shown in Fig. 3. The

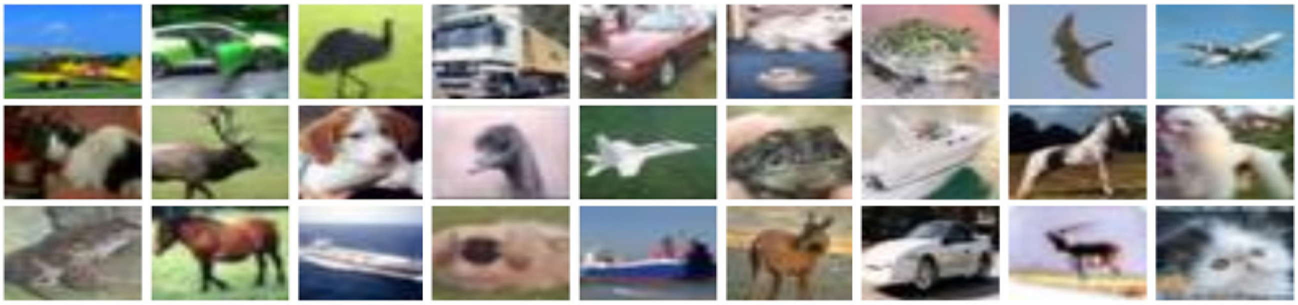
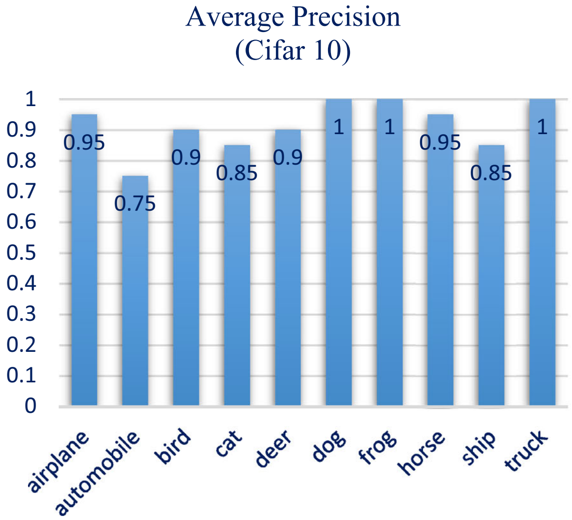
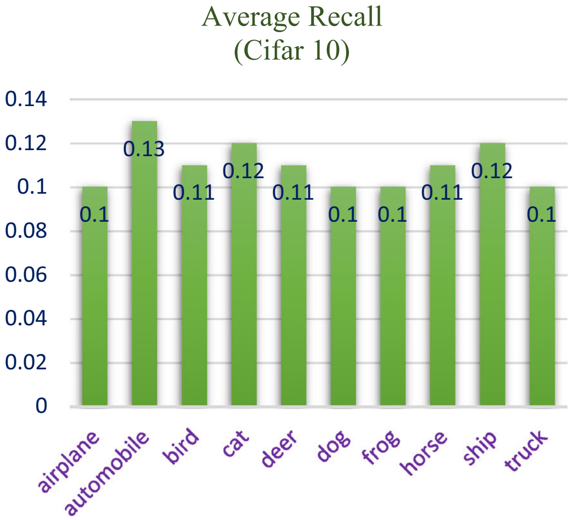


Fig. 3 Cifar 10 dataset sample images



(a)Average Precision



(b) Average Recall

Fig. 4 **a** Average precision. **b** Average recall

Table 1 Average precision for cifar 10 dataset

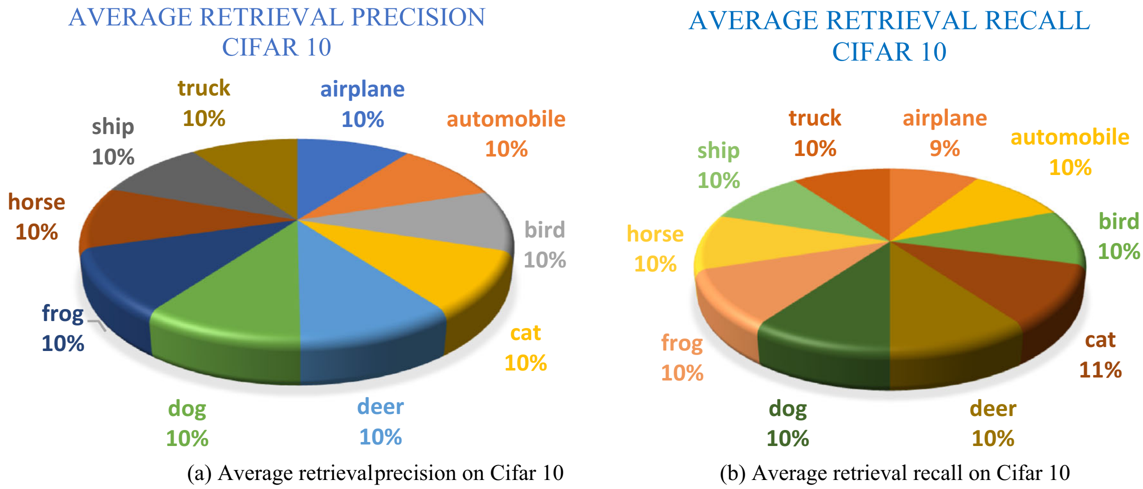
Cifar 10 average precision										
Categories	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Precision	0.95	0.75	0.90	0.85	0.90	1.00	1.00	0.95	0.85	1.00

mean average precision achieved by the proposed system is 95.6%, and in certain categories, such as dog, frog, and truck, the proposed approach also produced a better average precision performance as shown in Fig. 4a that shows the highest average precision for the all categories of cifar 10, whereas Fig. 4b shows the best recall ratio.

The average precision (AP) for the cifar10 dataset is presented in Table 1. The promising result has been attained using the proposed method for the cifar 10 dataset. The strength of the proposed method is shown in image categories, such as dog, frog, and truck, where it achieves a 100% precision ratio. Whereas, airplanes and horses achieve a 95% AP ratio, while birds and deer achieve a 90% AP ratio. In the cat and ship category, the proposed method achieves an 85% AP ratio in the category of cats. The category of automobile achieves a 75% AP ratio. The mAP of 91.5% has been achieved by the proposed method for the ten categories of the cifar 10 dataset.

Figure 5a shows the Average Retrieval Precision (ARP) on the cifar 10 dataset. The highest ARP ratio was achieved by the proposed method for the category of airplane and frog. Furthermore, outstanding performance has also been achieved by other categories of the cifar 10 dataset. Figure 5b shows and Average Retrieval Recall (ARR) of the cifar 10 dataset. The ARR rate is 10% for automobile, bird, deer, dog, frog, horse, ship, and truck, whereas an 11% ARR ratio has been measured for the category of cat.

The cifar 100 has also contained 60,000 images of 32 × 32 RGB color images same as cifar 10. Cifar 100 has 100 categories that belong to different semantic groups, such as apple,



(a) Average retrievalprecision on Cifar 10

(b) Average retrieval recall on Cifar 10

Fig. 5 **a** Average retrieval precision on Cifar 10. **b** Average retrieval recall on Cifar 10

bridge, can, flatfish, hamster, lamp, motorcycle, oak tree, and plain. Road, seal, tulip, etc. Each category of cifar 100 contains 600 images. Outstanding results have been achieved by the proposed method in most of the categories of the cifar 100 dataset. Sample images of the cifar 100 dataset are shown in Fig. 6.

In the complex categories of the cifar-100 dataset, proposed method gives an outstanding result and it achieves more than 80% AP in many categories of datasets, which is shown in Table 2. The proposed method utilizes the combination of CNN with auto-correlation and Gaussian smoothing for the classification of images that belongs to different semantic groups. Moreover, the proposed method achieves 100% average precision on most of the categories of the cifar-100 dataset. The mean average precision (mAP) of 75% has been shown by the proposed method. In addition to this, promising ARP and ARR ratios have been provided by the proposed method, as shown in Table 2.

The graphical representation of precision and recall is shown in Fig. 7a and b, whereas Fig. 7c shows ARP. The exceptional ARP rates are shown by the proposed method for the Cifar-100 dataset.

The proposed method showed an outstanding result in all categories of cifar 100. Moreover, the proposed method achieved above 80% average precision which shows the strength of the proposed method.

Experimentation on texture dataset

The 17-Flower [58] and FTVL [59] are the complex benchmark datasets to classify and categorize images. The primary use of these datasets is to classify texture images belonging to different semantic groups. In the domain of content base image retrieval, the number of images plays an important role in image classification. To test the effectiveness of the proposed method, 17 Flower dataset is selected which has variation in light and pose. Furthermore, it consists of 17 categories which contain 1360 images of flowers and each category contains 80 images. The 17 Flower dataset consists of images of different semantic groups. For the classification of images, the categories of the 17 Flowers dataset provide

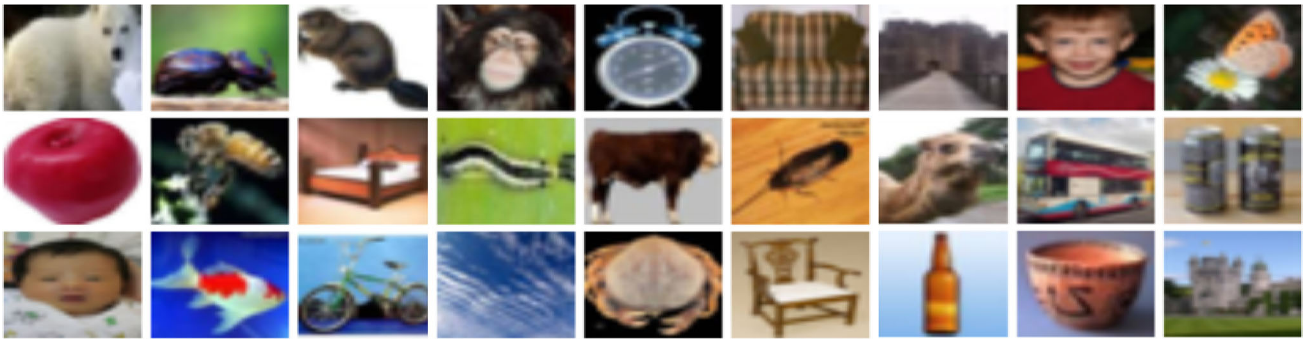


Fig. 6 Cifar 100 dataset sample images

Table 2 Average precision, recall, ARP, and ARR for the cifar 100 dataset

Cifar 100 dataset									
Image category	Avg precision	Avg recall	ARP	ARR	Image category	Avg precision	Avg recall	ARP	ARR
Apple	**1.00**	0.10	1.00	0.10	Mushroom	0.75	0.14	0.74	0.11
Aquarium Fish	**1.00**	0.10	1.00	0.10	Oak tree	0.85	0.11	0.74	0.15
Baby	0.65	0.15	0.88	0.11	Orange	**1.00**	0.10	0.75	0.15
Bear	**0.45**	0.22	0.77	0.14	Orchid	0.65	0.15	0.75	0.14
Beaver	0.50	0.20	0.72	0.15	Otter	0.45	0.25	0.74	0.15
Bed	0.50	0.20	0.68	0.16	Palm tree	0.95	0.10	0.74	0.15
Bee	0.45	0.25	0.64	0.17	Pear	0.75	0.14	0.74	0.15
Beetle	0.95	0.10	0.68	0.16	Pickup truck	**1.00**	0.10	0.75	0.15
Bicycle	**1.00**	0.10	0.71	0.15	Pine tree	0.55	0.18	0.74	0.15
Bottle	**1.00**	0.10	0.74	0.15	Plain	0.95	0.10	0.75	0.15
Bowl	**1.00**	0.10	0.76	0.14	Plate	0.95	0.10	0.75	0.15
Boy	0.50	0.20	0.74	0.15	Poppy	0.95	0.11	0.75	0.16
Bridge	0.80	0.12	0.75	0.17	Porcupine	0.75	0.14	0.75	0.15
Bus	0.85	0.11	0.75	0.18	Possum	0.70	0.20	0.75	0.14
Butterfly	0.45	0.28	0.73	0.15	Rabbit	0.50	0.25	0.74	0.15
Camel	0.45	0.22	0.71	0.16	Raccoon	0.40	0.15	0.74	0.15
Can	**1.00**	0.10	0.72	0.15	Ray	0.65	0.18	0.74	0.15
Castle	0.70	0.14	0.72	0.15	Road	0.95	0.10	0.74	0.15
Caterpillar	0.55	0.18	0.71	0.15	Rocket	0.95	0.10	0.74	0.15
Cattle	0.80	0.12	0.72	0.15	Rose	**1.00**	0.10	0.74	0.15
Chair	**1.00**	0.10	0.73	0.15	Sea	0.80	0.11	0.75	0.14
Chimpanzee	**1.00**	0.10	0.74	0.15	Seal	0.85	0.11	0.74	0.15
Clock	**1.00**	0.10	0.75	0.14	Shark	0.30	0.33	0.74	0.15
Cloud	**1.00**	0.10	0.76	0.14	Shrew	0.90	0.11	0.74	0.15
Cockroach	0.80	0.12	0.77	0.14	Skunk	0.50	0.20	0.74	0.15
Couch	0.55	0.18	0.76	0.14	Skyscraper	0.90	0.11	0.74	0.15
Crab	0.55	0.18	0.75	0.14	Snail	0.75	0.14	0.74	0.15
Crocodile	0.55	0.18	0.74	0.15	Snake	0.70	0.15	0.74	0.15
Cup	0.95	0.10	0.75	0.14	Spider	0.95	0.10	0.74	0.15
Dinosaur	0.55	0.18	0.74	0.14	Squirrel	0.75	0.14	0.74	0.23
Dolphin	0.75	0.13	0.74	0.14	Streetcar	0.90	0.11	0.74	0.15
Elephant	0.80	0.12	0.74	0.14	Sunflower	0.60	0.10	0.74	0.15
Flatfish	0.85	0.11	0.75	0.14	Sweet pepper	**1.00**	0.10	0.74	0.15
Forest	0.95	0.10	0.75	0.14	Table	0.95	0.10	0.74	0.15
Fox	**1.00**	0.10	0.76	0.14	Tank	0.75	0.18	0.74	0.15
Girl	0.45	0.25	0.75	0.14	Telephone	**1.00**	0.10	0.74	0.15
Hamster	**1.00**	0.10	0.76	0.14	Television	0.90	0.11	0.74	0.15
House	0.45	0.25	0.75	0.15	Tiger	0.85	0.13	0.74	0.15
Kangaroo	0.55	0.18	0.74	0.15	Tractor	0.60	0.15	0.74	0.15
Keyboard	**1.00**	0.10	0.75	0.14	Train	0.80	0.13	0.74	0.15
Lamp	0.60	0.16	0.74	0.15	Trout	**1.00**	0.10	0.74	0.15
Lawn mower	**1.00**	0.10	0.75	0.14	Tulip	0.95	0.10	0.73	0.15
Leopard	0.65	0.16	0.75	0.14	Turtle	0.50	0.20	0.73	0.15
Lion	0.95	0.10	0.75	0.14	Wardrobe	**1.00**	0.10	0.74	0.15
Lizard	0.75	0.13	0.75	0.14	Whale	0.90	0.12	0.74	0.15
Lobster	0.35	0.28	0.74	0.15	Willow tree	0.95	0.11	0.73	0.15

Table 2 continued

Cifar 100 dataset

Image category	Avg precision	Avg recall	ARP	ARR	Image category	Avg precision	Avg recall	ARP	ARR
Man	0.85	0.12	0.74	0.14	Wolf	**1.00**	0.10	0.74	0.15
Maple tree	0.75	0.13	0.74	0.14	Woman	0.95	0.11	0.74	0.15
Motorcycle	**1.00**	0.10	0.75	0.14	Worm	**1.00**	0.10	0.74	0.15
Mountain	0.95	0.10	0.75	0.14	Mouse	0.95	0.11	0.74	0.14

diverse object shape, spatial, and texture information. The images with identical background and foreground objects belong to the same semantic group, which are efficiently classified by the proposed method. The outstanding results for classifying images with different textures have been achieved by the presented method using auto-correlation and norm steps. Moreover, the proposed method uses auto-correlation, image scaling and gradient computation, and box filtering with CNN features for remarkable image classification. By applying these techniques, high AP rates are achieved on images with different textures. In the 17 Flower dataset, some categories have different object patterns, whereas most of the categories are based on texture images that have similar colors and patterns. Furthermore, an 80% average precision rate has been achieved by the proposed method in most of the categories of the 17 Flower dataset. The images of the 17 flowers dataset are shown in Fig. 8.

The significant average precision ratios are achieved by the proposed method, for images of identical colors and patterns shown in Fig. 9a. The texture images are essentially classified into vertical lines of the same color and directions that produced remarkable results for different types of image categories. With the usage of gradient computation with deep features, the texture images from different image categories are categorized efficiently.

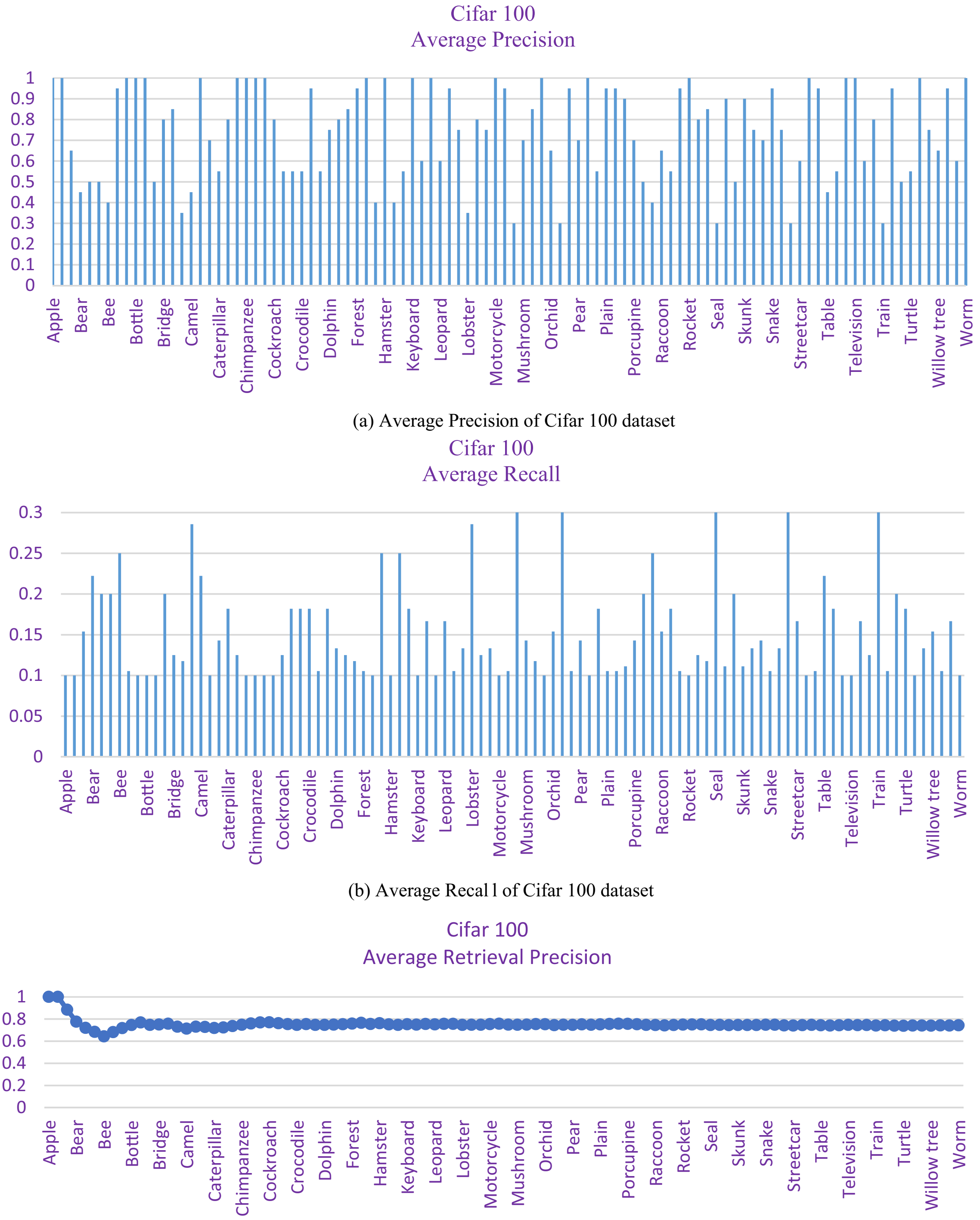
Moreover, the proposed method also implements Gaussian smoothing and localization for achieving a remarkable average precision ratio in texture images. 100% AP ratio has been achieved by the proposed method in the categories of daisy, dandelion, sunflower, and windflower. Whereas, 85% of AP achieve in categories of buttercup and coltsfoot. Furthermore, 80% of AP achieve in the categories of cowslip and pansy. 70% AP rate has been achieved in the categories of crocus, daffodil, Lilly valley, and tulip. Only snowdrop showed the 60% AP ratio. The proposed method shows an outstanding 81%mAP. Figure 9b shows the remarkable AR ratio for all categories of the 17 Flower dataset. To classify the different color images the proposed method uses the RGB coefficient.

Table 3 shows average precision, recall, APR and ARR for the 17-flower dataset. Moreover, outstanding performance has been achieved by the proposed method in most image categories which contain images that have different colors and shapes. CCN features with image gradient computation, Gaussian smoothing, shape base filtering, color coefficient, and spatial mapping give strength to the proposed method to classify the images efficiently and effectively. Moreover, the proposed method shows outstanding results in all the categories of the 17 Flower dataset and overall achieved 81% mAP (Table 4).

Figure 10a and b shows the ARP and ARR for all the categories of the dataset. The proposed method showed the remarkable ARP and ARR for all the categories of the 17 Flower dataset.

FTVL database [59] has 2612 images of different fruits and vegetables and it contains 15 challenging categories named Agata Potato, Asterix Potato, Cashew, Diamond Peach, Fuji Apple, Granny Smith Apple, Honeydew Melon, Kiwi, Nectarine, Onion, Orange, Plum, Spanish Pear, Watermelon, and Tahiti Lime. More than 26000 images of various foreground and backdrop textures are available in the FVTL dataset. The proposed method has an exceptional object detection capability, and it showed outstanding results on cluttered and complex objects. Using the FTVL dataset, the consistency and effectiveness of the proposed approach are tested. The FVTL dataset contains images of different types of shapes, colors, and texture objects which made it more suitable for texture analysis. The proposed method showed improved image classification and remarkable AP and AR rates for overlapping, cluttered, and complex images. Sample images of the FTVL database are shown in Fig. 11.

The average precision and average recall rates are shown in Fig. 12a and b. All categories of the FTVL dataset have been utilized to test the proposed method. The outstanding AP results are shown in Fig. 12a. Five categories of the FTVL dataset showed the 100% AP whereas, six categories achieved 95% AP. Only Asterix potato achieves 85% AP due to its colors and complex background images. The proposed method also produced superior results for complicated and overlay images, which were shown in Fig. 12b. Significant AR rates have been achieved by the proposed method in all the categories of the FVTL database. In addition, mean average precision is used to check the efficiency of the proposed method. The proposed method achieved a 95% mAP.



(a) Average Precision of Cifar 100 dataset

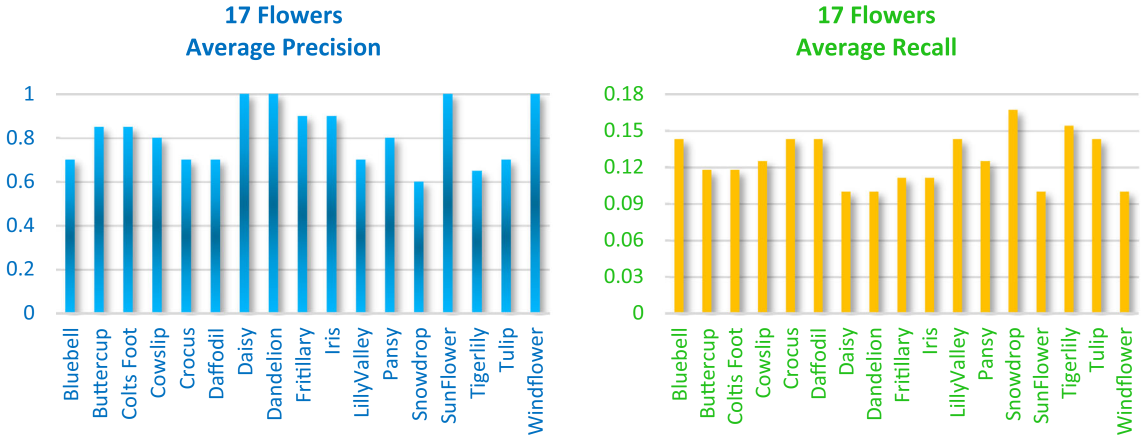
(b) Average Recal l of Cifar 100 dataset

(c) Average Retrieval Precision of Cifar 100 dataset

Fig. 7 **a** Average precision of Cifar 100 dataset. **b** Average recall of Cifar 100 dataset. **c** Average retrieval precision of Cifar 100 dataset

The ARP for FTVL dataset is shown in Fig. 13a. The significant results have been achieved by the proposed method using L2 color coefficients which effectively index and classify the images of the FTVL dataset. Above 90% ARP rates have been attained in most of the categories of the FTVL dataset. The proposed method provides outstanding results in most categories such as agata potato, cashew, diamond peach, fuji apple, granny smooth apple, honeydew, melon, kiwi, nectarine, onion, orange, plum, spanish pear, taiti lime, and watermelon. The encouraging ARR results have been achieved by the proposed method which is shown in Fig. 13b.

Fig. 8. 17 Flower dataset images



(a) Average Precision 17 flower dataset

(b) Average Recall 17 flower dataset

Fig. 9 **a** Average precision 17 flower dataset. **b** Average recall 17 flower dataset

These results show the superiority of proposed for the FTVL dataset.

Experimentation on the complex and cluttered dataset

The Oxford Buildings Dataset [60] consists of 5062 images with a resolution of 1024 × 768 which were collected from Flickr by searching for particular Oxford landmarks. For this purpose, 17 different queries named as All Souls, Balliol, Christ Church, Hertford, Jesus, Keble, Magdalen, New, Oriel, Trinity, Radcliffe Camera, Cornmarket, Bodleian, Pitt Rivers, Ashmolean, Worcester, Oxford has been used for flicker to download the relevant images in each category. This collection has been manually annotated to generate a comprehensive ground truth for eleven different landmarks of oxford, each landmark represented by 5 possible queries. The proposed method efficiently classifies the complex and clattered images. The proposed method uses auto-correlation and norm steps to gain outstanding results from the complex and cluttered dataset. The presented method presented promising results on the oxford building dataset and achieve up to 80% precision in most of the categories. Sample images from the oxford building dataset are shown in Fig. 14.

The average precision and average recall rates for the oxford building dataset are shown in Figs. 15a and b. The effectiveness of the proposed method was tested on all the categories of the oxford building dataset. The remarkable results of the AP ratio are shown in Fig. 15a. Christ church, oxford, and red cliff show 95% AP. Moreover, the proposed method performs well on the all-categories oxford dataset.

The efficient image classification results are achieved by applying gradient computing, proper scale integration, box filtering, RGB coefficient, and Gaussian second-order derivatives with CNN. In some categories such as Balliol, Oriel, and , proposed method achieves a 70% AP, whereas, in Heart fort, the proposed method shows only 65% AP due to complex and cluttered images.

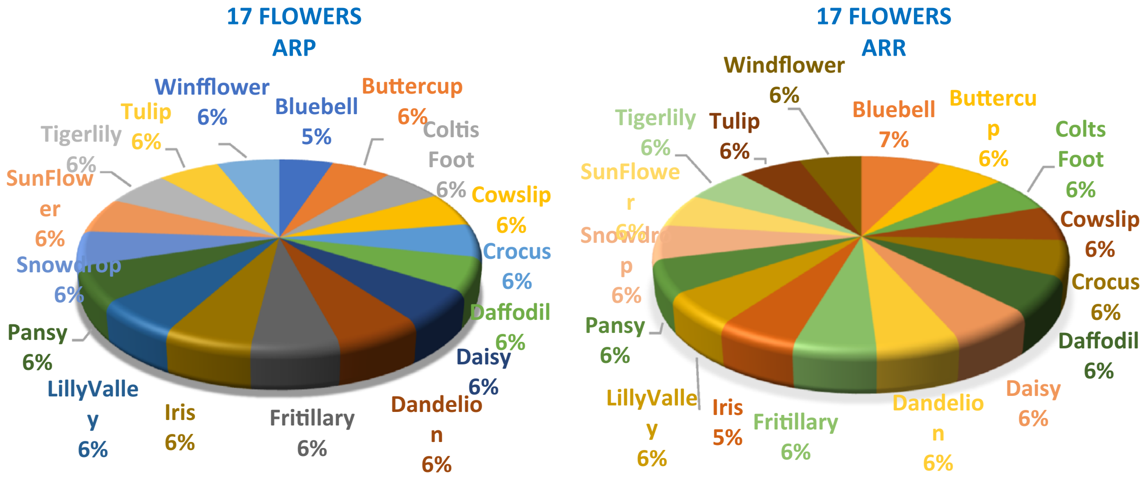
Moreover, mean average precision was also utilized to measure the performance proposed method, which shows more than 80% mAP. The proposed method shows a brilliant ARP result for the oxford building dataset, as shown in Fig. 16a. Whereas the proposed method also has outstanding ARR for the oxford building dataset shown in Fig. 16b.

Experimentation on overlay dataset

The most acceptable and widely used benchmark dataset for image retrieval and classification is Corel 1000 [57]. This dataset includes various image categories which contain images of complex objects with simple backgrounds. There are 1000 images in the Corel dataset which are divided into 10 categories and each category consists of 100 images. The images from various semantic groups are presented in Corel

Table 3 Avg. precision, avg recall, APR, and ARR of 17 flower dataset

17 Flower datasets				
Categories	Avg. precision	Avg. recall	APR	ARR
Bluebell	0.7	0.16	0.70	0.16
Buttercup	0.85	0.11	0.77	0.14
Colts Foot	0.85	0.11	0.80	0.13
Cowslip	0.8	0.12	0.80	0.13
Crocus	0.7	0.14	0.78	0.13
Daffodil	0.7	0.14	0.76	0.13
Daisy	1.0	0.10	0.80	0.13
Dandelion	1.0	0.10	0.82	0.12
Fritillary	0.9	0.11	0.83	0.12
Iris	0.9	0.11	0.84	0.12
Lilly Valley	0.7	0.14	0.82	0.12
Pansy	0.8	0.12	0.82	0.12
Snowdrop	0.6	0.16	0.80	0.13
Sun Flower	1.0	0.10	0.82	0.12
Tigerlily	0.65	0.16	0.81	0.13
Tulip	0.7	0.18	0.80	0.13



(a) ARP for 17 flower dataset

(b) Average Recall 17 flower dataset

Fig. 10 **a** ARP for 17 flower dataset. **b** Average recall 17 flower dataset

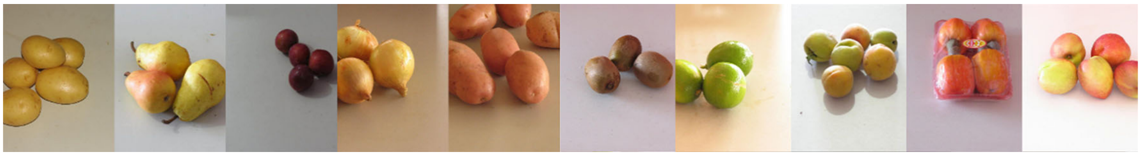
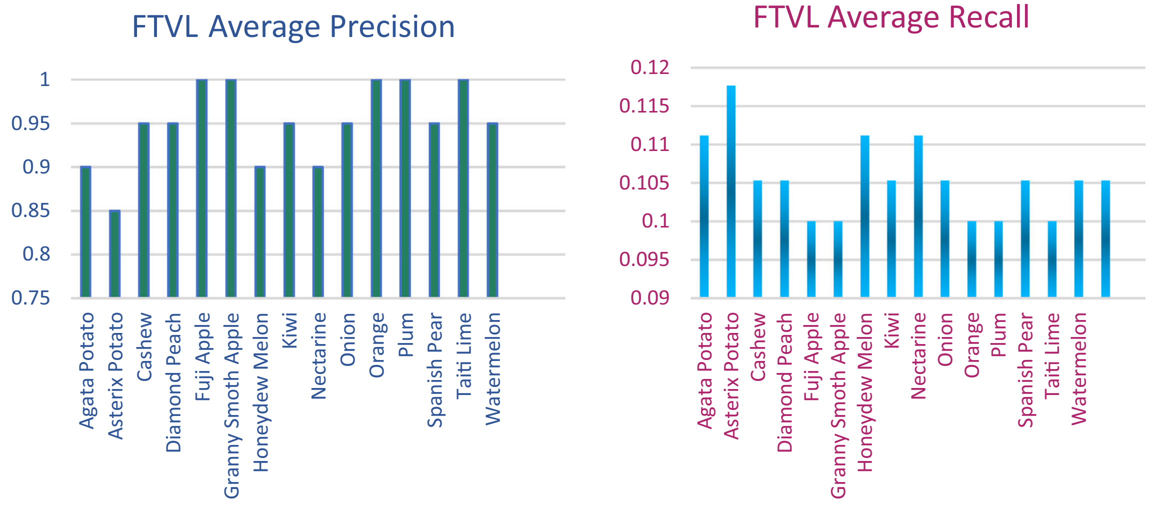


Fig. 11 FTVL dataset images

1000 dataset, such as buses, animals, flowers, beaches, people, and natural scenes. The Coral 1000 datasets contain a variety of image semantics that are useful for image detection, as shown in Fig. 17. The resolution of images of the Corel 1000 dataset is 256 × 384.

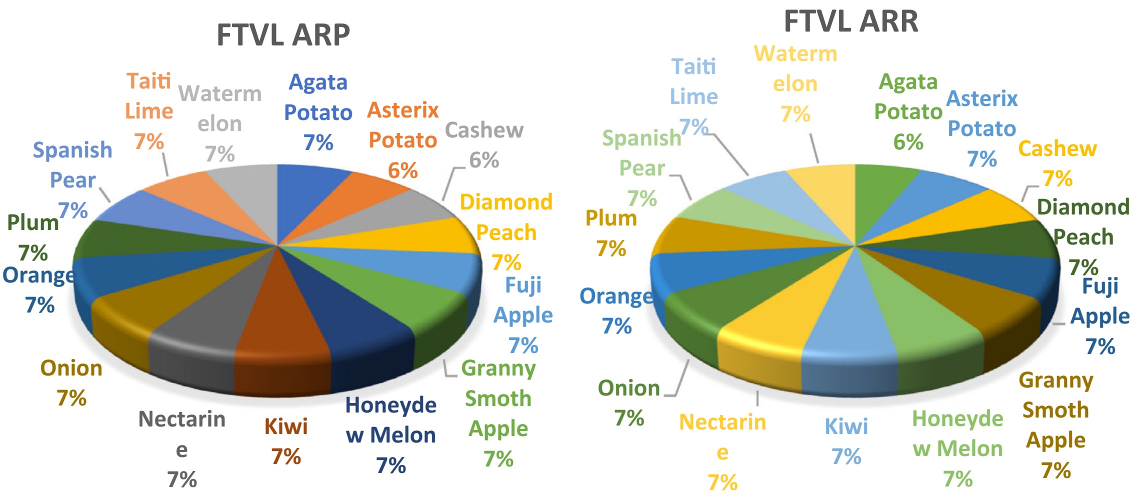
The proposed method efficiently classified the images which belong to the different semantic groups. Furthermore, these images have different backgrounds and foregrounds. The AP results for Corel 1000 are shown in Fig. 18a. The outstanding performance is achieved by the proposed method



(a) Average Precision FTVL dataset

(b) Average Recall FTVL dataset

Fig. 12 **a** Average precision FTVL dataset. **b** Average recall FTVL dataset



(a) ARP for FTVL dataset

(b) ARR for FTVL dataset

Fig. 13 **a** ARP for FTVL dataset. **b** ARR for FTVL dataset

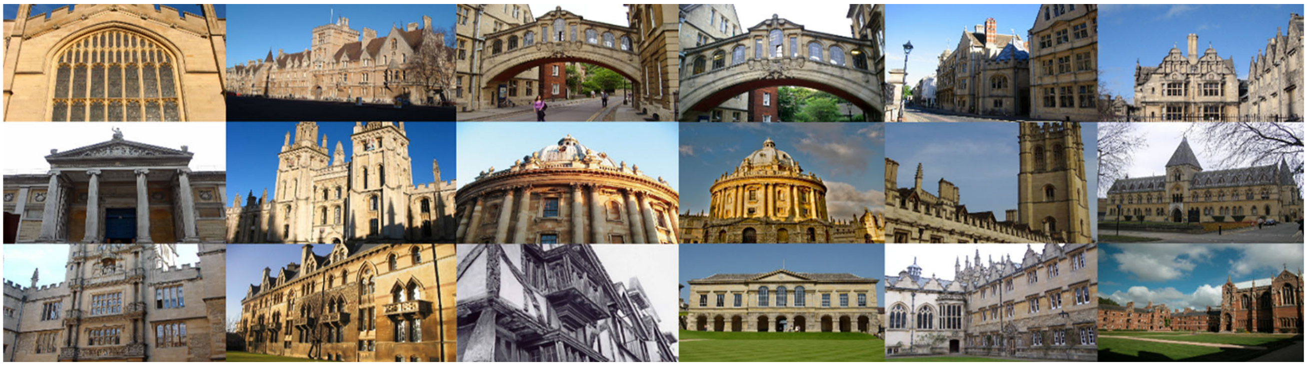
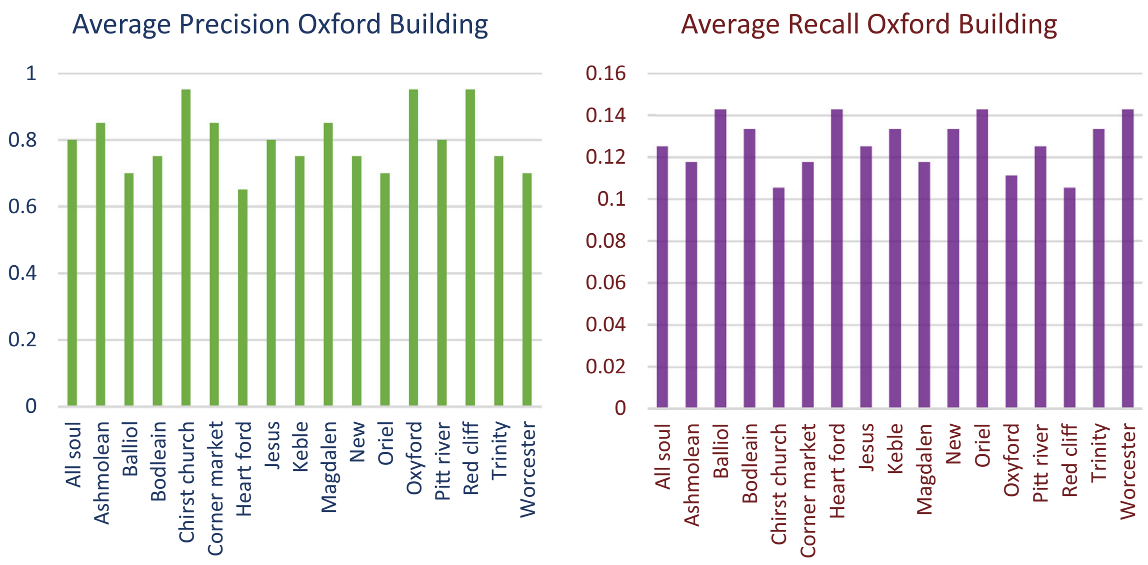


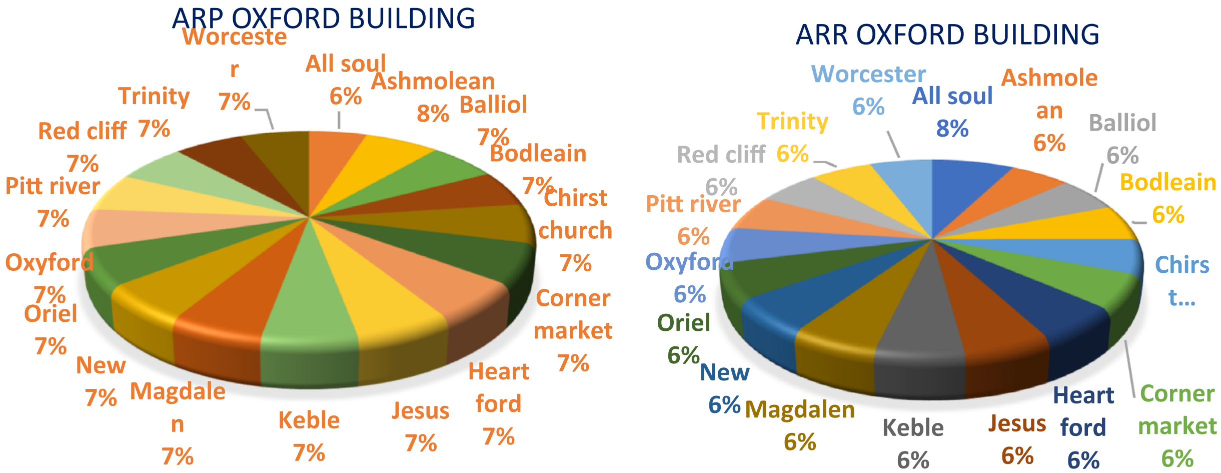
Fig. 14 Sample images of the oxford building dataset



(a) Average precision

(b) Average Recall

Fig. 15 **a** Average precision. **b** Average recall



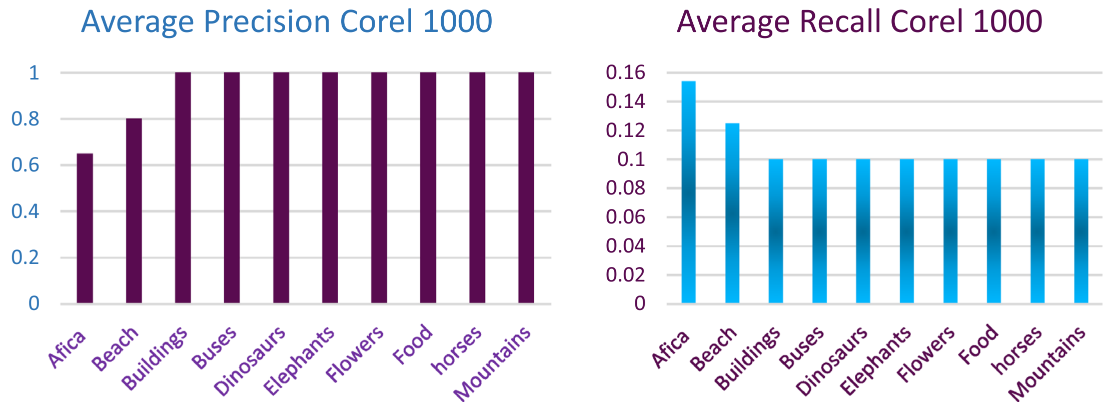
(a) ARP for oxford building dataset

(b) ARR for oxford building dataset

Fig. 16 **a** ARP for oxford building dataset. **b** ARR for oxford building dataset

Fig. 17 Sample images of Corel 1000 image dataset

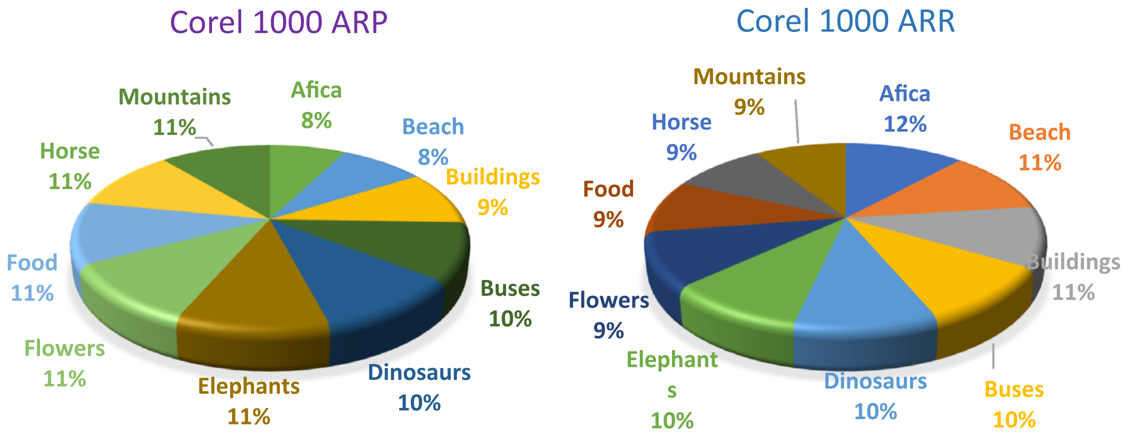
using deep learning. Autocorrelation, image gradient, Gaussian smoothing, box filtering, and RGB coefficients with deep learning features provide strength to the proposed method for effective image classification. Superior results of AR are achieved by the proposed method in almost all the categories, such as Africa, dinosaurs, buses, mountains, horses, and elephants. The categories of building, buses, dinosaur, elephant flower, food, mountain, and horse achieved a 100% AP rates.



(a) Average Precision Corel 1000 image dataset.

(b) Average Recall Corel 1000 image dataset.

Fig. 18 **a** Average precision Corel 1000 image dataset. **b** Average recall Corel 1000 image dataset



(a) A RP Corel 1000 image dataset.

(b) ARR Corel 1000 image datatset.

Fig. 19 **a** ARP Corel 1000 image dataset. **b** ARR Corel 1000 image dataset

Whereas, the category of beach achieves an 80% AP rate and the category of Africa achieve a 65% AP rate. Moreover, the proposed method achieves more than 94% mAP. Furthermore, outstanding AR rates have been shown by the proposed method in most of the categories. The categories of buildings, buses, horses, dinosaur, food, mountains, elephant, and flower achieve a 0.10 AR rate, as shown in Fig. 18b.

Moreover, significant ARP rates have been achieved by the proposed method which is shown in Fig. 19a. The categories of elephant, flower, food, horse, and mountains achieve 11% ARP ratios. Whereas, the categories of dinosaurs and buses showed 10% ARP ratios. Moreover, the category of building achieves a 9% ARP ratio, whereas Africa and beach achieve an 8% ARP ratio. Outstanding ARR is shown in Fig. 19b. The category of Africa shows a 12% ARR ratio. Whereas, beaches and building achieve 11% ARR. Buses, dinosaurs, and elephants report a 10% ARR ratio. Flowers, food, horses, and mountains report a 9% ARR ratio which shows the significance of the proposed method.

Accuracy comparison of the Corel-1000 dataset with existing state-of-the-art methods

The accuracy and effectiveness of the proposed method have been compared with the existing state-of-the-art methods using the results of the Corel 1000 dataset. For this purpose, Ahmed et al. [42], Shah et al. [43], Artiza et al. [62], Alsmadi et al. [63], Mehmood et al. [64], Zeng et al. [66], and Dubey et al. [67] have been used. Moreover, remarkable precision was achieved by the proposed method in most of the categories of Corel 1000 when compared with other methods, as shown in Fig. 20. The categories of building, buses, dinosaur, elephant flower, food, mountain, and horse achieved the highest AP rate. Whereas, the category of beach achieves an 80% AP rate and the category of Africa achieves a 65% AP rate.

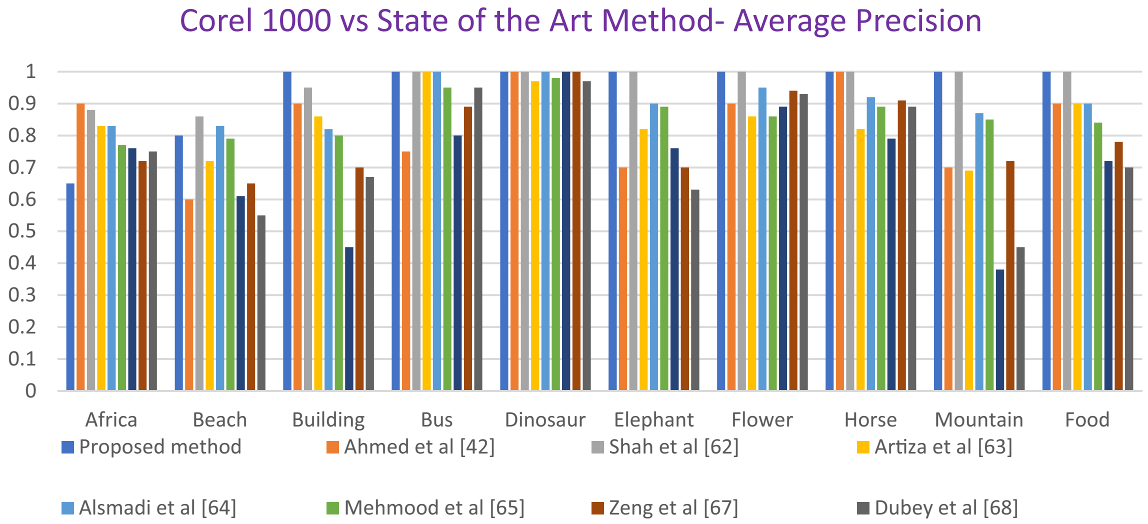


Fig. 20 Comparison of proposed method with state-of-the-art methods on Corel 1000

Table 4 Coral 1000 vs state-of-the-art methods—average precision

Coral 1000 vs state-of-the-art methods—average precision

Categories	Proposed method	Ahmed et al. [42]	Shah et al. [43]	Artiza et al. [62]	Alsmadi et al. [63]	Mehmood et al. [64]	Zeng et al. [66]	Dubey et al. [67]
Bus	**1.00**	0.75	0.95	**1.00**	**1.00**	0.95	0.89	0.95
Dinosaur	**1.00**	**1.00**	0.95	0.97	**1.00**	0.98	**1.00**	0.97
Horse	**1.00**	**1.00**	**1.00**	0.82	0.92	0.89	0.91	0.89
Building	**0.95**	0.90	0.95	0.86	0.82	0.80	0.70	0.67
Elephant	**0.95**	0.70	0.90	0.82	0.90	0.89	0.70	0.63
Flower	**0.95**	0.90	0.95	0.86	0.95	0.86	0.94	0.93
Food	**0.95**	0.90	0.90	0.90	0.90	0.84	0.78	0.70
Mountain	**0.90**	0.70	0.89	0.69	0.87	0.85	0.72	0.45
Beach	0.80	0.60	0.86	0.72	0.83	0.79	0.65	0.55
Africa	0.65	0.90	0.88	0.83	0.83	0.77	0.72	0.75

Figure 20 gives an overall representation of the proposed method's average precision relative to the current state-of-the-art techniques. In most categories of the Corel 1000 dataset, the proposed method demonstrated superior performance relative to other methods. The categories of building, bus, dinosaurs, flowers, horses, mountains, and food showed the highest average precision using the proposed method. Whereas, in the categories of Africa and beaches, state-of-the-art methods showed better performance. Ahmed et al. [42] presented a novel idea that uses two channels for an image. The gray-scale channel detects the edges and corners of images that are fused with the features extracted through the RGB channel. The better AP for the categories of dinosaurs and horses was achieved by this method. Shah et al. [43] use CNN for image feature extraction in the CBIR system. The retrieved features were utilized to determine the Euclidean distance between the query and the stored images. This method showed a remarkable performance for the categories of Bus, dinosaur, elephant, flower, horse, mountain, and food.

Artiza et al. [62] present a genetic classifier comity learning (GCCL) technique based on a genetic algorithm (GA) for generating stable classifiers. This technique merges ANN with SVMs through asymmetric and symmetric bagging. Moreover, it provides a solution to the disagreement over-classification that existed between several classifiers. Additionally, it finds a solution to the class imbalance issue that existed inside CBIR. This method showed a better AP for the category of buses. Alsmadi et al. [63] implement a CBIR technique that uses several algorithms to implement the

CBIR technique, it uses a neutrosophic clustering algorithm for the construction of RGB images, whereas shape features are extracted using a canny edge detector. The canny edge histogram and YCbCr color were used to extract the color features. For the extraction of texture features, gray-level matrix is used. This method achieves better AP for the categories of buses and dinosaur. Mehmood et al. [64] developed a method that implemented a weighted average of triangular histograms (WATH) for visual words. Moreover, an inverted index for the BoVW model was used that incorporates image spatial contents which avoids overfitting and semantic gap concerns between high-level and low-level image semantics. Dubey et al. [67] suggest a new way to describe images using decoded LBPs from more than one channel. We provide two schemas based on adders and decoders for combining LBPs from several channels. Both these methods showed better results for the category of the dinosaur. Furthermore, Zeng et al. [66] presented a unique image representation approach that describes an image as a spatiogram, or extended histogram, of colors quantized using Gaussian Mixture Models (GMMs). This method showed remarkable AP for the category of dinosaurs.

Conclusion

Images are growing rapidly due to which image retrieval systems are required which can work with large-scale databases. Deep learning for image retrieval is used for retrieving images efficiently, but still retrieving images with the highest accuracy is the biggest challenge. This paper introduces a novel method that combines with the Vgg-16 architecture signature, and identifies prominent objects, colors, and features which accurately represent the contents of an image. The signatures of Vgg-16 architecture strengthen the information carriers to extract strong image features which represent the content of an image. The image features are identified using the signature of image smoothing, gradient computation, scaling, and feature reduction. The proposed method shows impressive results on benchmark datasets which include 10 categories of Corel 1000 dataset and 17 categories of oxford building dataset. Outstanding results are achieved on a challenging dataset of Cifar 10 and Cifar 100. Moreover, high precision for the texture features is achieved on 17 flower and FTVL datasets by the presented method.

Funding This work was supported by the GRRC program of Gyeonggi province. [GRRCGachon2020(B04), Development of AI-based Healthcare Devices] and by the National Research Foundation of Korea (NRF) Grant funded by the Korean government under Grant NRF-2021R1I1A1A01045177.

References

1. Bay H, Tinne T, Luc VG (2006) Surf: speeded up robust features. In: European conference on computer vision. Springer, pp 404–417
2. Lowe DG (2004) Distinctive image features from scale-invariant key points. Int J Comput Vis 60(2):91–110
3. Juneja K, Akhilesh V, Savita G, Swati G (2015) A survey on recent image indexing and retrieval techniques for low-level feature extraction in CBIR systems. In: IEEE international conference on computational intelligence and communication technology, pp 67–72
4. Saritha RR, Varghese P, Ganesh Kumar P (2019) Content-based image retrieval using deep learning process. Cluster Comput 22(2):4187–4200
5. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26
6. Guo Y, Liu Yu, Oerlemans A, Lao S, Song Wu, Lew MS (2016) Deep learning for visual understanding: a review. Neurocomputing 187:27–48
7. Szegedy C, Wei L, Yangqing J, Pierre S, Scott R, Dragomir A, Dumitru E, Vincent V, Andrew R (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1–9
8. Mohamed O, Ouanan M, Aksasse B (2017) Content-based image retrieval using convolutional neural networks. In: First international conference on real time intelligent systems. Springer, pp 463–476
9. Donahue J, Yangqing J, Oriol V, Judy H, Ning Z, Eric T, Trevor D (2014) Decaf: a deep convolutional activation feature for generic visual recognition. In: International conference on machine learning. pp 647–655
10. Alsmadi MK (2018) Query-sensitive similarity measure for content-based image retrieval using meta-heuristic algorithm. J King Saud Univ Comput Inf Sci 30(3):373–381
11. Alsmadi MK (2017) An efficient similarity measure for content-based image retrieval using memetic algorithm. Egypt J Basic Appl Sci 4(2):112–122
12. Vassou SA, Nektarios A, Angelos A, Klitos C, Savvas AC (2017) CoMo: a compact composite moment-based descriptor for image retrieval. In: Proceedings of the 15th international workshop on content-based multimedia indexing. pp 1–5
13. Wan J, Dayong W, Steven CHH, Pengcheng W, Jianke Z, Yongdong Z, Jintao L (2014) Deep learning for content-based image retrieval: a comprehensive study. In: Proceedings of the 22nd ACM international conference on Multimedia. pp 157–166
14. Girshick R, Jeff D, Trevor D, Jitendra M (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 580–587
15. Amodei D, Sundaram A, Rishita A, Jingliang B, Eric B, Carl C, Jared C et al (2016) Deep speech 2: end-to-end speech recognition

in english and mandarin. In: International conference on machine learning. pp 173–182
16. Toshev A, Christian S (2014) Deeppose: human pose estimation via deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 1653–1660
17. Young T, Devamanyu H, Soujanya P, Erik C (2018) Recent trends in deep learning based natural language processing. IEEE Comput Intell Mag 13(3):55–75
18. Babenko A, Anton S, Alexandr C, Victor L (2014) Neural codes for image retrieval. In: European conference on computer vision. Springer, Cham, pp 584–599
19. Babenko A, Victor L (2015) Aggregating local deep features for image retrieval. In: Proceedings of the IEEE international conference on computer vision. pp 1269–1277
20. Krizhevsky A, Geoffrey H (2009) Learning multiple layers of features from tiny images
21. Torralba A, Fergus R, Freeman WT (2008) 80 million tiny images: a large data set for nonparametric object and scene recognition. IEEE Trans Pattern Anal Mach Intell 30(11):1958–1970
22. Liu H, Zhao Q, Mbelwa JT, Tang S, Zhang J (2019) Weighted two-step aggregated VLAD for image retrieval. Vis Comput 35(12):1783–1795
23. Liu Z, Shengjin W, Qi T (2016) Fine-residual VLA for image retrieval. Neurocomputing 173:1183–1191
24. Gao Y, Oscar B, Ning Z, Trevor D (2016) Compact bilinear pooling. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp 317–326
25. Kalantidis Y, Clayton M, Simon O (2016) Cross-dimensional weighting for aggregated deep convolutional features. In: European conference on computer vision. Springer, Cham, pp 685–701
26. Hasan MS (2017) An application of pre-trained CNN for image classification. In: 2017 20th international conference of computer and information technology (ICCIT). IEEE, pp 1–6
27. Gidaris S, Nikos K (2015) Object detection via a multi-region and semantic segmentation-aware cnn model. In: Proceedings of the IEEE international conference on computer vision. pp 1134–1142
28. Chadha A, Andreopoulos Y (2017) Voronoi-based compact image descriptors: efficient region-of-interest retrieval with VLAD and deep-learning-based descriptors. IEEE Trans Multimed 19(7):1596–1608
29. Yu W, Yang K, Yao H, Sun X, Pengfei Xu (2017) Exploiting the complementary strengths of multi-layer CNN features for image retrieval. Neurocomputing 237:235–241
30. Paulin M, Julien M, Matthijs D, Zaid H, Florent P, Cordelia S (2017) Convolutional patch representations for image retrieval: an unsupervised approach. Int J Comput Vis 121(1):149–168
31. Gordo A, Almazan J, Revaud J, Larlus D (2017) End-to-end learning of deep visual representations for image retrieval. Int J Comput Vis 124(2):237–254
32. Mohedano E, Kevin M, Noel EO, Amaia S, Ferran M, Xavier G (2016) Bags of local convolutional features for scalable instance search. In: Proceedings of the 2016 ACM on international conference on multimedia retrieval. pp 327–331
33. Alzu'bi A, Amira A, Ramzan N (2017) Content-based image retrieval with compact deep convolutional features. Neurocomputing 249:95–105
34. Li Z, Tang J (2015) Weakly supervised deep metric learning for community-contributed image retrieval. IEEE Trans Multimed 17(11):1989–1999
35. Radenović F, Tolias G, Chum O (2018) Fine-tuning CNN image retrieval with no human annotation. IEEE Trans Pattern Anal Mach Intell 41(7):1655–1668
36. Deng J, Wei D, Richard S, Li-Jia L, Kai L, Li F-F (2009) Imagenet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp 248–255
37. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25:1097–1105
38. Liu X, Zhang S, Huang T, Tian Qi (2020) E2BoWs: An end-to-end Bag-of-Words model via deep convolutional neural network for image retrieval. Neurocomputing 395:188–198
39. Barz B, Joachim D (2019) Hierarchy-based image embeddings for semantic image retrieval. In: 2019 IEEE winter conference on applications of computer vision (WACV). pp 638–647
40. Yan L, Hanlin Lu, Wang C, Ye Z, Chen H, Ling H (2019) Deep linear discriminant analysis hashing for image retrieval. Multimed Tools Appl 78(11):15101–15119
41. Zhu H (2020) Massive-scale image retrieval based on deepvisual feature representation. J Vis Commun Image Represent 70:102738
42. Ahmed KT, Shahida U, Amjad I (2019) Content based image retrieval using image features information fusion. Inf Fusion 51:76–99
43. Shah A, Rashid N, Shahid I, Muhammad AS (2017) Improving cbir accuracy using convolutional neural network for feature extraction. In: 2017 13th international conference on emerging technologies (ICET). IEEE, pp 1–5
44. Shaha M, Meenakshi P (2018) Transfer learning for image classification. In: 2018 second international conference on electronics, communication and aerospace technology (ICECA). IEEE, pp 656–660
45. Kumar MD, Morteza B, Shujin Z, Shivam K, Hamid RT (2017) A comparative study of CNN, BoVW and LBP for classification of histopathological images. In: 2017 IEEE symposium series on computational intelligence (SSCI). pp 1–7
46. Harris CG, Mike S (1988) A combined corner and edge detector. Alvey Vis Conf 15(50):10–5244
47. Bay H, Andreas E, Tinne T, Luc VG (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110(3):346–359
48. Robertson C, George SC (2012) Theory and practical recommendations for auto-correlation-based image correlation spectroscopy. J Biomed Opt 17(8):080801
49. https://www.sciencedirect.com/topics/computer-science/gradient-computation
50. Jacobs D (2005) Image gradients. Class Notes for CMSC 426
51. Lindeberg T (2018) Spatio-temporal scale selection in video data. J Math Imaging Vis 60(4):525–562
52. Alazzawi A (2015) Edge detection-application of (first and second) order derivative in image processing: communication. Diyala J Eng Sci 8(4):430–440
53. Neubeck A, Luc VG (2006) Efficient non-maximum suppression. In: 18th international conference on pattern recognition (ICPR'06), vol 3. IEEE, pp 850–855
54. Brown M, David GL (2002) Invariant features from interest point groups. In: BMVC, vol 4
55. Simonyan K, Andrew Z (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
56. Zhu X, Michael B (2017) B-CNN: branch convolutional neural network for hierarchical classification. arXiv preprint arXiv:1709.09890
57. Li B, Yijuan L, Chunyuan L, Afzal G, Tobias S, Masaki A, Martin B et al (2015) A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Comput Vis Image Underst 131:1–27
58. Nilsback M-E, Andrew Z (2006) A visual vocabulary for flower classification. In: 2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06), vol 2. IEEE, pp 1447–1454
59. Patino-Saucedo A, Horacio R-G, Jorg C (2018) Tropical fruits classification using an alexnet-type convolutional neural network and image augmentation. In: International conference on neural information processing. Springer, Cham, pp 371–379

60. Philbin J, Ondrej C, Michael I, Josef S, Andrew Z (2007) Object retrieval with large vocabularies and fast spatial matching. In: 2007 IEEE conference on computer vision and pattern recognition. pp 1–8
61. Steger C (2002) Occlusion, clutter, and illumination invariant object recognition. Int Arch Photogramm Remote Sens Spatial Inf Sci 34(3/A):345–350
62. Irtaza A, Syed MA, Khawaja TA, Arfan J, Ahmad K, Ali J, Muhammad TM (2018) An ensemble based evolutionary approach to the class imbalance problem with applications in CBIR. Appl Sci 8(4):495
63. Alsmadi MK (2020) Content-based image retrieval using color, shape and texture descriptors and features. Arab J Sci Eng 45(4):3317–3330
64. Mehmood Z, Toqeer M, Muhammad AJ (2018) Content-based image retrieval and semantic automatic image annotation based on the weighted average of triangular histograms using support vector machine. Appl Intell 48(1):166–181
65. Kanwal K, Khawaja TA, Rashid K, Aliya TA, Jing L (2020) Deep learning using symmetry, FAST scores, shape-based filtering and spatial mapping integrated with CNN for large scale image retrieval. Symmetry 12(4):612
66. Zeng S, Huang R, Wang H, Kang Z (2016) Image retrieval using spatiograms of colors quantized by Gaussian mixture models. Neurocomputing 171:673–684
67. Dubey SR, Satish KS, Rajat KS (2016) Multichannel decoded local binary patterns for content-based image retrieval. IEEE Trans Image Process 25(9):4018–4032

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.