sensors

MDPI

Article

Deep Learning Using Isotroping, Laplacing, Eigenvalues Interpolative Binding, and Convolved Determinants with Normed Mapping for Large-Scale Image Retrieval

Khadija Kanwal [1], Khawaja Tehseen Ahmad [2], Rashid Khan [3], Naji Alhusaini [1] and Li Jing [1,*]

[1] School of Computer Science and Technology, University of Science and Technology of China, Hefei 230009, China; khadijakanwal@mail.ustc.edu.cn (K.K.); husaini@ustc.edu.cn (N.A.)

[2] Department of Computer Science, Bahauddin Zakariya University, Multan 60800, Pakistan; tehseen@bzu.edu.pk

[3] Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230009, China; rashidkhan@mail.ustc.edu.cn

* Correspondence: lj@ustc.edu.cn

Abstract: Convolutional neural networks (CNN) are relational with grid-structures and spatial dependencies for two-dimensional images to exploit location adjacencies, color values, and hidden patterns. Convolutional neural networks use sparse connections at high-level sensitivity with layered connection complying indiscriminative disciplines with local spatial mapping footprints. This fact varies with architectural dependencies, insight inputs, number and types of layers and its fusion with derived signatures. This research focuses this gap by incorporating GoogLeNet, VGG-19, and ResNet-50 architectures with maximum response based Eigenvalues textured and convolutional Laplacian scaled object features with mapped colored channels to obtain the highest image retrieval rates over millions of images from versatile semantic groups and benchmarks. Time and computation efficient formulation of the presented model is a step forward in deep learning fusion and smart signature capsulation for innovative descriptor creation. Remarkable results on challenging benchmarks are presented with a thorough contextualization to provide insight CNN effects with anchor bindings. The presented method is tested on well-known datasets including ALOT (250), Corel-1000, Cifar-10, Corel-10000, Cifar-100, Oxford Buildings, FTVL Tropical Fruits, 17-Flowers, Fashion (15), Caltech-256, and reported outstanding performance. The presented work is compared with state-of-the-art methods and experimented over tiny, large, complex, overlay, texture, color, object, shape, mimicked, plain and occupied background, multiple objected foreground images, and marked significant accuracies.

Keywords: deep learning; CNN; image retrieval; image content analysis; BoW; color image retrieval

check for updates

Citation: Kanwal, K.; Tehseen Ahmad, K.; Khan, R.; Alhusaini, N.; Jing, L. Deep Learning Using Isotroping, Laplacing, Eigenvalues Interpolative Binding, and Convolved Determinants with Normed Mapping for Large-Scale Image Retrieval. *Sensors* **2021**, *21*, 1139. https://doi.org/10.3390/s21041139

Academic Editor: Marcin Woźniak

Received: 11 January 2021
Accepted: 1 February 2021
Published: 6 February 2021

1. Introduction

Nowadays artificial intelligence (AI) and machine learning (ML) based convolutional neural network (CNN) research has fascinated significant consideration. Due to its exceptional performance, CNN is applied in various fields—including image detection, image retrieval, and image classification [1]. AlexNet [2] was a good breakthrough and then VGGNet [3] was developed for large scale classification. GoogLeNet [4] applies combined inception modules including concatenation operations, convolutions with various scales, and pooling. Recently, ResNet [5] was introduced with more than 100 convolution layers. The number of trained weights are increased with number of increasing layers. This requires a huge amount for computation and significant computational time in training and the classification steps. The classification for these networks in the real time applications is also a challenging task. To overcome this problem, graphics processing units (GPUs) are normally applied to speed up the classification and training time [6]; however, a

Tensor Processing Unit (TPU) is an equally good choice in non-parallel computing environment [7]. GPUs uses 1000 s of Arithmetic and Logic Units (ALUs) to compute heavy parallel and matrix processing neural networks. TPU do not contain general purpose hardware due to its domain specific architecture and normally adopted for massive multiplications and additions.

In practical applications, the classification acceleration is a main factor which is obtained by utilizing the power and storage resources [8]. The CNN architectures are employed for highest level of abstractions using deep learning architectures that are combined for many non-linear transformations [9]. State-of-the-art performance is achieved using CNN for various applications, including object recognition [10] and speech recognition [11]. CNN architectures have been presented to improve the workflow of image retrieval [12,13]. Furthermore, the features extraction by using deep learning methods have taken developments in the processing of multimedia contents. The CNN features [14] have attained high effectiveness in different retrieval and computer vision tasks. The combination of local features—such as color, shape, and texture—is performed with CNN features by incorporating scale invariant feature transform (SIFT) to detect the potential image contents. This is interested phenomena that exposes the combination of deep learning features with local features to find the background and foreground objects.

Image retrieval is a process of image extraction based on salient image features including shapes, texture, colors, and objects. Image retrieval system focuses on foreground and background image features with their semantic interpretation and spatial aspects. Therefore, content-based image retrieval is an intelligent image extraction system that retrieves the results based upon the actual image contents as contrast to text based retrieval [15]. Versatile local features have been proposed in last decades to support object recognition tasks and content-based image retrieval (CBIR). These local features includes SIFT [16] and SURF [17] methods to efficiently match local structures along with thousands of deep features being extracted. Six feature extraction techniques are extensively applied including image segmentation, color features, shape features, texture features, interest point detectors and combination of visual features. Image segmentation is applied to extract similar regions. For this, the common techniques are contour-based [18], thresholding [19], region growing segmentation [20], grid-based [21], *k*-means clustering [22], watershed segmentation [23], texture-based segmentation [24], statistical model [25], and normalized cut [26]. To focus the color representation, it is better achieved by applying different color models and spatial aspects. Color moments are also applied to match color similarity and indexing. Color space technique is used to convert color bins having their representational frequency. The color histogram of multiple objects with similar colors are also focused with large image databases. Color histogram has a problem to ignores main information like shape and texture. For image analysis, image quantization is attained in RGB space with pixel classification as interior or border pixel localization with four neighbors. In this regard, texture features analysis with their neighbor is not an easy task and rough, irregular, smooth, and random results. For this, the corner edges and similarities techniques are applied to model texture features.

In [27], texture features are extracted by incorporating different techniques which are invariant to shape [28] and rotation in [29]. The spatial texture features face complex indexing and searching process. It applies spectral texture methods for image transformation on frequency image domain using spatial filter banks. The feature extraction is achieved on transformed domain using statistical tools. Consequently, the techniques for spatial filter banks are vigorous to noise and makes prominent than spatial texture features. The fast Fourier transform (FFT) is also applied for spectral analysis to obtain spatial results [30].

The novelty of the presented method is the application of local features in combination with the strength of GoogLeNet, ResNet-50 and VGG-19 architectures for effective features extraction methods. The proposed method introduces a novel technique using Gaussian filters, isotropic filtering, multi-scale filtering, LOG (Laplacian of Gaussian), convolutions, derivatives, scale spacing, Eigenvalue computations, and feature reduction. The presented approach applies Gaussian filters whose results are purified with Laplacian of Gaussian. To produce the results on which Isotropic filtering is applied that builds the foundation for multiscale filtering. The multiscale filtering is applied to obtain the deep image filtering at various abstract levels. These features are aggregated with gray level convolution features which are concatenated with first order partial derivatives. To obtain the best texture values, Eigen coefficients are computed for these signatures. At an elementary level, BoW kernel function is applied on which non-maximum suppression (NMS) is performed to create the basis for interpolation of corner responses. Determinants are required to compute data points which are already produced by interpolated corner responses. After the computation of determinants, again convolution and derivatives are applied where the difference is that now second order of derivatives are applied instead of first order derivatives to obtain better differences. On this research, Laplacian of Gaussian (LOG) approximation is applied for scale spacing. These features are representative of image objects. On parallel, RGB channels are computed with the selection of color coefficients and normalized values mapping to get the color features. At this step, texture, objects and color features are extracted which are then concatenated with the CNN based trained feature vectors. The final assembly of feature vectors is the representation of textures, objects, colors CNN based image features for all strong image content candidates. Finally, for image retrieval the BoW architecture is applied to retrieve relevant image. The image conversion is performed from color to gray levels 0 to 255 and second level of normalization is applied for all channels, and integrated with produced signatures for presenting the massive feature vectors compactly. Principal component analysis (PCA) is used for redundant feature vectors. The GoogLeNet, ResNet-50, and VGG-19 architectures return the feature vectors for every image for which deep features have already been fetched. The GoogLeNet, ResNet, and VGG-19 based feature vectors are fused with deep features extracted from presented technique. These powerful features are combined to find the tiny, large, complex, overlay, texture, color, object, shape, mimicked, plain background, and multiple-object-based images. These image feature vectors are used as input to the BoW for efficient image indexing and image retrieval. To test the competitiveness of the presented method, it is experimented on ten standard image benchmarks such as Cifar-100, ALOT (250), Oxford Buildings, Cifar-10, FTVL Tropical Fruits, Corel-1000, Fashion (15), Caltech-256, Corel-10000, and 17-Flowers which contains millions of images from versatile semantic groups. The presented work has following novelties and contributions:

1. It comprehensively analyses and collects image contents such as color, object, texture, shape, and spatial information which produces significant recall and precision rates.
2. To attain better improved results, a method is introduced that improves the capabilities of ResNet-50, VGG-19, and GoogLeNet architectures using its internal coupling.
3. A light-weight description model and feature detection is presented that efficiently and effectively retrieves the related images from cluttered and complex databases.
4. First time proposed a technique that performs multilevel scaling, suppression, interpolation, determinants and scale spacing collectively to attain the detail of deep finer image contents.
5. Presented a color and gray level features based retrieval system to encompass the image features for edge, corner detection capabilities with color carrier candidate features.
6. The presented technique introduces a methodology that effectively returns significant performance on similar textures, overlay ambiguous objects, tiny objects, resize images, mimicked, color dominant arrangements, cropped objects, cluttered patterns, overlay ambiguous objects, color dominant arrangements, and similar textures.
7. A storage, computation and time efficient image retrieval system is presented that retrieves the images in fraction of time.
8. A new idea is presented that strengthens the normalized scaled features with BoW architecture for quick classification and indexing.

The remaining article is organized as follows. The related work about CNN with deep learning is presented in Section 2. The presented methodology is explained in Section 3. In Section 4, the experimental results which presents with graphs and tables, also discusses the results of experiments. In Section 5, the conclusion of the presented method is discussed.

2. Related Work

The Convolutional Neural Networks (CNN) have been researched by many existing methods to achieve highest performance for different applications. The recent trends are focused on understanding the Deep Neural Networks (DNN) performance for different complexity levels. In [31], a novel technique is presented to show the minor changes in results of image classification. Moreover, networks can be improved by adding new layers, and by contributing new computer vision algorithms [32]. The researchers introduced a feature extraction technique for image retrieval and image representation using CNN in [33]. The experimentation is performed on Cifar-100 and Cifar-10 benchmarks. For image classification using deep networks, a PCA approach is proposed in [34]. In [35], factorization is performed on similar pairwise semantic matrices into hash codes approximation for training images. Moreover, for efficient and effective image retrieval, CNN is used to make binary hash codes in [36]. Karhunen–Loeve transform (KLT) is used in discrete cases to present the stochastic processes in proper conditions [37]. An effective algorithm is presented in [36] for fast calculation for KLT operator. This fast KLT calculates eigenvectors efficiently when this is treated with small samples. In [38], deep learned features are computed by finding the symmetry in FAST scores with neighborhood, smoothing, and standard deviation. Feature scaling, reduction, and filtering is applied to resize the features for a variety of datasets. This approach indicates the potential research outcomes if it is compared with existing ResNet architecture.

A fusion technique presents the ResNet, VGG, and GoogLeNet models [39] with an interconnection. The SVM and random forest techniques are applied for image classification. The experimentation is performed to achieve better performance on Standford 40 actions. In [40], an approach is proposed that is based on deep learning using five stages including image pre-processing, pre-trained CNN, semantic segmentation, query analysis, and image retrieval for NTCIR13Lifelog-2 dataset. The stemming for image labels from deep neural networks (DNN) in [41], researchers are applied AlexNet [3] and GoogLeNet for object recognition on ImageNet dataset. Moreover, the AlexNet, ResNet [5], GoogLeNet, and VGG [3] architectures are used for scene recognition on places365. The histogram of oriented gradients (HOG) method is employed to detect number of person in each image. In [42], a new technique is performed for comparison and dissimilarities of latest CNN algorithms including AlexNet, VGGNet, GoogLeNet, and ResNet on BelgiumTS benchmark. In addition, an object detection technique is presented to find better processing speed and object recognition ratios among CNN. Symmetric solution by using SVD, PCA, VGG, and Gaussian mixture model (GMM), and highest dimensional feature extraction, feature selection, region segmentation, and softmax respectively in [43]. Furthermore, feature selection for SVD and PCA using FC layers validated the accuracy of images classification.

Various algorithms and methods are proposed for feature extraction, description and detection, and object detection and classification. Research literature is shared to induce the similarity among object detection, filter banks, and texture filters. The implementation of semantic concept this technique is relatively hard, and difficult to understand the overlapped and cluttered objects. Therefore, the previous approaches are focused on image classification and single object detection for feature set [44]. The researchers in [45] focused their research on lower level feature extraction, and object recognition, including filter banks, HOG, GIST, and the bag of feature (BoF) applied using word vocabulary.

The content-based image retrieval (CBIR) is used visual features including image edge, name suitability, color, and texture in input images [46]. The CNN is applied for image classification to retrieve images using cosine similarity. The CNN approach has demonstrated successful image retrieval tasks based on image classification. Many researchers contributed to the image processing task, the semantic gap among the lowest level of image features. In [47], a robust technique of CNN and sparse representation was proposed. Moreover, a novel technique is presented with an in-depth feature extraction using CNN, and increased image retrieval accuracy and speed using sparse representation. In this method is tested for image retrieval on MPEG7, ALOI, and Corel databases. Recent research has been deploying CNN for various types of object-based image classification. However, how to efficiently take advantage of deep CNN features with a trained network to increase the object-based retrieving images still required more research.

The presented method is formulated with best suited approach for efficient image retrieval. The Gaussian filters whose results are purified with LOG. Multi-scale filtering is used after Gaussian filtering in presented technique. Derivatives and Eigenvalues are also used in presented method. NMS is performed to create the basis for interpolation of corner responses. LOG approximation is applied for scale spacing. CNN is applied to reduce features, and the image classification is performed by ResNet-50, VGG-19, and GoogLeNet. Spatial mapping and L2 normalization are applied to normalize images for searching and indexing purposes. The GoogLeNet, ResNet-50, and VGG-19 based feature vectors are fused with the deep features extracted from the proposed method. These powerful features are combined to find out overlay, texture, color, object, shape, mimicked, plain, complex and occupied background, multiple objected foreground images from the challenging image benchmarks. BoW is applied for efficient image retrieval and index. The presented method shows outstanding performance for all challenging benchmarks including Cifar-10, Caltech-256, Corel-1000, Oxford Buildings, Cifar-100, FTVL Tropical Fruits, Corel-10000, 17-Flowers, ALOT (250), and Fashion (15).

3. Methodology

In Figure 1, the presented architecture inputs an image and converts it into gray level to extract Gaussian, Laplacian, Isotropic, and scale-based features mapping. Moreover, the convolution and derivations are encapsulated of Eigen values to input for bag-of-words architecture to show the separation effects with interpretation. In parallel, data blots with determinants are computed at second order with LOG and convolved values. These feature vectors are fused and assembled with RGB maps, colors, norm features and concatenated with ResNet-50, VGG-19, and GoogLeNet separately for quick indexing and retrieval.

3.1. Image Channeling

At the first and foremost step, is to find out the key points in the image. For this, image conversion is performed from color to gray levels 0 to 255 and that is input for first level of processing. The noise is removed from gray scale image by applying primitive techniques such that the gray scale image is formed with high and low intensity represented by white and black levels.

3.2. Isotroping and LOG Filtering with Derivations

In this step, The MR8 filter bank [48] is applied with 38 filters over eight responded filters to obtain texture orientation, neighboring and patterns. The arrangement of these 38 filters consist of eight Laplacian of Gaussian (LOG) filters along with four Gaussian filters to form 12 filters to be applied at three scales and generates 36 filters. These 36 filters are concatenated with first and second order of Gaussian to form 38 filter bank. To achieve the rotational invariance, the filters are employed at different orientations and scales. At each scale, the maximum response among multiple orientations is preserved. The final response at each position is an eight-dimensional feature vector having three scales for bar filters and edge, plus two for isotropic [49]. The filters comprise Gaussian, and Laplacian of a Gaussian (LOG) filter at scale $\varphi = 10$, a bar (second derivative) and also an edge filter of first derivative at six orientations and similar three scales respectively (φ_h, φ_p) = $\{i, j, k\}$ where $i = (1,3)$, $j = (2,6)$ and $k = (4,12)$. To obtain equal insensitive and mono directional context, isotroping is applied on the determined filters responses. These are collapsed at every scale due to filter responses through orientations using eight responses of filters to ensure rotational invariance in filter bank. Moreover, the filter bank of MR4 [50] is employed (φ_h, φ_p) = k scale [51]. MR4 contains 14 filters in which four are responded to obtain neighboring, pattern, and texture orientation. This classifier is used to complete the local neighbor values for pixel filter responses. The small neighborhoods with 3 × 3 size over to 7 × 7 attain higher categorization performance for multiple scale filter banks. It is important to distinguish between texture classes referred as joint classifier. The texture pattern analysis is a potential carrier for background and neighborhood similarities and helpful in semantic interpretation based upon texture distributed. Equation (1) defines for Markov random field (MRF) [51]

$$\gamma(W(h_e) \mid W(h), \forall_h = h_e) = \gamma(W(h_e) \mid W(h), h \in Ŧ(h_e)) \tag{1}$$

In Equation (1), h_e is used as a site in two-dimensional integer lattice, W represent image and $Ŧ$ (h_e) is the neighborhood of the site. It is defined $Ŧ$ as $Ŧ \times Ŧ$ the square neighborhood. The value of central pixel is significant but its distribution must coordinate to its neighbors. To test this distribution conditioned, retain classifiers on the feature vectors from the set of $Ŧ \times Ŧ$ neighborhoods with left out central pixel. Classification ratios for $Ŧ$ = 5 are improved when left out the center pixel and slightly worse for cases of $Ŧ$ = 3 and $Ŧ$ = 7. The joint distribution is sufficient to validate MRF model of the textures in dataset. It is an explicit model containing $\gamma(W(h_e) \mid W(h), h \in Ŧ\,(h_e))$. At this point, textons are used to define central pixel joint probability density function (PDF) conditioned and their neighbors. The center pixels are applied for feature vectors in $Ŧ_2 - 1$ dimensional space by similar dictionary for q textons for every m textons is putative elementary units of texture where q = 610, one-dimensional distribution of central pixels learns by t bin histogram. The joint PDF is represented by $m \times t$ matrix. Furthermore, a Gaussian filter property is that it is an operator which satisfying uncertain relation in Equation (2) [52]

$$\Delta_z \, \Delta_v \geq \frac{1}{2} \tag{2}$$

In Equation (2), Δ_v and Δ_z are variances in frequency and spatial domains respectively. Also, this property is allowed the Gaussian operator for provide best tradeoff among conflicting localization areas in frequency and spatial domains. The two-dimensional Gaussian filter uses rotational symmetric filter; that is a separate coordinate. The separability is significant for efficient computation when applying operation of smoothing using convolutions in the spatial domain. Moreover, the optimal smoothing filter is localized for both frequency and spatial domains for images, thus satisfies the uncertain relation which is given in Equation (2). Consider two dimensions Gaussian operator defines in Equation (3) [52]

$$s(z,r) = \frac{1}{2\,(3.14)\varphi^2}\, e^{-(z^2+r^2/2\varphi^2)} \tag{3}$$

In Equation (3), φ is used for standard deviation and (z,r) are Cartesian coordinates for image. Gaussian filters are applied for various scales to image in Hildreth and Marr, a set for images with various smoothness levels is obtained. For image edges detection, it is also essential to find zero-crossings for the second derivatives. The Laplacian of a Gaussian (LoG) function is used as filter in Equation (4) [52]

$$\begin{aligned} \nabla^2 s(z,r) &= \tfrac{d^2}{dz^2}\, s(z,r) + \tfrac{d^2}{dr^2}\, s(z,r) \\ &= \tfrac{z^2 + r^2 - 2\varphi^2}{2(3.14)\varphi^6} e^{-(z^2+r^2/2\varphi^2)} \end{aligned} \tag{4}$$

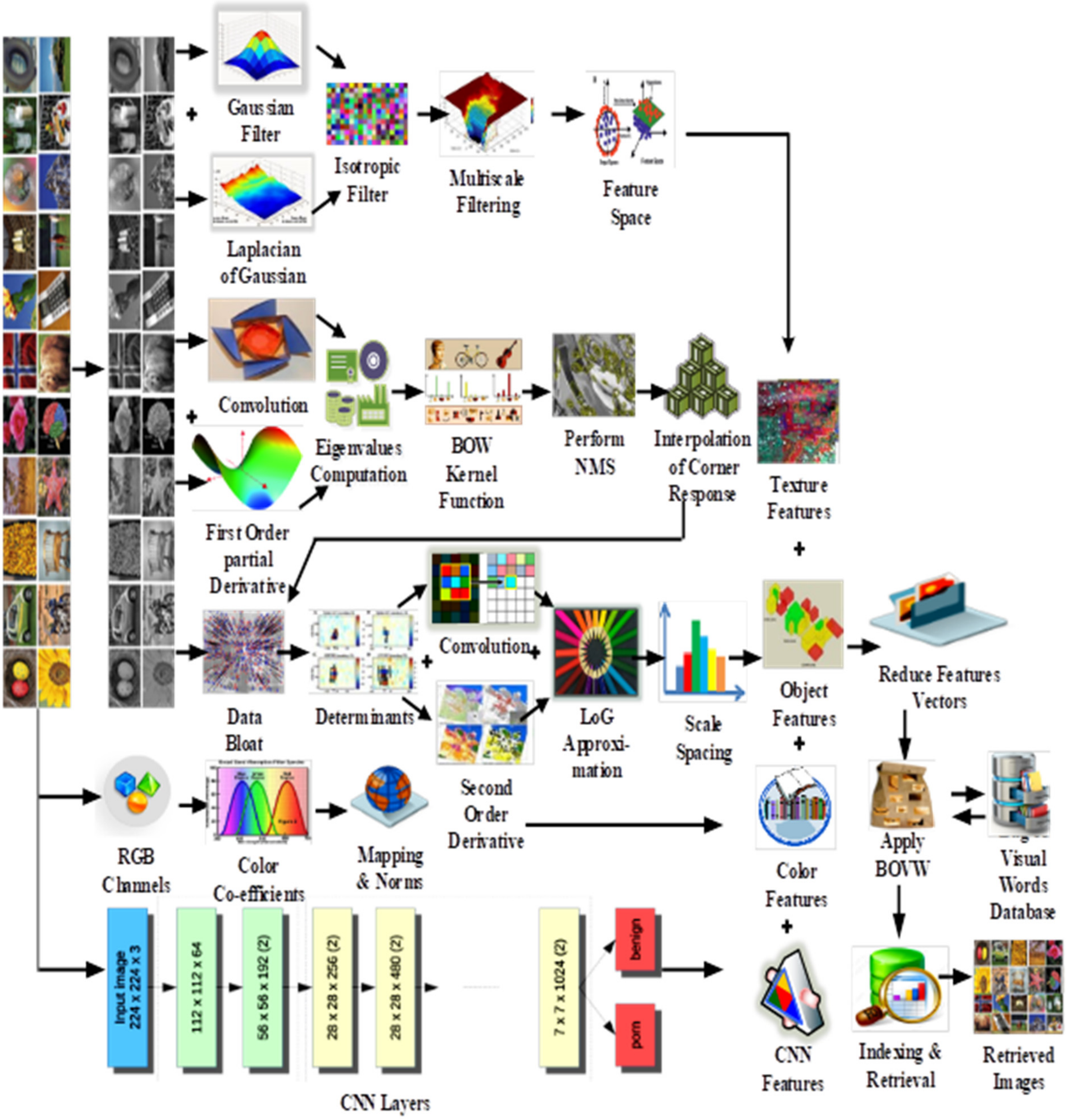


Figure 1. Step-by-step process of object detection.

In Equation (4), it is an operator of independent orientation and the scale is assumed by φ. It is broken down at locations, at curves, and corners, the image intensity function changes in non-linearity manner beside an edge. In Algorithm 1, it computes the square neighborhoods for the intensity values which are already joints in step-1 and step-2. Then, it computes the central pixel with Ŋ $\times$ Ŋ neighbor which is classified for odd numbers in step-3 and step-4. The feature vectors are computed for $Ŋ^2 - 1$ for which ңand ω bin are represented in step-6 and step-7. Finally, the co-occurrence representation is shown for filter banks and probability is computed in step-8 to step-10.

Algorithm 1 *Neighborhood computations algorithm.*

```
Step-1: Square neighborhood preferences calculation
Step-2: Jointly introduce the intensity values
Step-3: Compute Ŋ × Ŋ neighborhoods with excluding central pixel
Step-4: Test classification for Ŋ = 3,5,7
Step-5: Include central pixel and retest step 2–5
Step-6: Form Ŋ2 1 feature vector
Step-7: Show in ңvalues ω bin representation
Step-8: Compute co-occurrence representation of filter banks
Step-9: Compute joint probability P (İ)
Step-10: Compute conditional probability
P (Ç (yˆ) | Ç (ʹH (yˆ))
```

3.3. *Eigen, Harris, Gaussian Coefficients Composition*

At this step, the KLT [37] and Harris algorithms [53] are applied to identify corner pixels of an image. The first order Gaussian derivative kernel approximates using box kernel to speed up the algorithm on which convolution is computed at with less resources on integral images. The integral images of $u^2{}_x$, $u^2{}_y$, and $u_x u_y$ are used to increase the computation of cornerness response. The cornerness concentrates computation, and time consumption for Harris algorithms. Furthermore, the $D(\gamma)$ cornerness of pixel is estimated by adding squares, and multiplies gradients with integration window N such as shown in Equation (5). Both algorithms measure the alteration in intensities because shifts the N for each directions are given points in the cornerness response of corner pixels are defined as in Equation (5) [53]

$$\begin{aligned} D(\gamma) &= \sum_{h\in N}\left\{\begin{bmatrix} u_h^2(H) & u_h(H)u_p(H) \\ u_h(H)u_p(H) & u_p^2(H) \end{bmatrix} \times \pounds\,(H)\right\} \\ &= \begin{bmatrix} U_{hh} & U_{hp} \\ U_{hp} & U_{pp} \end{bmatrix} \end{aligned} \tag{5}$$

with,

$$u_i = \varphi_i\,(u \otimes W) = (\varphi_i u) \otimes W,\ i \in h, p),\ u = U\,(h, p;\ \phi) \tag{6}$$

Here, $\pounds\,(H)$ is used for weighted function, γ is for central pixel, N is used for the integration window centralized at γ, W is an image, u is used for two dimensional Gaussian function, and u_h and u_p are gradients for image attained using convolution for Gaussian first order partial derivatives for h and p directions in Equation (5). The Harris corner detector is used to evaluate cornerness for each pixel without explicit decomposition of eigenvalues as described in Equation (7) [53]:

$$A = |D| - ƙ \times (\text{Trace}\,(D))^2 \tag{7}$$

with,

$$|D| = ƛ_1 \times ƛ_2 \text{ and Trace}\,(D) = ƛ_1 + ƛ_2 \tag{8}$$

where $ƛ_1$ and $ƛ_2$ are eigenvalues of D. The value of $ƙ$ is taken between 0.04 to 0.06. The image pixel is like a corner if the eigenvalues are large, the resultant A in a peak response.

The $D(\gamma)$ eigenvalues are calculated by KLT. It is selected for points that computed the minimum eigenvalue as defined in Equation (9) [53]

$$A = ƛ_{\min} = \min\,(ƛ_1, ƛ_2) \tag{9}$$

$$ƛ_{\min} = \frac{1}{2}\left(U_{hh} + U_{pp} - \sqrt{(U_{hh} - U_{pp}) + 4 \times U_{hp}{}^2}\right) \tag{10}$$

The KLT and Harris algorithms uses similar method to detect the corner points, the difference is only the cornerness functions that is estimated in Equations (7)–(10) respectively. The KLT and Harris algorithms are divided in three steps as:

(a) The Gaussian derivatives for image W, u_h, and u_p are calculated using convolution with kernel of Gaussian derivative;

(b) The $D(\gamma)$ matrix and cornerness measure A evaluate individually of every pixel; and

(c) The NMS and Quick-sort are applied to suppress local low points.

The complexity reduces with two aspects. First, the integral image reduces the complexity for both evaluation and convolution of cornerness response. Second, the efficient NMS adopts of NMS, then avoid high complex sorting.

3.4. First Ordering, Kernel Approximation, and Recursive Filtering

At this stage, detection of corner feature point is required for computation of Gaussian first order partial derivatives, u_h and u_p, in directions of h and p with image W respectively. Moreover, the Gaussian kernels should be discrete and approximated with finite instinct response filter. It is required at least 4ϕ filter length. The Gaussian derivative kernel results improved achievement of the Harris corner detector using better repeatability. The Gaussian derivative kernel is applied recursively for fast computations of convolutions. The Gaussian derivative filter is estimated with infinitely impulse of response filter [54] using recursive filter approach, which is allowed to set the length of filter. Using this method, computation for Gaussian derivative kernel of various scales are applied in time constantly. In SURF [55], first order Gaussian partial derivative kernel with box kernel is approximated. The gray areas are set as 0 in it. Also, the white and black areas are estimated using +1 and -1, respectively. The gradients are computed at low computation cost and also in constant time with integral image. Finally, only seven operations and eight memory accesses need to calculate gradient. Furthermore, the filter response normalizes with the size of filter. The convolution using Gaussian derivative kernel integrates two steps including low-pass filtering using differentiation and Gaussian. Next, Gaussian function is approximated using a triangle function subsequently integration with integral image method. Furthermore, the operations number are essential for box kernel that is independent for kernel size, then linearly increase with the kernel size. The multiplication operations reduce to 1 and other multiplications replace by subtraction and addition operations.

3.5. Multiple Gradients Responses

The cornerness step is the most intensive in computation and also time consumption for KLT [37] and Harris algorithms [53]. The cornerness $D(\gamma)$ for pixel evaluates by adding squares and multiplies of gradients in N as in Equation (5), which evaluates at each pixel. Therefore, the computation overlapping among pixels is lying in N occurs. It creates integral image for gradients such as in Equation (5), accelerating computation for corner response. The following Equations (11)–(13) are defined in [53]

$$jj_{hh}(h,p) = \sum_{h' \le h,\ p' \le p} u_h^2\ (h',\ p') \tag{11}$$

$$jj_{pp}(h,p) = \sum_{h' \le h,\ p' \le p} u_p^2\ (h',\ p') \tag{12}$$

$$jj_{hp}(h,p) = \sum_{h' \le h,\ p' \le p} u_h\ (h',\ p')h_p(h',\ p') \tag{13}$$

From Equations (11)–(13), the integral images are created like summations U_{hh}, U_{pp}, and U_{hp} in Equation (5), evaluated at low computational cost with four memory accesses and three operations. The repeated summation and multiplication operations in N from image of each pixel is exchanged with one-time creation for integral image; and after that easy operations of subtraction and addition. Furthermore, no loss is observed for detector efficiency with modification and contribution using huge speedup. The speedup is achieved by additional memory access for jj_{hp} which not exist in original algorithm. The $u_h u_p$ can be computed at same time for the present CPU internal registers and gradients read from memory, that is not for W as it is located and pre calculated in the memory.

3.6. Perform Non Maximal Suppressions with Corner Responses

In this step, the NMS is performed over cornerness response images that preserves a single location of each corner feature point. NMS is applied in two steps [56] for the KLT detector. Firstly, the quick-sort is performed of arranging A cornerness response in descending order over image [56–58], that is a more expensive operation computationally as sorting for the list of response points corresponds with image size. The non-maximal points are blocked using selective robust response points from response sorted list and eliminating low strong response points consecutive in list inside the distance ƈ. Finally, a lowest distance among feature points are required. NMS naive employment of local neighborhood for (2ƈ + 1) × (2ƈ + 1), each cornerness response is leading a highest Harris complexity [59]. Comparison is performed for all pixels lying in integration window (2ƈ + 1) × (2ƈ + 1) to find the maximum point. The maximum point is chosen if it is above threshold and greater than all pixels. This procedure is repeated for each pixel in cornerness response image. In [60], the efficient non maximum suppression (E-NMS) is introduced for efficient extraction of single feature locations for each corner region. The E-NMS is performed NMS in image blocks rather than pixel by pixel, then reduces the complexity of computation. Moreover, the minimum distance enforcement and quick-sort steps are mostly intended for tracking [56] a unique location for each corner region. Furthermore, it switches order by applying the effective NMS, that is low complex computationally, and sort the feature points as cornerness response. The sorting performs on a small number of points, complexity reduces extremely. The E-NMS algorithm works as follows. First, it partitions the image in size blocks (ƈ + 1) × (ƈ + 1). Secondly, the maximum element is found within each individual block. If it passes the test of local maximum and existing threshold, then the feature point location is retained. For the KLT algorithm, sorting is performed for the feature points as cornerness response is related with them. In Algorithm 2, interest points are detected by applying non-maximal suppression on neighborhoods with their variants and determinants as shown in step-1 to step-3. After this, interpolation is computed at different scale factors for which the neighborhood extraction is performed based on contents description in step-4 to step-6. Similarly, variant extraction is computed based on gradients in step-7 and result in Haar wavelet response in step-8. Algorithm 3 shows Haar wavelet response by applying square region with their orientation and splits up in step-1 to step-3. Space based information is preserved with 5 × 5 boxes for which horizontal and vertical directions are computed and increased interval in step-4 to step-7. Geometric aspects are computed with deformation in step-8 and localization in step-9. Finally, results in weighted centered interest points to represent the feature vectors in step-10 and step-11.

Algorithm 2 *Find Interest Points algorithm.*

```
Step 1: Apply cube 3 NMS neighboring
Step 2: Compute variants
Step 3: Separate the determinants
Step 4: Apply interpolation
Step 5: Apply scaling and image spacing
Step 6: Distribution content description ↔ neighboring extraction
Step-7: Gradient ↔ variant extraction
Step-8: Compute first order Haar wavelet response
```

Algorithm 3 *Haar wavelet response effects algorithm.*

```
Step 1: Square region formation
Step 2: Orientation selection
Step 3: Region split up
Step 4: Spatial information preservation
Step 5: 5 × 5 regularly spaced samples
Step 6: Compute $K_x$ for horizontal directions
Step 7: Compute $K_y$ for vertical directions
Step 8: Increase interest point orientation
Step 9: Deformation $\leftarrow$ geometric aspects
Step 10: Localization $\leftarrow$ geometric aspects
Step 11: Result $\leftarrow$ weighted centered interest points
```

3.7. Hessian Blobs Detection

In this step, blob structures are detected by locations for determinants calculation. Consider a point Z = (h, p) for an image W, Hessian Matrix U (h, φ) of h at the scale θ is represented such as in Equation (14) [55]

$$U(h,\varphi) \;=\; \begin{bmatrix} l_{hh}(h,\ \varphi) & l_{hp}(h,\ \varphi) \\ l_{hp}(h,\ \varphi) & l_{pp}(h,\ \varphi) \end{bmatrix} \tag{14}$$

In Equation (14), $l_{hh}(h,\ \varphi)$ is convolution for Gaussian second order derivative $\frac{\partial^2}{\partial h^2}\, z(\varphi)$ and same for $l_{hp}(h,\ \varphi)$ and $l_{pp}(h,\ \varphi)$ with image A of point h. Gaussians are used for the optimal analysis of scale-space [61]. Moreover, Gaussians are used for discretization and cropping. The resultant is the repeated image rotated around of odd multiples $\frac{3.14}{4}$ which is the weakness of Hessian detectors. The repeatability is computed using multiples of $\frac{3.14}{2}$ due to square shape filter. LoG approximations are second order derivatives and evaluated with low computation cost by integral images. The nine square box filters are applied to Laplacian of Gaussian and presented at low scale for computing the blob response maps. These are represented by l_{hh}, l_{pp}, and l_{hp}. The weights are applied to rectangular regions for easy and efficient computation, as in Equation (15) [55]

$$Determinant\ (Lapprox) = lhh\ lpp - (wlhp)^2 \tag{15}$$

In Equation (15), the weight w for filter responses is applied of balancing Hessian's determinant expression. Moreover, filter responses normalize according to their size. This is guaranteed as a constant Frobenius normalization for some filter size and also used for analysis of scale space. The approximation of Hessian determinant is represented for blob response of an image at the location h. The responses are kept in map form for blob response over various scaling level and detected the local maxima. These scale spaces are applied as an image pyramid. The repeated images are smoothen using Gaussian and the sub-sampled according to highest level for pyramid. In [62], Lowe subtracts pyramid layers to make difference for Gaussians of images to find blobs and edges. The box filters can apply for each size at exact similar speed on the real image. This scale space is used to analyze the increase in filter size. Next, the following layers are managed using image filtering along with applying gradually higher masks. The divided scale space is represented as a series of filter response maps attained by convoluting similar input image by increasing filter size. Moreover, each octave is sub-divided to constant number for scale levels. Integral images have discrete nature, the difference of minimal scale among two consequent scales is depended on length l_1 for negative and positive lobes of second order partial derivative in derivation direction (h or p), that is set as size length of third filter. Moreover, for 9 × 9 filter, the length l_1 is 3. It is necessary to increase the size by the minimal of two pixels to obtain uneven size and to assure the existence of central pixel for two successive levels. Re-scaling mask presents round off errors. The errors are smaller than l_1, this type of approximation is conventional. The construction of scale space initiates with

9×9 filter, that computes the blob response for image of smallest scale. Algorithm 4 shows the intermediate response and summed up operations in step-1. All the absolute values are computed and polarity with intensity is calculated in step-2 and step-3, respectively. The differentials for the sub-regions are computed and their aggregate is scale factor with descriptor in step-4 to step-7.

Algorithm 4 *Intermediate response and summed operation algorithm.*

```
Step-1: Compute sum of absolute values
Step-2: Check polarity intensities
Step-3: Compute $|V_x|$ and $|V_y|$
        Where V contains differentials
Step-4: $s_V \leftarrow \sum V_x, \sum V_y$
Step-5: Apply s to all sub-regions
Step-6: Sum Ǵ = $s_1$ + $s_2$ + $s_3$ +———+$s_n$
Step-7: scale factoring ← descriptor ← unit vector
```

3.8. Spatial RGB Mapping

The presented approach introduces a new way of presenting color channels by their coefficients to present the contents of color image in efficient and compact way. For this, a color histogram is captured to show the color distribution. The spatial correlation for changing the color is presented by distance in presented approach. Let £ be an image, i colors in £ are quantized such as $x_{£}, \ldots\ x_i$. Color represents a pixel $Q = (a,\ k)$ can be described in Equation (16) [63]

$$Q = (a, k)\ \in\ £ \tag{16}$$

For computation of distance between pixels $Q_1\ (a_1, k_1)$ and $Q_2\ (a_2, k_2)$ in Equation (17) [63], we define

$$\Big|Q_1\ -\ Q_2\Big| \triangleq\ max\ \{|(a_1\ -\ a_2)|,\ |k_1\ -\ k_2|\} \tag{17}$$

In Equation (18) [63], the color histogram Y of an image I and $n \in [i]$, we define as

$$\begin{gathered} Y_{dn}(£)\ \triangleq i^2.\ Pr\ [Q \in £x_n] \\ Q \in\ £ \end{gathered} \tag{18}$$

In Equation (18), $Y_{xn}\ (W)/i^2$ returns the probability of pixel color Wx_n where W is an image and x_n is a pixel color. The histogram Y is a linear function in image size and it can be calculated as time O(i^2). Let the distance $x \in [i]$ is used to fix a priori. The correlogram for the image £ is represented of a,$k \in [i]$; $k \in [x]$ in Equation (19) [63]

$$\begin{gathered} \text{´H}^{l}_{x_a x_k}(£) \triangleq Pr\ [Q_2 \in £x_a\ ||\ Q_1 - Q_2\ |= P] \\ Q_1 \in £\ x_k \\ Q_2 \in £ \end{gathered} \tag{19}$$

Equation (19) is used to define the spatial arrangement for color pixels in image. ´H is represented by the probability and color pixel x_k at distance P that is away from a given color pixel in Equation (16). Then, the spatial relationship among similar color values is presented in Equation (20) [63]

$$\text{ꞑ}^{P}_{x}(£) \triangleq Q^{P}_{x}.x^{(£)} \tag{20}$$

Equation (20) derives from Equation (18), where ꞑ is denoted by probability of x color pixel and p is used for distance.

3.9. Covariant Selection for Detection

The presented approach shows compact image features; however, to reduce image retrieval time, the features are presented with their coefficients using PCA. After performing different scaling level steps, the proposed approach applies the algorithm to reduce features which applies various mathematical designs to reduce irrelevant images and apply compression on images [48]. The principal component analysis is used to reduce variables, and these coefficients are applied for the measurement of number of factors which are not correlated [64]. The PCA is applied in situation where most of variables are measured for same construct [64] and also applied to process data comprising extraction of few synthetic variables; that is named as principal components. The principal components are sequence of data projection. PCA is employed to compress and reduce dimension for finding high variance based coefficients [64]. Let m be represented by dimensions on a vector v of q_1 random variables and reduce measurement from q_1 to s [65]. The principal component analysis is applied to find out the linear combinations such as $a'_1 v, \;\; a'_2 v \;\ldots\; a\,'_r v$ that have an extreme data variance focuses on un-correlated with last $a'_1 ve$. To solve maximization problem, eigenvectors are $a_1,\; a_2, \ldots.., a_r$ of covariance matrix e that corresponds to s largest eigenvalues. Furthermore, the eigenvalues show variances of principal components respectively and summation results for first r eigenvalues to sum of variances for every q_1 real variables that is denoted by proportion for each variance in real database, applied for s principle components [65].

3.10. Image Retrieval and Indexing Using BoW

In this step, the bag-of-words (BoW) architecture is used of quick image retrieval and indexing. In BoW architecture, each image is denoted with single linear vector. First, BoW model applies the controls for local feature descriptor as Scale Invariant Features Transform (SIFT) [16]. Second, BoW comparison of single vector is applied using dissimilarity score that is simply employed. The SIFT descriptor is represented by patches like numerical vectors. The SIFT is created to collect equal sized dimensional vectors having 128 bits that is represented by one byte for compact and effective image representation. The occurrence count of each visual word is represented in form for histogram of each image. An inverted image index generates efficient image retrieval based on histogram. Each index is represented by one visual word where small parts of an object carry versatile information—including shape, color, and texture. Visual words contain pixel changes with respect to low level features, descriptors and filters. An image identity list also creates to map terms using images. Finally, image ranking is employed to count visual words number shared between indexed images and query image. The highest number for shared words of image takes its rank to the top. The BoW model is not captured the location, spatial information and co-occurrence of visual words. In the presented approach, the spatial color extraction technique is embedded spatial information for feature vectors (FV) at feature extraction time that is the results of more relevant retrieved images.

3.11. Deep Image Networks

The presented method employs the architecture of deep convolutional neural network that is compared with ResNet-50, GoogLeNet, and VGG-19 to check the affectivity and accuracy of presented approach. The presented method uses inception-v1 [4], that is also called GoogLeNet, is a deep lightweight network such as its basic concept is improved performance and efficient computation. The GoogLeNet is relatively less computation cost is a product of two concepts: (1) optimum CNN with sparsity as introduced in [66]; and (2) the dimension reduction with 1 × 1 convolutional layer as presented in [67]. The Inception-v1 components in GoogLeNet architecture are used three filter sizes including 5 × 5, 3 × 3 and 1 × 1 and also a max-pooling layer. For dimension reduction, the 5 × 5 and 3 × 3 filters precedes with 1 × 1 convolutional layer, while the max-pooling layer succeeds with 1 × 1 convolutional layer. GoogleNet architecture designed for efficient computation and reduced the number of parameters. GoogleNet architecture is 22 deep layers when counts layers with parameters and 27 layers when counts pooling. In independent building blocks, there are 100 layers which are used for GoogLeNet architecture. Alternatively, VGG-19 architecture is employed to test its strength that primarily depends on the CNNs model. The VGG-19 employs 16 layers or 19 layers because of its simplicity like 3 × 3 convolutional layers are fixed on the top for increment with the depth level. In VGG-19, for reduction of volume size, the max pooling layers are applied as a handler. Two FC layers are applied with 4096 neurons. In the training phase, the convolutional layers are employed for feature extraction and the max pooling layers connected with some convolutional layers for reduction of feature dimensionality. In the first convolutional layer, 64 kernels were used for the feature extraction from input images. Fully connected layers were applied to prepare the feature vector (FV). The PCA is used to reduce dimensionality and the features selection for image with better results of classification. It is a significant task to reduce the highly dimensional data using PCA. Ten-fold cross validation is used to categorize the DR images depend on softmax method in the testing phase [43]. The performance of presented technique using VGG-19 is used to compare with other features extraction models including GoogLeNet and ResNet-50. Moreover, the presented method uses ResNet-50 architecture to fuse with the proposed feature extraction and detection to achieve maximum accurate results. The presented approach adopts second level non-linearity. The dimensions are essential of equal size as i and Ŗ in Y = Ŗ $(i, \{W_j\}) + i$. The output and input channels are used to apply changes and a linear estimation W_s of shortcut connection performs to match the dimensions in Equation (21) [5]

$$Y = R\left(i,\ \left\{W_j\right\}\right)\ +\ W_t I \tag{21}$$

In Equation (21), the identity mapping appropriates the degradation problem and W_t is employed for matching dimensions. Ŗ $(i, \{W_j\})$ is used to represent multiple convolutional layers.

Using ResNet-50, GoogLeNet and VGG-19, performance of many applications for computer vision have been increased including objects detection and recognition. The feature vectors are fused with the GoogLeNet, ResNet-50 and VGG-19 generated feature vectors to create a powerful image signature that deeply represents the object and shape features.

4. Experimentation

4.1. Databases

The effective and accurate image retrieval (IR) system is experimented on a variety of suitable image databases. Many contributions are essentially domain oriented. Various images datasets are applied to their object information, complexity, spatial color, generic usage of CBIR, versatility, and object occlusion. Experimentation is performed by selecting standardized benchmarks including Cifar-100 [68], Fashion (15) [69], Cifar-10 [68], (250) [63], Corel-1000 [63], FTVL Tropical Fruits [63], Corel-10000 [63], Oxford Buildings [70], Caltech-256 [71], and 17-Flowers [63]. These challenging databases exist to a wide range of semantic images groups. The effectivity and accuracy of results are affected using image characteristics including occlusion, cluttering, size, quality, object location color, and overlapping. The selected datasets characteristics are in different range of images, image classes and classes have various objects types located at foreground and background [72].

4.1.1. Input Process

The system takes a query image that is generally a color image. The conversion is performed on colored image into gray scale 0 to 255 levels for presented algorithm and this converted image is used as input image to CNN. The input image is used from image datasets in input process. In presented work, the input image is selected from Cifar-100, Corel-1000, Cifar-10, ALOT (250), Corel-10000, Oxford Buildings, Fashion (15), FTVL Tropical Fruits, 17-Flowers, and Caltech-256. For the query image, features are extracted, and the bag-of-words architecture is employed to find k-nearest images and index. The robustness and superiority of presented approach is capable to classify the shapes, colors, textures, and objects. The images are tested for testing and training with 30% and 70%

proportions of testing and training respectively. The random images selection is performed from each image category using permutation.

4.1.2. Recall and Precision Evaluation

The two metrics, recall and precision are applied for evaluation of performance accuracy. The calculation of true positive ratio is used as recall and predicted positive values are employed to precision. Then, precision is calculated for each image category using Equation (22) [72] and recall is calculated for each image category using the following Equation (23) [72]

$$Precision = \frac{E_{w(n)}}{E_{u(m)}} \tag{22}$$

$$Recall = \frac{E_{w(m)}}{E_o} \tag{23}$$

where $E_{w(m)}$ is applied at query image to retrieve the related images, $E_{u(m)}$ represents the contrast of query image and aggregates of the available relevant images.

4.1.3. Mean Average Precision Evaluation

The mean average precision (mAP) is mean precision that is computed over all queries. The mAP is described as in Equation (24) [73]

$$\text{mAP} = \frac{\sum_{q=1}^{l} E(q) \times rel(q)}{r} \tag{24}$$

In Equation (24), $E(q)$ is denoted by average precision for top q image retrieval ratios; a binary indicator function rel (q) is equal to one if qth image retrieved rates are related to current query and otherwise zero; and r and l represent number of related results for current query and summation of retrieved images results respectively.

4.1.4. Average Retrieval Precision Evaluation

The graphs of average retrieval precision (ARP) indicate ARP for presented approach of different databases. The ARP is calculated for every category by applying Equation (25) [63]

$$\text{ARP} = \sum_{b=1}^{i} AQ_b | \, i \tag{25}$$

The AQ is applied for the average precision and b is applied of all categories. The ARP is employed to calculate AQ for all classes of every dataset. The ARP graphs are used to show data placement in order where each data bar is denoted by corrected number of retrieved images regardless of the class. The x-axis represents number of categories contrary to AQ. The AQ is decreased slowly when increased number of classes. The reason is that the large number of classes plot a big denominator. The ARP is calculated for databases such as Cifar-100, Cifar-10, Corel-1000, Oxford Buildings, ALOT (250), Fashion (15), Tropical Fruits, Corel-10000, Caltech-256, and 17-Flowers.

4.1.4.1. f-Measure Evaluation

The calculation of f-measure is performed as harmonic mean (HM) for recall (t) and average precision (s) as described in Equation (26) [74]

$$f = \frac{2 \times s \times t}{s + t} \tag{26}$$

Here, f is applied for f-measure in Equation (26). Also, t is used for recall and s for precision.

4.2. Discussion and Results

The experimentation is performed on Core i7 machine (GPU) with 8 GB RAM. MATLAB R2019a is used for experimentation using CNN and image processing toolboxes. The strong and versatile deep image and layer functions are integrated in code-base to reveal the image. Moreover, the computational efficiency and execution time to retrieve images and are computed against inverted file indexing and non-hierarchical searching.

To test the efficiency of the presented technique, the experiments are performed on Cifar-100, Corel-1000, Cifar-10, 17-Flowers, ALOT (250), Corel-10000, Oxford Buildings, Fashion (15), Caltech-256, and Tropical Fruits datasets. The versatility and superiority for presented approach is tested on three different convolutional neural networks including GoogLeNet, VGG-19, and ResNet-50.

4.2.1. Performance on the Cifar-10 Dataset

This dataset [68] comprises of different semantic groups including ships, cats, cars, airplane, dogs, trucks, horses birds, deer, and frogs. This dataset contains 6000 images in every category. The presented technique provides high AP ratios for most of Cifar-10 classes. The sample images of various categories for cifar-10 database is shown in Figure 2.

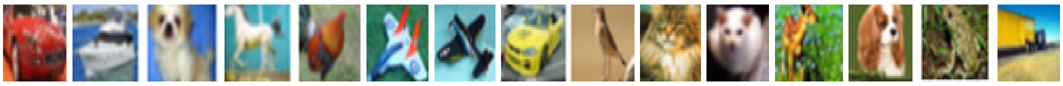
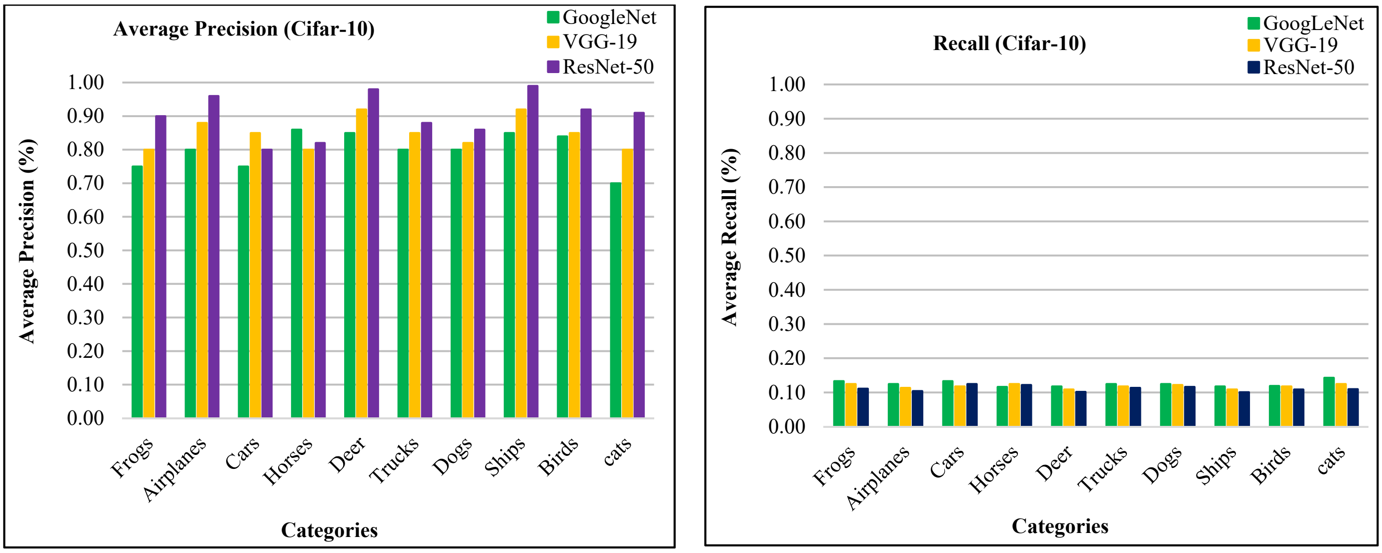


Figure 2. Various sample images for cifar-10 dataset [68].

The images are classified accurately due to CNN feature applied in presented approach. The Gaussian filters and non-maximal suppressions (NMS) with deep learning features make possible to efficiently classify the images from a huge range of image semantic groups—such as ships, trucks, cars, dogs, horse, frogs, deer, airplane and birds. The presented approach is expected image retrieval results with maximum throughput. At this stage, the computational load is very important issue. The presented technique is adjusted it using proper image channeling, isotroping, LOG filtering with derivations, NMS, multilevel filtering, scaling, and feature reduction in different stages; firstly, image channeling in Section 3.1 to convert color image into gray level. Secondly, isotroping and LOG filtering with derivations are applied in Section 3.2. Thirdly, applying first ordering, Kernel approximation, and recursive filtering in Section 3.4. Finally, applying PCA in Co-variants selection for detection in Section 3.9.

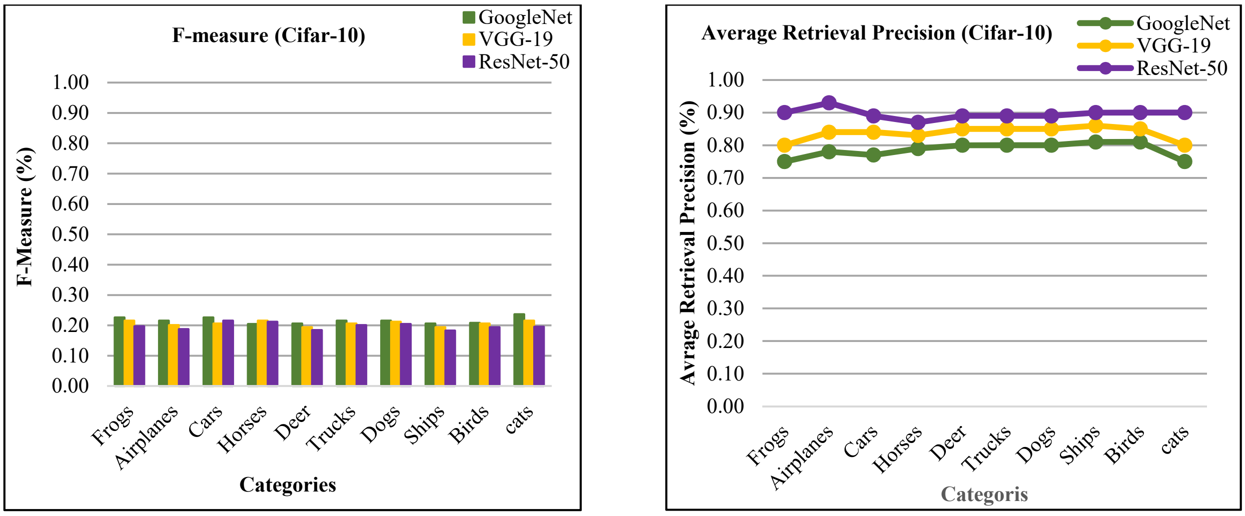
In Figure 3a, the proposed method shows AP results for Cifar-10 database using ResNet-50, VGG-19, and GoogLeNet. Figure 3a reports highest AP ratios using ResNet-50 for most of the Cifar-10 categories. The presented technique shows above 85% AP rates by using features extracted from ResNet-50, above 80% AP ratios using VGG-19 and not less than 70% AP rates using GoogLeNet for most of the semantic groups of Cifar-10 dataset. The presented aooroach is outperformed for the tiny, mimicked, occupied background and multiple objected foreground images due to its object recognition capability. The image classification achieves remarkable AP ratios by presented method for cluttered, complex, and overlapping objects. The proposed approach shows significant AR results for cifar-10 in Figure 3b.



(**a**) (**b**)

Figure 3. (**a**) Average precision for cifar-10 dataset; (**b**) recall for cifar-10 dataset.

In Figure 4a, the presented approach reports outstanding f-measure ratios for the large, mimicked, complex foreground and background images. Moreover, the presented approach provides significant f-measure ratios using ResNet-50, VGG-19, and GoogLeNet. The proposed approach provides 90% mean average precision using ResNet-50, 85% using VGG-19, and 81% using GoogLeNet for Cifar-10. The presented approach reports outstanding ARP ratios using ResNet-50 for the categories including frogs, dogs, ships, airplanes, and birds as shown in Figure 4b. The proposed approach also shows above 85% average retrieval precision ratios for the other categories which reports significant performance for the proposed technique of Cifar-10 database. The presented approach shows better ARP rates by using features extracted from VGG-19 and GoogLeNet.



(**a**) (**b**)

Figure 4. (**a**) f-measure results for cifar-10 dataset; (**b**) Average retrieval precision for cifar-10 dataset.

4.2.2. Performance on the Cifar-100 Dataset

This database is similar as Cifar-10 database using 32 × 32 color images rather it consists of 100 various categories. The set from [68] is comprised of different semantic groups including road, bowls, butterfly, rabbit, mountain, lamp, forest, bus, house, elephant, tractor, tiger, willow, clock, motorcycle, person, palm, rocket, etc. It contains 600 images for every category. Figure 5 shows various sample images for Cifar-100.

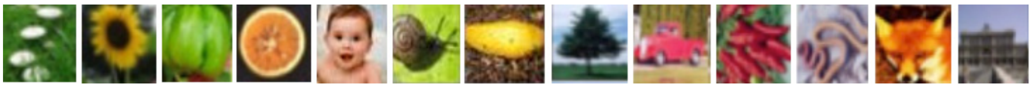


Figure 5. Various sample images for cifar-100 dataset [68].

Table 1 shows significant AP and f-measure ratios for Cifar-100 dataset. The presented approach reports outperformance using ResNet-50 with more than 90% AP ratio for many categories. It is noticed that proposed approach using VGG-19 shows more than 85% AP results in categories including forest, mountain, bus, rocket, butterfly, willow, tiger, person, and elephant. It is also observed that the presented approach using GoogLeNet provides 90% AP rates in motorcycle and palm, 95% in truck and 92% in road category. The strength of the presented approach provides outstanding AP results of Cifar-100.

Table 1. Average precision and f-measure ratios of Cifar-100 dataset.

Cifar-100 Dataset (Average Precision and F-Measure)													
Category	**ResNet-50**		**VGG-19**		**GoogLeNet**		**Category**	**ResNet-50**		**VGG-19**		**GoogLeNet**	
1	0.98	0.18	0.90	0.20	0.85	0.21	51	0.85	0.21	0.80	0.22	0.75	0.23
2	0.75	0.23	0.72	0.23	0.60	0.26	52	0.96	0.19	0.92	0.19	0.85	0.21
3	0.92	0.19	0.96	0.20	0.80	0.22	53	0.80	0.22	0.85	0.23	0.70	0.24
4	0.86	0.20	0.83	0.21	0.80	0.22	54	0.94	0.19	0.90	0.20	0.84	0.21
5	0.72	0.23	0.75	0.23	0.60	0.26	55	0.85	0.21	0.83	0.21	0.75	0.23
6	0.98	0.18	0.96	0.19	0.90	0.20	56	0.98	0.18	0.95	0.19	0.90	0.20
7	0.97	0.19	0.90	0.20	0.88	0.20	57	0.97	0.19	0.94	0.19	0.89	0.20
8	0.92	0.19	0.88	0.20	0.85	0.21	58	0.90	0.20	0.87	0.20	0.85	0.21
9	0.88	0.20	0.80	0.22	0.90	0.22	59	0.88	0.20	0.85	0.21	0.90	0.20
10	0.80	0.22	0.75	0.23	0.70	0.24	60	0.95	0.19	0.90	0.20	0.88	0.20
11	0.86	0.20	0.88	0.20	0.80	0.22	61	0.82	0.21	0.88	0.20	0.78	0.22
12	0.95	0.19	0.86	0.20	0.80	0.22	62	0.86	0.20	0.80	0.22	0.82	0.21
13	0.90	0.20	0.84	0.21	0.78	0.22	63	0.89	0.20	0.90	0.20	0.80	0.22
14	0.70	0.24	0.63	0.25	0.61	0.26	64	0.90	0.20	0.87	0.20	0.81	0.21
15	0.83	0.21	0.78	0.22	0.70	0.24	65	0.93	0.19	0.90	0.20	0.86	0.20
16	0.65	0.25	0.70	0.24	0.60	0.26	66	0.92	0.19	0.88	0.20	0.85	0.21
17	0.70	0.24	0.75	0.23	0.70	0.24	67	0.95	0.19	0.90	0.20	0.88	0.20
18	0.88	0.20	0.80	0.22	0.80	0.22	68	0.88	0.20	0.86	0.20	0.90	0.21
19	0.90	0.20	0.93	0.21	0.70	0.24	69	0.88	0.20	0.84	0.21	0.80	0.22
20	0.96	0.19	0.95	0.19	0.89	0.20	70	0.83	0.21	0.80	0.22	0.70	0.24
21	0.83	0.21	0.78	0.22	0.70	0.24	71	1.00	0.18	0.95	0.19	0.92	0.19
22	0.98	0.18	0.96	0.19	0.90	0.20	72	0.83	0.21	0.81	0.21	0.80	0.22

Table 1. *Cont.*

Cifar-100 Dataset (Average Precision and F-Measure)

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
23	0.83	0.21	0.81	0.21	0.85	0.21	73	0.92	0.19	0.90	0.20	0.80	0.22
24	0.98	0.18	0.95	0.19	0.90	0.20	74	0.98	0.18	0.90	0.20	0.80	0.22
25	0.92	0.19	0.94	0.19	0.83	0.21	75	0.90	0.20	0.87	0.20	0.80	0.22
26	0.85	0.21	0.80	0.22	0.80	0.22	76	0.95	0.19	0.97	0.20	0.80	0.22
27	0.80	0.22	0.78	0.22	0.70	0.24	77	0.98	0.18	0.95	0.19	0.90	0.20
28	0.75	0.23	0.70	0.24	0.60	0.26	78	0.80	0.22	0.70	0.24	0.65	0.25
29	1.00	0.18	0.95	0.19	0.90	0.20	79	0.93	0.19	0.97	0.19	0.90	0.20
30	0.99	0.18	0.92	0.19	0.88	0.20	80	0.98	0.18	0.94	0.19	0.89	0.20
31	0.76	0.22	0.80	0.22	0.72	0.23	81	0.75	0.23	0.70	0.24	0.60	0.26
32	0.96	0.19	0.88	0.20	0.92	0.19	82	1.00	0.18	0.98	0.18	0.93	0.19
33	0.94	0.19	0.90	0.20	0.95	0.19	83	0.92	0.19	0.90	0.20	0.88	0.20
34	0.85	0.21	0.80	0.22	0.75	0.23	84	0.82	0.21	0.75	0.23	0.65	0.25
35	0.97	0.19	0.90	0.20	0.86	0.20	85	0.96	0.19	0.88	0.20	0.82	0.21
36	1.00	0.18	0.94	0.19	0.88	0.20	86	0.92	0.19	0.80	0.22	0.88	0.20
37	0.80	0.22	0.82	0.21	0.75	0.23	87	0.98	0.18	0.95	0.19	0.90	0.20
38	0.87	0.20	0.85	0.21	0.80	0.22	88	0.94	0.19	0.90	0.20	0.88	0.20
39	0.88	0.20	0.80	0.22	0.83	0.21	89	0.93	0.19	0.96	0.19	0.90	0.20
40	0.98	0.18	0.90	0.20	0.89	0.20	90	0.92	0.19	0.85	0.21	0.80	0.22
41	0.85	0.21	0.88	0.20	0.84	0.21	91	0.85	0.21	0.82	0.21	0.82	0.21
42	0.92	0.19	0.85	0.21	0.82	0.21	92	0.80	0.22	0.70	0.24	0.66	0.25
43	0.97	0.19	0.95	0.19	0.88	0.20	93	1.00	0.18	0.96	0.19	0.90	0.20
44	0.96	0.19	0.92	0.19	0.90	0.20	94	0.80	0.22	0.75	0.23	0.75	0.23
45	0.92	0.19	0.96	0.19	0.89	0.20	95	0.96	0.19	0.80	0.22	0.90	0.20
46	1.00	0.18	0.98	0.18	0.92	0.19	96	0.90	0.20	0.82	0.21	0.92	0.20
47	0.98	0.18	0.90	0.20	0.85	0.21	97	0.95	0.19	0.90	0.20	0.88	0.20
48	0.94	0.19	0.90	0.20	0.88	0.20	98	1.00	0.18	0.98	0.18	0.90	0.20
49	0.89	0.20	0.87	0.20	0.85	0.21	99	0.90	0.20	0.92	0.21	0.80	0.22
50	0.75	0.23	0.72	0.23	0.70	0.24	100	0.98	0.18	0.90	0.20	0.86	0.20

In Figure 6a, the presented approach reports 98% mean average precision using ResNet-50, 90% using VGG-19, and 81% using GoogLeNet for Cifar-100 dataset. The presented technique provides the highest average retrieval precision ratios by using features extraction from ResNet-50 for most of the image categories which reports the significant performance of presented approach for Cifar-100 database as reported in Figure 6b. The presented method shows above-80% ARP rates using VGG-19 and better ARP rates using GoogLeNet.

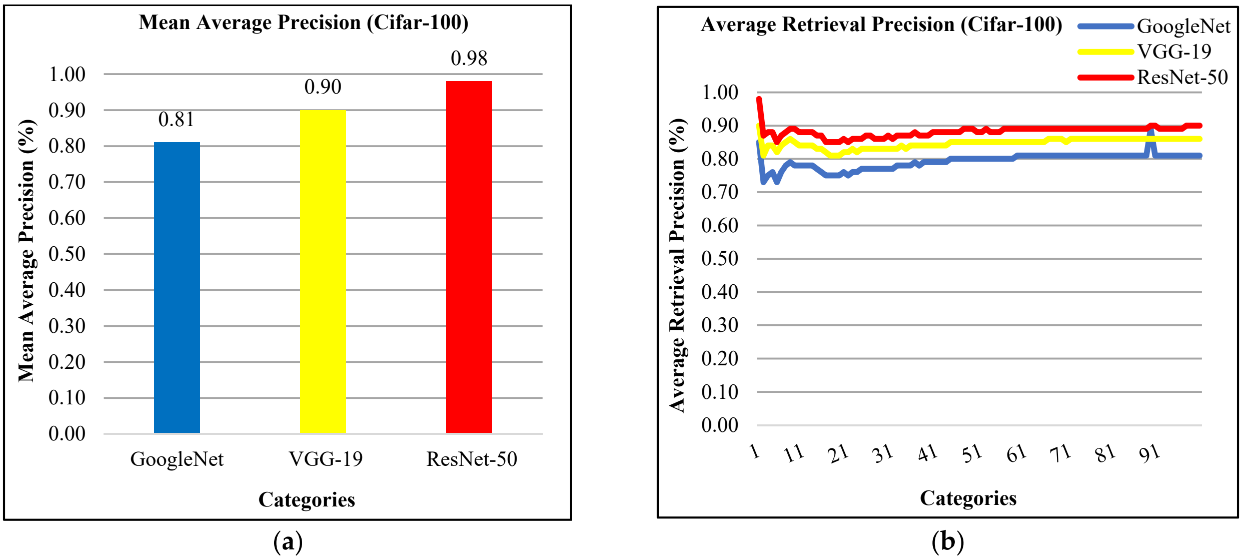


Figure 6. (**a**) Mean average precision for cifar-100 database; (**b**) Average retrieval precision for cifar-100 dataset.

4.2.3. Performance on the Oxford Building Dataset

This dataset [70] contains 5062 images composed from the Flickr to search for specific Oxford landmarks. The collection of images is manually classified into 11 various landmarks images, and the query set comprises of 55 images. The oxford buildings dataset is challenging for detection of cluttered and occlusion background objects. In Figure 7, the sample images of oxford buildings dataset.

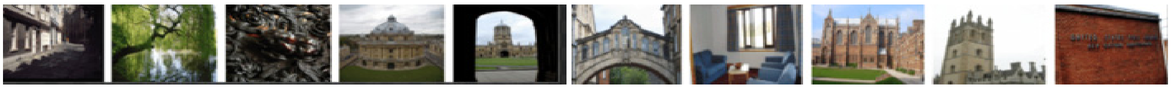


Figure 7. Various sample images of oxford buildings dataset [70].

The presented approach provides significant performance using ResNet-50 with more than 85% AP rates for most of the categories as shown in Table 2. It is noticed that the presented approach using VGG-19 shows more than 80% AP ratios in categories including ashmolean, balliol, chirst church, hertford, jesus, magdalen, oriel, oxford, pitt rivers, Radcliffe, and trinity. It is also observed that the presented technique using GoogLeNet provides remarkable AP rates for most of the image categories. Moreover, the strength of the presented approach presents outstanding AP results for oxford buildings dataset. The presented approach extracts features based on shape and color which reports results accuracy. The proposed approach reports outstanding AR rates for most of the categories of Oxford buildings. Moreover, Table 2 shows f-measure results for oxford buildings database. The presented approach shows f-measure between 19% to 30% for all categories using ResNet-50, VGG-19, and GoogLeNet. In Table 2, the presented approach reports highest ARP rates using ResNet-50 for most of the categories which reports the significant performance of the proposed approach for oxford buildings dataset. The presented approach shows above 80% ARP rates using ReNet-50, 75% ARP rates using features extracted from VGG-19, and 70% ARP rates using GoogLeNet. The presented approach reports 82% mAP using ResNet-50, 90% using VGG-19, and 78% using GoogLeNet for Oxford buildings dataset.

Table 2. Average precision, F-measure, and ARP ratios for Oxford buildings dataset.

Oxford Buildings Dataset (Average Precision, F-Measure, and ARP)									
Category	**ResNet-50**			**VGG-19**			**GoogLeNet**		
All soul	0.65	0.20	0.65	0.60	0.26	0.60	0.58	0.27	0.58
Ashmolean	0.88	0.21	0.77	0.85	0.21	0.73	0.80	0.22	0.69
Balliol	0.82	0.25	0.78	0.75	0.23	0.73	0.70	0.24	0.69
Bodleain	0.64	0.19	0.75	0.60	0.26	0.70	0.50	0.29	0.65
Chirst church	0.95	0.23	0.79	0.90	0.20	0.74	0.87	0.20	0.69
Corner market	0.75	0.20	0.78	0.72	0.23	0.74	0.70	0.24	0.69
Hertford	0.88	0.19	0.80	0.82	0.21	0.75	0.90	0.20	0.72
Jesus	0.96	0.22	0.82	0.90	0.20	0.77	0.88	0.20	0.74
Keble	0.76	0.18	0.81	0.75	0.23	0.77	0.70	0.24	0.74
Magdalen	1.00	0.22	0.83	0.97	0.19	0.79	0.90	0.20	0.75
New	0.80	0.20	0.83	0.75	0.23	0.78	0.65	0.25	0.74
Oriel	0.86	0.21	0.83	0.90	0.20	0.79	0.83	0.21	0.75
Oxford	0.85	0.19	0.83	0.88	0.20	0.80	0.86	0.20	0.76
Pitt rivers	0.94	0.20	0.84	0.94	0.19	0.91	0.90	0.20	0.77
Radcliffe	0.90	0.18	0.84	0.95	0.19	0.82	0.88	0.20	0.78
Trinity	0.98	0.22	0.85	0.92	0.19	0.83	0.84	0.21	0.78
Worcester	0.80	0.20	0.85	0.78	0.22	0.82	0.70	0.24	0.78

4.2.4. Performance on the ALOT (250) Dataset

The ALOT (250) [63] dataset is a challenging benchmark for the image classification and categorization. The ALOT (250) dataset is specially used for texture image classification. Furthermore, all classes are essential aspect for content-based image retrieval. Due to this challenge, the presented approach is experimented on a huge database contains 250 classes to test the versatility and efficiency of the presented technique. The ALOT dataset [63] comprises 250 classes by 100 samples to each. The ALOT database images have 384 × 235 pixels resolution [63]. Semantically different groups of ALOT includes clothes, spices, leaves, cigarettes, sands, vegetables, fruit, stones, sea shells, fabrics, coins and seeds, embossed fabrics, horizontal and vertical lines, bubbles, small repeated patterns, etc. These various classes are contributed for several texture information, object shapes, spatial information and objects to effectively images classification. The proposed technique efficiently classifies texture images from semantic same groups with same large, complex, overlay, texture, background and foreground objects. Gaussian filters, multi-scale filters, color coefficients, and L2 normalization steps are used by the presented model to achieve significant results for images with various textures. The images are efficiently classified by CNN features with partial derivatives, Eigenvalues, and LoG in the presented t approach. Multi-scale filtering on various levels and scale spacing are applied to achieve outstanding average precision ratios for different texture images. Most of the categories of ALOT dataset contain texture images with same colors and patterns where other categories consist of various object patterns. In Figure 8, the various sample images of ALOT (250) database are shown.

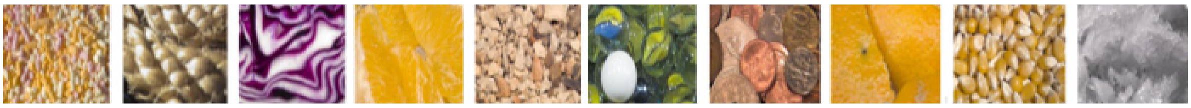


Figure 8. Various sample images for ALOT (250) [63].

Table 3 shows remarkable AP and f-measure rates for ALOT (250) database. The proposed approach shows significant AP results up to 80% using ResNet-50 for most of the ALOT categories. It is observed that the presented approach shows better results for many image categories with various shape and color. Multiscale filtering, spatial mapping, scale spacing, and RGB coefficients with CNN features make it possible to efficiently and effectively classify images. Different images with various categories including fruits, spices, vegetables, and seeds with various color are used. The presented technique shows more than 90% AP rates using ResNet-50, above 85% using VGG-19 and above 80% using GoogLeNet for these kinds of image classes. Also, image semantic groups for bubble textures, horizontal and vertical lines, small repeated patterns, embossed fabrics, and others are classified and experimented accurately. It is also observed that presented technique shows f-measure ratios between 18% to 27% for all categories in Table 3. Moreover, the presented approach reports highest mean average precision using ResNet-50 for ALOT 250 dataset. Overall, mean average precision using ResNet-50 is 89%, mAP by using features extracted from VGG-19 is 84% and mAP using GoogLeNet is 78% for all categories of ALOT (250) database. Moreover, the presented method provides above 85% ARP rates using ResNet-50, above 80% using VGG-19, and above 70% ARP rates by using features extracted from GoogLeNet for most of the image categories of ALOT (250) database.

Table 3. Average precision and F-measure ratios for ALOT (250) dataset.

ALOT (250) Dataset (Average Precision and F-Measure)													
Bubble Textures													
Category	**ResNet-50**		**VGG-19**		**GoogLeNet**		**Category**	**ResNet-50**		**VGG-19**		**GoogLeNet**	
1	0.80	0.19	0.90	0.20	0.85	0.21	12	0.92	0.18	0.90	0.19	0.80	0.20
2	0.94	0.22	0.75	0.23	0.72	0.23	13	0.98	0.20	0.96	0.20	0.90	0.21
3	0.85	0.18	0.90	0.20	0.82	0.21	14	0.91	0.22	0.88	0.24	0.83	0.25
4	0.98	0.19	0.95	0.19	0.90	0.20	15	0.80	0.20	0.70	0.19	0.65	0.20
5	0.97	0.19	0.88	0.20	0.80	0.22	16	0.90	0.22	0.92	0.24	0.90	0.26
6	0.90	0.21	0.86	0.20	0.84	0.21	17	0.76	0.20	0.70	0.20	0.60	0.22
7	0.88	0.20	0.82	0.21	0.70	0.24	18	0.88	0.22	0.86	0.22	0.80	0.24
8	0.95	0.19	0.90	0.20	0.80	0.22	19	0.80	0.21	0.76	0.22	0.70	0.22
9	0.82	0.19	0.89	0.20	0.80	0.22	20	0.85	0.20	0.80	0.21	0.78	0.22
10	0.86	0.19	0.90	0.20	0.78	0.22	21	0.89	0.18	0.82	0.19	0.80	0.20
11	0.89	0.24	0.60	0.26	0.55	0.27							
Stone Textures													
Category	**ResNet-50**		**VGG-19**		**GoogLeNet**		**Category**	**ResNet-50**		**VGG-19**		**GoogLeNet**	
1	0.70	0.24	0.65	0.25	0.60	0.26	14	0.92	0.19	0.88	0.20	0.80	0.22
2	0.85	0.21	0.80	0.22	0.86	0.20	15	0.98	0.18	0.93	0.19	0.90	0.20
3	0.65	0.25	0.60	0.26	0.50	0.29	16	0.83	0.21	0.80	0.22	0.85	0.21
4	0.80	0.22	0.76	0.22	0.70	0.24	17	0.96	0.19	0.92	0.19	0.90	0.20
5	0.90	0.20	0.86	0.20	0.80	0.22	18	0.88	0.20	0.86	0.20	0.82	0.21

Table 3. *Cont.*

ALOT (250) Dataset (Average Precision and F-Measure)

Stone Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
6	0.90	0.20	0.82	0.21	0.75	0.23	19	0.92	0.19	0.87	0.20	0.80	0.22
7	0.60	0.26	0.55	0.27	0.50	0.29	20	0.83	0.21	0.75	0.23	0.70	0.24
8	0.88	0.20	0.80	0.22	0.74	0.23	21	0.85	0.21	0.80	0.22	0.78	0.22
9	0.87	0.20	0.80	0.22	0.73	0.23	22	0.86	0.20	0.83	0.21	0.80	0.22
10	0.98	0.18	0.89	0.20	0.88	0.20	23	0.90	0.20	0.92	0.19	0.80	0.22
11	0.86	0.20	0.82	0.21	0.70	0.24	24	0.86	0.20	0.80	0.22	0.77	0.22
12	0.86	0.20	0.80	0.22	0.75	0.23	25	0.80	0.22	0.76	0.22	0.72	0.23
13	0.94	0.19	0.90	0.20	0.87	0.20	26	0.88	0.20	0.83	0.21	0.80	0.22

Leaf Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.76	0.22	0.72	0.23	0.68	0.24	15	0.76	0.22	0.72	0.23	0.70	0.24
2	0.93	0.19	0.88	0.20	0.85	0.21	16	0.80	0.22	0.78	0.22	0.70	0.24
3	0.96	0.19	0.91	0.20	0.83	0.21	17	0.91	0.20	0.88	0.20	0.80	0.22
4	0.80	0.22	0.74	0.23	0.70	0.24	18	0.88	0.20	0.80	0.22	0.76	0.22
5	0.95	0.19	0.90	0.20	0.80	0.22	19	0.90	0.20	0.87	0.20	0.84	0.21
6	0.86	0.20	0.81	0.21	0.76	0.22	20	0.87	0.20	0.82	0.21	0.80	0.22
7	0.90	0.20	0.88	0.20	0.83	0.21	21	0.86	0.20	0.80	0.22	0.79	0.22
8	0.80	0.22	0.75	0.23	0.70	0.24	22	0.80	0.22	0.70	0.24	0.63	0.25
9	0.96	0.19	0.90	0.20	0.80	0.22	23	0.90	0.20	0.88	0.20	0.86	0.20
10	0.98	0.18	0.92	0.19	0.88	0.20	24	0.91	0.20	0.80	0.22	0.78	0.22
11	0.87	0.20	0.81	0.21	0.78	0.22	25	0.96	0.19	0.90	0.20	0.80	0.22
12	0.92	0.19	0.90	0.20	0.83	0.21	26	0.98	0.18	0.88	0.20	0.83	0.21
13	0.97	0.19	0.92	0.19	0.88	0.20	27	0.89	0.20	0.80	0.22	0.76	0.22
14	0.96	0.19	0.94	0.19	0.92	0.19	28	0.80	0.22	0.78	0.22	0.75	0.23

Fabric Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.90	0.22	0.83	0.21	0.80	0.22	19	0.95	0.19	0.92	0.19	0.82	0.21
2	0.96	0.21	0.90	0.20	0.82	0.21	20	0.98	0.18	0.95	0.19	0.90	0.20
3	0.98	0.20	0.90	0.20	0.88	0.20	21	0.80	0.22	0.75	0.23	0.70	0.24
4	0.94	0.21	0.88	0.20	0.84	0.21	22	0.75	0.23	0.70	0.24	0.60	0.26
5	0.80	0.24	0.75	0.23	0.67	0.24	23	0.90	0.20	0.88	0.20	0.88	0.20
6	0.89	0.22	0.84	0.21	0.80	0.22	24	0.98	0.18	0.90	0.20	0.82	0.21
7	0.86	0.24	0.80	0.22	0.70	0.24	25	0.95	0.19	0.90	0.20	0.88	0.20
8	0.92	0.22	0.86	0.20	0.80	0.22	26	0.90	0.20	0.88	0.20	0.81	0.21
9	0.88	0.24	0.82	0.21	0.70	0.24	27	0.98	0.18	0.95	0.19	0.90	0.20
10	0.98	0.22	0.89	0.20	0.80	0.22	28	0.94	0.19	0.90	0.20	0.80	0.22
11	0.80	0.22	0.82	0.21	0.80	0.22	29	0.96	0.19	0.92	0.19	0.88	0.20

Table 3. *Cont.*

ALOT (250) Dataset (Average Precision and F-Measure)

Fabric Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
12	0.92	0.21	0.90	0.20	0.83	0.21	30	0.86	0.20	0.80	0.22	0.74	0.23
13	0.98	0.22	0.90	0.20	0.80	0.22	31	0.88	0.20	0.85	0.21	0.80	0.22
14	0.96	0.21	0.92	0.19	0.82	0.21	32	0.92	0.19	0.90	0.20	0.80	0.22
15	0.94	0.20	0.91	0.20	0.90	0.20	33	0.98	0.18	0.92	0.19	0.82	0.21
16	0.90	0.22	0.86	0.20	0.80	0.22	34	0.97	0.19	0.92	0.19	0.85	0.21
17	0.78	0.26	0.70	0.24	0.60	0.26	35	0.94	0.19	0.90	0.20	0.80	0.22
18	0.85	0.24	0.80	0.22	0.70	0.24	36	0.89	0.20	0.83	0.21	0.76	0.22

Vegetable Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.98	0.18	0.90	0.20	0.87	0.20	7	0.80	0.22	0.70	0.24	0.66	0.25
2	0.88	0.20	0.80	0.22	0.75	0.23	8	0.90	0.20	0.80	0.22	0.76	0.22
3	0.90	0.20	0.87	0.20	0.80	0.22	9	0.65	0.25	0.60	0.26	0.50	0.29
4	0.92	0.19	0.88	0.20	0.82	0.21	10	0.98	0.18	0.94	0.19	0.90	0.20
5	0.98	0.18	0.93	0.19	0.90	0.20	11	0.70	0.24	0.66	0.25	0.60	0.26
6	0.90	0.20	0.82	0.21	0.74	0.23	12	0.96	0.19	0.90	0.20	0.80	0.22

Fruit Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.92	0.19	0.88	0.20	0.80	0.22	11	0.89	0.18	0.80	0.22	0.70	0.24
2	0.90	0.20	0.87	0.20	0.83	0.21	12	0.93	0.20	0.88	0.20	0.83	0.21
3	0.88	0.20	0.82	0.21	0.73	0.23	13	0.96	0.19	0.90	0.20	0.80	0.22
4	0.86	0.20	0.80	0.22	0.78	0.22	14	0.95	0.19	0.93	0.19	0.85	0.21
5	0.98	0.18	0.90	0.20	0.81	0.21	15	0.97	0.19	0.95	0.19	0.90	0.20
6	0.97	0.19	0.95	0.19	0.82	0.21	16	0.89	0.19	0.85	0.21	0.80	0.22
7	0.94	0.19	0.90	0.20	0.88	0.20	17	0.98	0.20	0.92	0.19	0.88	0.20
8	0.92	0.19	0.91	0.20	0.82	0.21	18	0.95	0.18	0.90	0.20	0.82	0.21
9	0.80	0.22	0.78	0.22	0.70	0.24	19	0.75	0.19	0.73	0.23	0.70	0.24
10	0.99	0.18	0.95	0.19	0.90	0.20	20	0.89	0.23	0.80	0.22	0.70	0.24

Seed Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.78	0.22	0.75	0.23	0.70	0.24	8	0.90	0.20	0.83	0.21	0.80	0.22
2	0.99	0.18	0.94	0.19	0.90	0.20	9	0.89	0.20	0.84	0.21	0.81	0.21
3	0.96	0.19	0.92	0.19	0.88	0.20	10	0.92	0.19	0.86	0.20	0.82	0.21
4	0.92	0.19	0.88	0.20	0.82	0.21	11	0.95	0.19	0.92	0.19	0.83	0.21
5	0.88	0.20	0.84	0.21	0.80	0.22	12	0.98	0.18	0.94	0.19	0.86	0.20
6	0.95	0.19	0.90	0.20	0.83	0.21	13	0.95	0.19	0.92	0.19	0.88	0.20
7	0.80	0.22	0.76	0.22	0.70	0.24	14	0.70	0.24	0.66	0.25	0.60	0.26

Table 3. *Cont.*

ALOT (250) Dataset (Average Precision and F-Measure)

Bean Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.89	0.20	0.85	0.21	0.83	0.21	9	0.95	0.19	0.94	0.19	0.90	0.20
2	0.87	0.20	0.82	0.21	0.70	0.24	10	0.98	0.18	0.92	0.19	0.87	0.20
3	0.90	0.20	0.86	0.20	0.83	0.21	11	0.60	0.26	0.55	0.27	0.50	0.29
4	0.90	0.20	0.82	0.21	0.80	0.22	12	0.80	0.22	0.82	0.21	0.84	0.21
5	0.75	0.23	0.70	0.24	0.60	0.26	13	0.85	0.21	0.81	0.21	0.80	0.22
6	0.98	0.18	0.90	0.20	0.87	0.20	14	0.76	0.22	0.80	0.22	0.70	0.24
7	0.60	0.26	0.55	0.27	0.52	0.28	15	0.90	0.20	0.85	0.21	0.76	0.22
8	0.90	0.20	0.82	0.21	0.76	0.22							

Coin Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.87	0.20	0.80	0.22	0.74	0.23	12	0.80	0.22	0.75	0.23	0.72	0.23
2	0.90	0.20	0.86	0.20	0.82	0.21	13	0.90	0.20	0.86	0.20	0.81	0.21
3	0.94	0.19	0.82	0.21	0.78	0.22	14	0.90	0.20	0.88	0.20	0.80	0.22
4	0.98	0.18	0.86	0.20	0.82	0.21	15	0.80	0.22	0.75	0.23	0.70	0.24
5	0.95	0.19	0.90	0.20	0.80	0.22	16	0.98	0.18	0.95	0.19	0.90	0.20
6	0.94	0.19	0.88	0.20	0.84	0.21	17	0.96	0.19	0.90	0.20	0.82	0.21
7	0.98	0.18	0.90	0.20	0.80	0.22	18	0.86	0.20	0.84	0.21	0.80	0.22
8	0.89	0.20	0.82	0.21	0.75	0.23	19	0.89	0.20	0.83	0.21	0.79	0.22
9	0.87	0.20	0.81	0.21	0.76	0.22	20	0.98	0.18	0.95	0.19	0.88	0.20
10	0.95	0.19	0.92	0.19	0.84	0.21	21	0.89	0.20	0.85	0.21	0.80	0.22
11	0.87	0.20	0.82	0.21	0.70	0.24	22	0.90	0.20	0.84	0.21	0.78	0.22

Sea Shell Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.98	0.18	0.95	0.19	0.85	0.21	11	0.89	0.20	0.83	0.21	0.76	0.22
2	0.98	0.19	0.93	0.19	0.82	0.20	12	0.91	0.20	0.88	0.20	0.82	0.21
3	0.96	0.19	0.92	0.20	0.88	0.20	13	0.95	0.19	0.90	0.20	0.84	0.21
4	0.92	0.20	0.88	0.21	0.86	0.22	14	0.96	0.19	0.92	0.19	0.80	0.22
5	0.90	0.22	0.83	0.22	0.80	0.24	15	0.98	0.18	0.92	0.19	0.90	0.20
6	0.80	0.18	0.78	0.19	0.70	0.21	16	0.80	0.22	0.79	0.22	0.73	0.23
7	0.98	0.18	0.94	0.19	0.83	0.20	17	0.55	0.27	0.53	0.28	0.50	0.29
8	0.99	0.18	0.93	0.20	0.88	0.21	18	0.75	0.23	0.72	0.23	0.70	0.24
9	0.98	0.19	0.90	0.19	0.83	0.22	19	0.86	0.20	0.80	0.22	0.72	0.23
10	0.96	0.18	0.92	0.19	0.80	0.21	20	0.98	0.18	0.88	0.20	0.83	0.21

Spice Textures

Category	ResNet-50		VGG-19		GoogLeNet		Category	ResNet-50		VGG-19		GoogLeNet	
1	0.97	0.19	0.84	0.21	0.80	0.22	19	0.95	0.19	0.91	0.20	0.86	0.20
2	0.92	0.19	0.90	0.20	0.90	0.20	20	0.98	0.18	0.94	0.19	0.88	0.20
3	0.90	0.20	0.82	0.21	0.74	0.23	21	0.89	0.20	0.85	0.21	0.80	0.22

Table 3. *Cont.*

ALOT (250) Dataset (Average Precision and F-Measure)													
Spice Textures													
Category	**ResNet-50**		**VGG-19**		**GoogLeNet**		**Category**	**ResNet-50**		**VGG-19**		**GoogLeNet**	
4	0.75	0.23	0.70	0.24	0.65	0.25	22	0.85	0.21	0.80	0.22	0.70	0.24
5	0.75	0.23	0.73	0.23	0.68	0.24	23	0.98	0.18	0.90	0.20	0.81	0.21
6	0.99	0.18	0.90	0.20	0.87	0.20	24	0.94	0.19	0.92	0.19	0.82	0.21
7	0.60	0.26	0.58	0.27	0.50	0.29	25	0.98	0.18	0.95	0.19	0.90	0.20
8	0.88	0.20	0.80	0.22	0.70	0.24	26	0.88	0.20	0.80	0.22	0.76	0.22
9	0.95	0.19	0.90	0.20	0.84	0.21	27	0.86	0.20	0.82	0.21	0.70	0.24
10	0.80	0.22	0.77	0.22	0.72	0.23	28	0.98	0.18	0.90	0.20	0.83	0.21
11	0.94	0.19	0.90	0.20	0.80	0.22	29	0.94	0.19	0.88	0.20	0.80	0.22
12	0.89	0.20	0.84	0.21	0.80	0.22	30	0.92	0.19	0.84	0.21	0.80	0.22
13	0.86	0.20	0.80	0.22	0.76	0.22	31	0.97	0.19	0.94	0.19	0.82	0.21
14	0.85	0.21	0.80	0.22	0.74	0.23	32	0.88	0.20	0.80	0.22	0.70	0.24
15	0.97	0.19	0.84	0.21	0.80	0.22	33	0.89	0.20	0.84	0.21	0.78	0.22
16	0.92	0.19	0.90	0.20	0.90	0.20	34	0.96	0.19	0.90	0.20	0.76	0.22
17	0.90	0.20	0.82	0.21	0.74	0.23	35	0.99	0.18	0.96	0.19	0.90	0.20
18	0.75	0.23	0.70	0.24	0.65	0.25	36	0.90	0.20	0.88	0.20	0.85	0.21

4.2.5. Performance on the Fashion (15) Dataset

The robustness and versatility of the presented method is experimented by testing fashion (15) database. The fashion (15) database is appropriate for analyzing texture, also it contains images with different color, texture, size, and shapes. The object classes comprise with various fabrics kinds including jersey t-shirt, undergarments, shirt, long dress, robe, blouses, coat jacket, sweater, uniform, cloak, suit, polo-sport shirt, and vest-waistcoat [69]. The fashion database contains above 260 thousand images with various complex, cluttered, background and foreground textures. Figure 9 shows sample images of fashion dataset.

Figure 9. Various sample images for fashion (15) database [69].

The presented approach outperforms for large, cluttered, mimicked and complex occupied objects due to its capability of object recognition. The presented approach is applied significantly image classification and provides improved average precision for complex, cluttered, and overlapping objects. The presented technique reports above 85% AP results using ResNet-50, above 75% using VGG-19, and above 70% using GoogLeNet in most of the categories as shown in Figure 10a. The presented approach is reported outstanding AP results using ResNet-50 for fashion dataset. The average recall ratios are shown in Figure 10b for fashion (15) dataset.

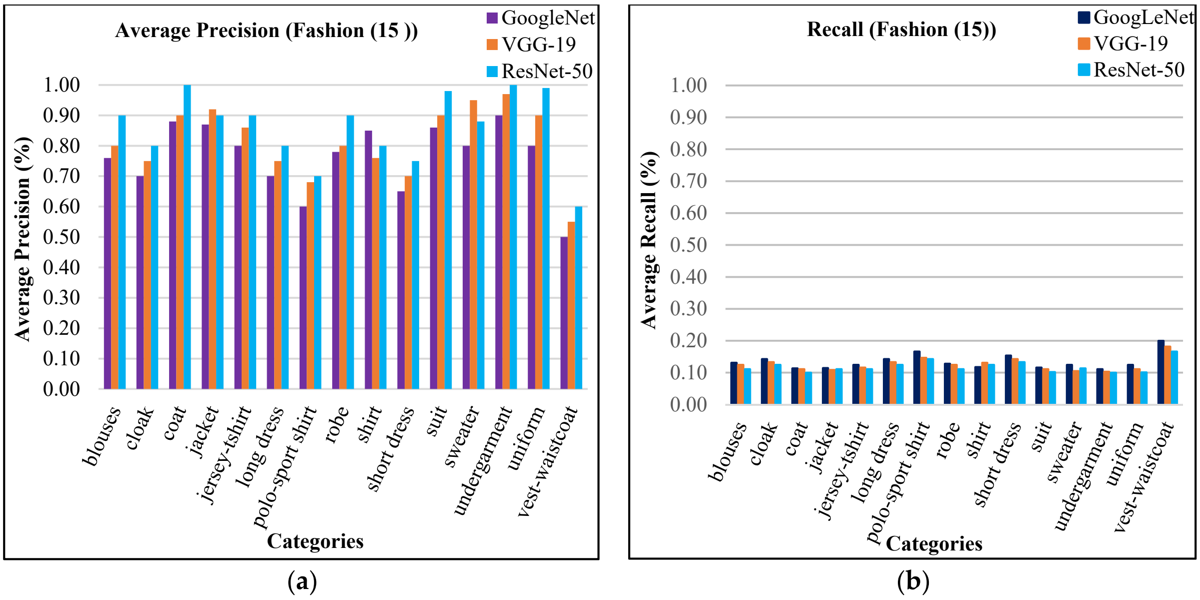


Figure 10. (**a**) Average precision for fashion (15) database; (**b**) Recall for fashion (15) database.

The presented method shows mAP in Figure 11a of fashion dataset. The mean average precision (mAP) using ResNet-50 is 86%, mAP using VGG-19 is 81%, and mAP using GoogLeNet is 76% for all categories of fashion (15) dataset. Figure 11b shows above 85% results by using features extraction from ResNet-50, above 75% ARP rates using VGG-19 and above 70% ARP rates using GoogLeNet for many categories. The presented technique shows remarkable ARP ratios of fashion (15) database using convolutional Laplacian scaled object features with mapped colored channels to obtain the highest image retrieval rates to efficiently classify and index images. Many categories of the fashion dataset—such as jersey-t-shirt, coat, blouses, robe, uniform jacket, and long dress texture—show encouraging performance of the presented technique. Moreover, the presented method reports improved ratios between 18% to 29% for all categories using ResNet-50, VGG-19, and GoogLeNet.

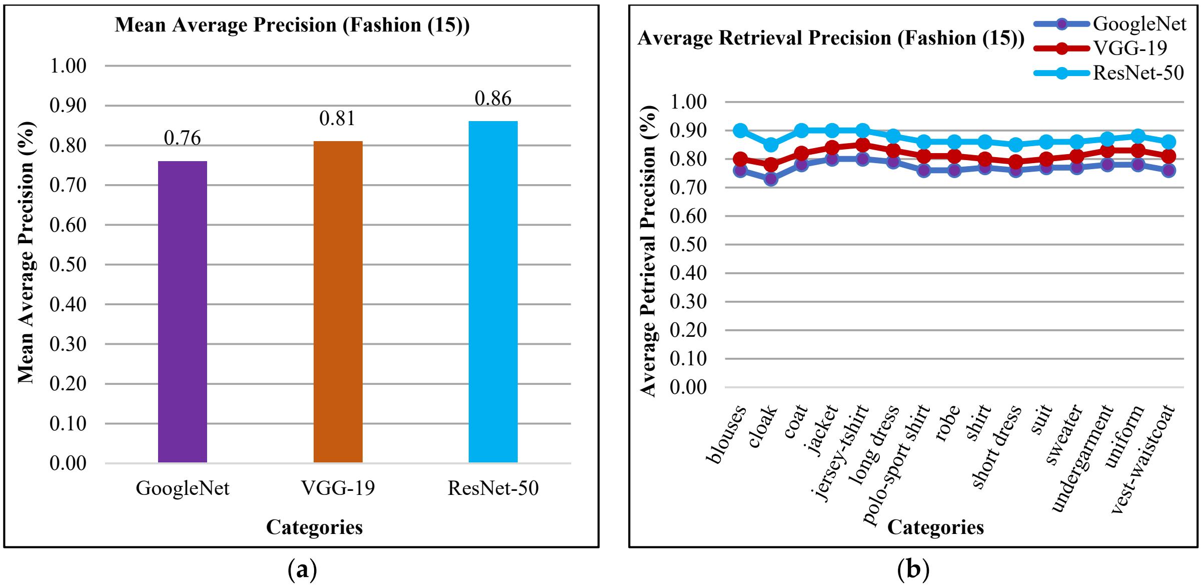


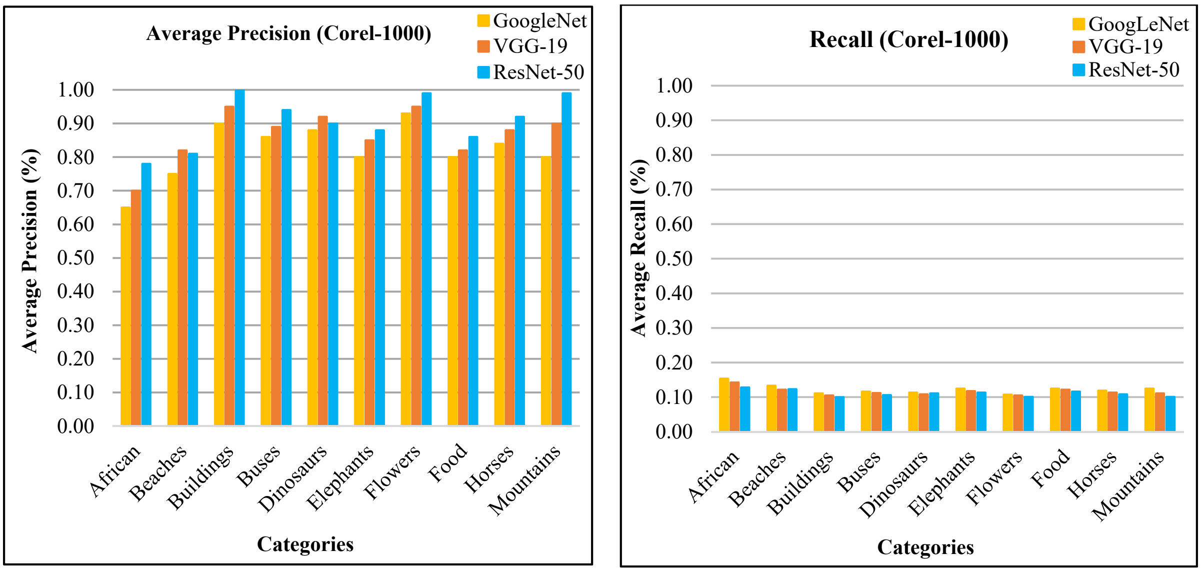
Figure 11. (**a**) Mean average precision of fashion (15) database; (**b**) Average retrieval precision of fashion (15) database.

4.2.6. Performance on the Corel-1000 Dataset

This database is normally applied to retrieving and classifying images [75–77]. The Corel-1000 dataset consists of several image classes comprising plain foreground and background images for complex and cluttered objects. This dataset consists of different semantic groups including buildings, natural scenes, mountains, people food, buses, animals, and flowers. Figure 12 shows various sample images of Corel-1000.

Figure 12. Various sample images for Corel-1000 database [38].

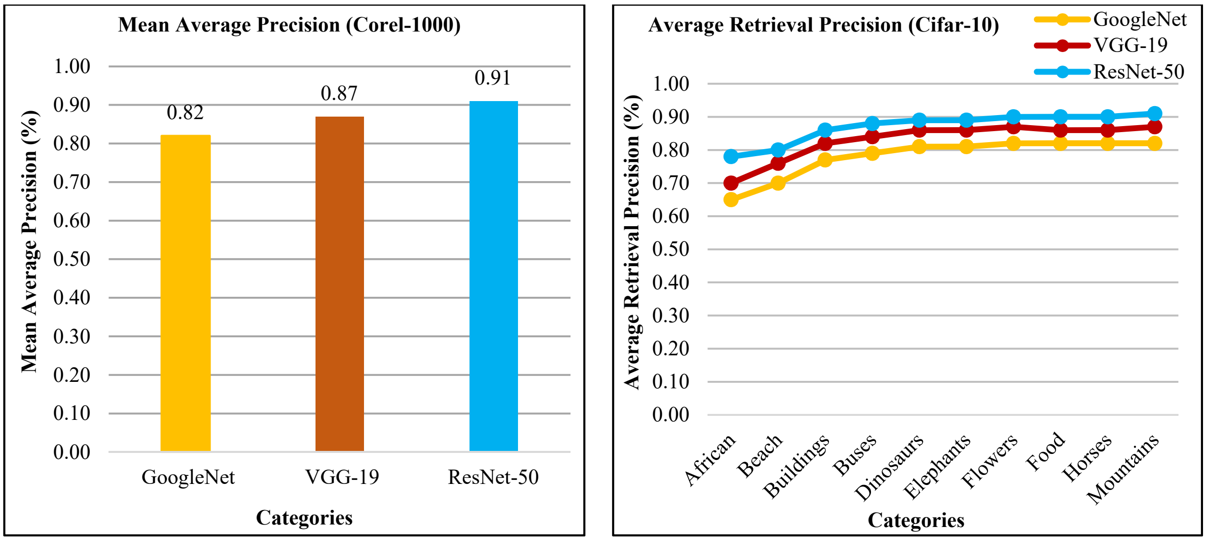
The AP ratios are shown in Figure 13a for Corel-1000 database. The presented approach efficiently and effectively classifies images from various kinds of groups comprising different occupied background, blobs, complex and foreground objects. The spatial mappings with CNN features, image scaling, integration, and multilevel filtering techniques make it possible to proficiently image classification. The AP rates of Corel-1000 database show dominant performance of presented approach for most of the categories due to its L2 norm, scale spacing, RGB coefficient and spatial mappings. The presented approach shows highest performance in many classes such as horse, flowers, buses, dinosaurs, buildings, food, and mountains. For complex foreground and background categories—including mountains, horses, flowers, and buildings—the presented approach reports above 90% AP ratios using ResNet-50. The category buildings and mountains show 100% and 99% AP rates using ResNet-50 respectively. Moreover, all other categories also report more than 77% AP ratios. The presented approach shows AR rates for all categories of Corel-1000 dataset in Figure 13b.



(**a**) (**b**)

Figure 13. (**a**) Average precision of Corel-1000 database; (**b**) Recall of Corel-1000 database.

The presented approach shows the mAP in Figure 14a for Corel-1000 dataset. The presented technique reports 91% mAP using ResNet-50, 87% using VGG-19, and 82% using GoogLeNet. In Figure 14b, the presented method shows above 87% ARP using ResNet-50, above 82% ARP rates using VGG-19, and above 75% ARP rates using GoogLeNet for many categories. The presented approach shows outstanding ARP ratios for Corel-1000 dataset using L2 norm, color coefficient, and filtering to effectively and efficiently image classification and indexing. Moreover, the presented approach shows outstanding f-measure ratios using ResNet-50, GoogLeNet, and VGG-19 for Corel-1000 database.



(**a**) (**b**)

Figure 14. (**a**) Mean average precision of Corel-1000 database; (**b**) Average retrieval precision for Corel-1000 database.

4.2.7. Performance on the Corel-10000 Dataset

The corel-10000 database [38] comprises several image classes. The corel-10000 dataset is comprised of 100 classes where each class consists of 100 images. This dataset comprises different semantic groups—including cars, flowers, texture, shining stars, butterfly, human texture flags, trees planets, ketch, hospital, text, sunset, food, animals, etc. Sample images from this dataset are shown in Figure 15.

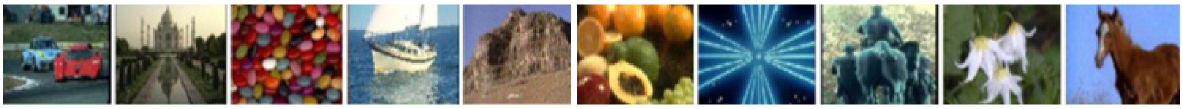


Figure 15. Various sample images for corel-10000 dataset [38].

The presented approach provides significant AP ratios in Figure 16 for corel-10000 dataset. The AP ratio is between 60% and 100%. The presented method reports highest AP results using ResNet-50 with more than 85% AP rates for mostly classes. It is noted that the proposed method using VGG-19 shows above 80% AP results in many categories. It is also noticed that the presented technique using GoogLeNet provides above AP rates 75% in most of the categories. This dataset effectively classifies images with its deep learning feature of the presented technique. The spatial mapping and multi-scale filtering with CNN features make possible to effectively and proficiently image classification. The strength of the presented method is presented outstanding AP rates of corel-10000. The presented method provides better recall rates for all categories of corel-10000.

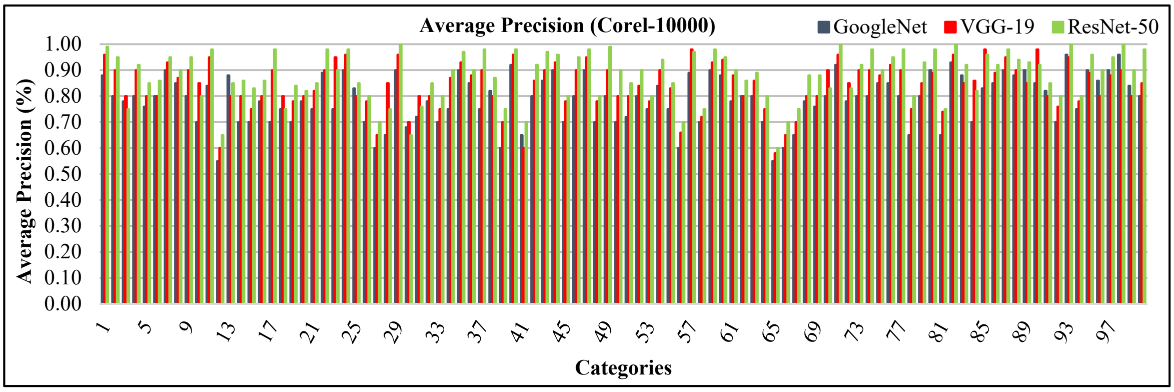


Figure 16. Average precision of corel-10000 database.

The presented approach reports mAP results in Figure 17a. The mAP precision using ResNet-50 is 88%, mAP using VGG-19 is 84% and mAP using GoogLeNet is 79% for all categories of Corel-10000 dataset. Moreover, the presented method shows more than 85% using ResNet-50, above 80% ARP rates using VGG-19 and above 75% ARP rates using GoogLeNet for most of the categories in Figure 17b. The presented approach shows significant ARP ratios of Corel-10000 dataset using L2 normalization, RGB color coefficients to efficiently and effectively image classification and indexing.

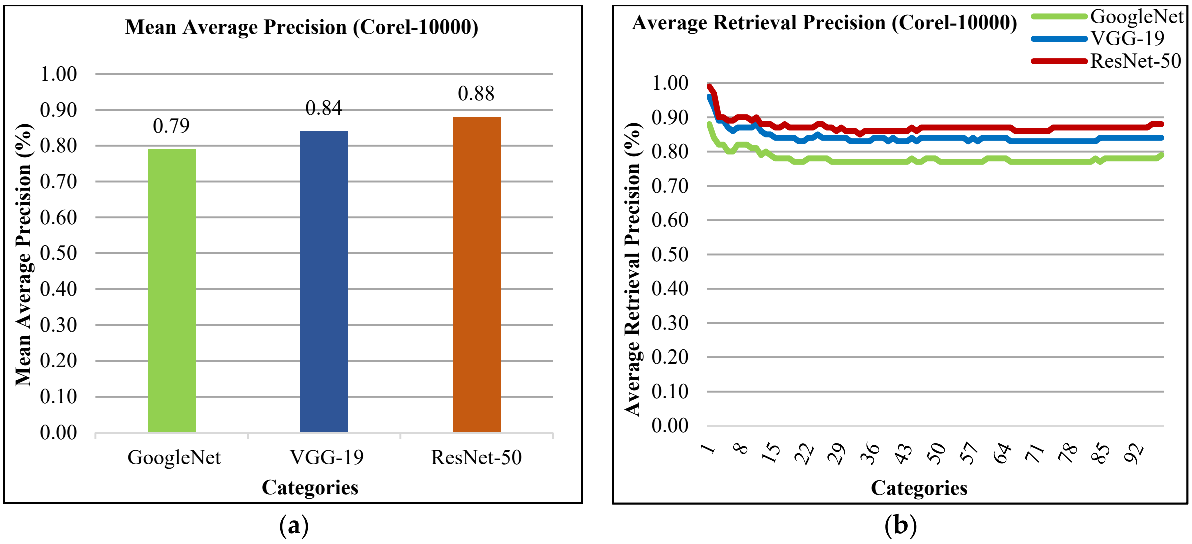


Figure 17. (**a**) Mean average precision for the corel-10000 database; (**b**) Average retrieval precision for the corel-10000 database.

4.2.8. Performance on the 17-Flowers Dataset

This dataset [63] contains 80 images for each class. These flowers images select from very common flowers that are existed in UK. The image has attributes like light and pose variations. Some categories of flowers are different in shape but same in color. The samples images of 17-flowers dataset are shown in Figure 18. 17-flowers dataset consists various types of flowers are Lily Valley, Daffodils, Crocus, Snowdrop, Iris, Dandelion, Tulip, Bluebell, Pansy, Buttercup, Sunflower, Tigerlily, Windflower, Daisy, Colts' Foot, Fritillary, and Cowslip.

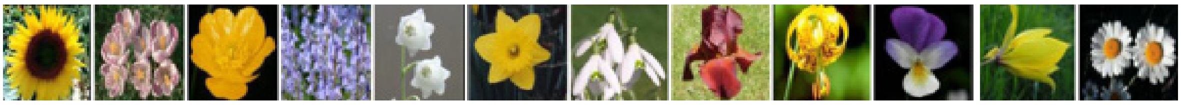


Figure 18. Various sample images for 17-flowers dataset [63].

The presented approach shows remarkable AP rates for 17-flowers dataset as shown in Figure 19a. It is observed that the presented technique reports better results for most images of 17-flowers classes with various color, shape, and texture. The scale spacing, spatial mapping, L2 norm, and RGB coefficients with CNN features make it possible to affectively classify images of flowers. The presented technique reports significant average precision ratios using ResNet-50 in flower categories including Bluebell, Cowslip, Colts' Foot, Daisy, Crocus, Dandelion, Tiger lily, Tulip, Fritillary, Sunflower, Lily valley, and Windflower. The versatility and superiority of the presented approach is to differentiate object depended on their color and texture which plays main role in flowers classification. Furthermore, the presented approach shows amazingly highest rates using ResNet-50 in 12/17 flower classes. The proposed technique reports high AP ratios using VGG-19 in Buttercup, Iris, and Snowdrop flower categories. It is noticed that the presented approach shows high AP ratio using GoogLeNet in Daffodils and Pansy flower categories. Sunflower and fritillary categories are similar in size and shape with different color. In Figure 19b, the presented approach shows f-measure results for 17-flowers. The proposed method shows f-measures between 18% and 30% for all image categories of 17-flowers dataset.

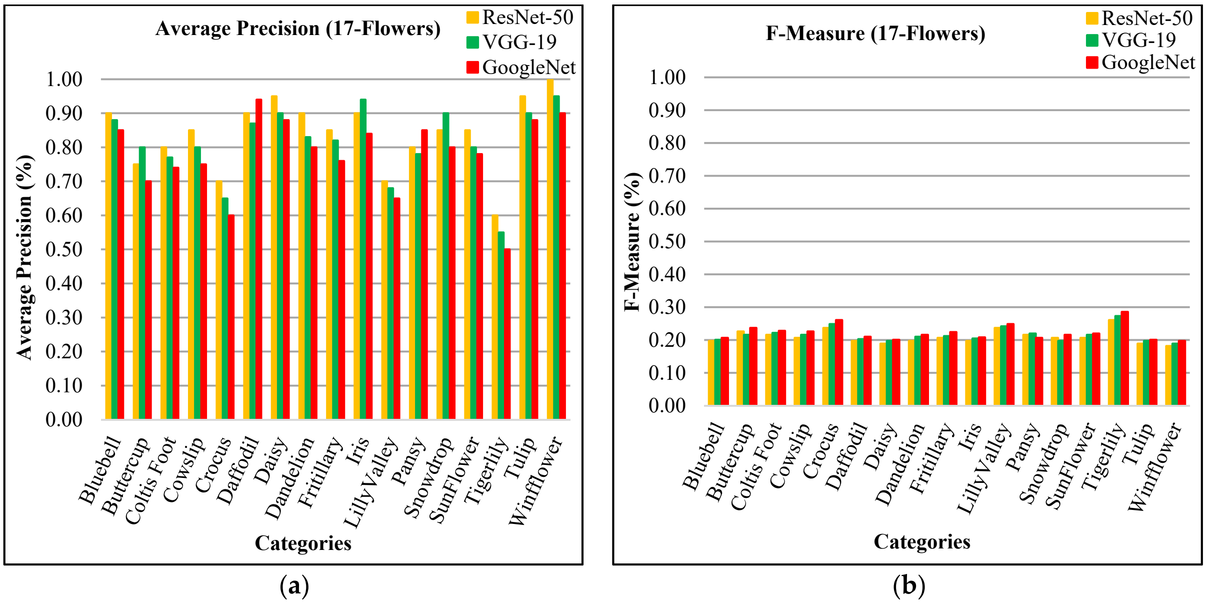


Figure 19. (**a**) Average precision for 17-flowers dataset; (**b**) F-measure results for 17-flowers dataset.

The presented approach shows 84% mAP using ResNet-50, 81% using VGG-19, and 77% using GoogLeNet for 17-flowers dataset in Figure 20a. In Figure 20b, the presented approach reports highest ARP rates using ResNet-50 for most of the categories which reports the significant performance of the proposed method for 17-flowers dataset. The presented approach presents above 80% ARP rates by using features extracted from ResNet-50, above 75% ARP using VGG-19, and 70% ARP rates using GoogLeNet.

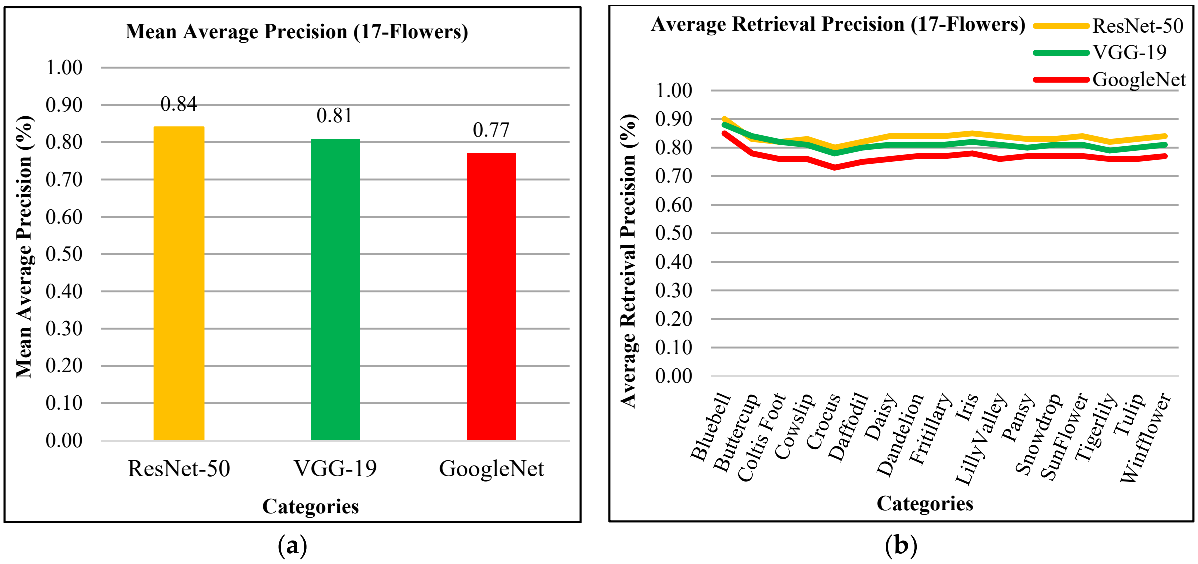


Figure 20. (**a**) Mean average precision for 17-flowesr dataset; (**b**) Average retrieval precision for 17-flowers dataset.

4.2.9. Performance on the FTVL Tropical Fruits Dataset

The FTVL tropical fruits database [63] consists 2612 images of 15 various types of fruits including Tahiti Lime, Fuji Apple, Cashew, Diamond Peach, Granny Smith Apple, Asterix Potato, Nectarine, Watermelon, Honeydew Melon, Agata Potato, Spanish Pear, Plum, Kiwi, Onion, and Orange. Sample images for FTVL Tropical Fruits dataset are shown in Figure 21.

Figure 21. Various sample images from FTVL Tropical Fruits dataset [63].

In Table 4, the presented approach shows remarkable AP, f-measure and ARP rates of the FTVL tropical fruits database. It is noted that presented technique shows highest results for most of the images of FTVL tropical fruits categories with similar shape and color. The spatial mapping, L2 norm, color coefficients, and multilevel scaling with CNN features make it possible to effectively classify images of tropical fruits. The presented technique reports high precision rates using ResNet-50 in tropical fruits categories including Tahiti Lime, Cashew, Agata Potato, Diamond Peach, Astrix Potato, Nectarine, Fuji Apple, Watermelon, Plum, Onion, and Orange. The superiority of the presented technique is to differentiate objects that are based on color, shape and texture which plays important role for classification of tropical fruits. However, the presented method shows amazingly high rates using ResNet-50 in 12/15 tropical fruits classes. The proposed approach reports highest AP rate using VGG-19 in Spanish Pear and Granny Smith Apple categories of tropical fruits. It is observed that the presented approach shows significant AP ratio using GoogLeNet in Honeydew Melon flower category. The presented approach shows outstanding f-measure results in Table 4 of FTVL Tropical Fruits database. However, the presented approach provides outstanding f-measure results using ResNet-50, VGG-19 and GoogLeNet. Moreover, the presented approach reports highest ARP rates using ResNet-50 for most of the categories which reports significant performance of the proposed method for tropical fruits dataset. The presented method presents above 90% ARP rates by using features extracted from ResNet-50, above 85% ARP using VGG-19 and 80% ARP rates using GoogLeNet. The proposed approach shows 92% mAP using ResNet-50, 88% using VGG-19, and 82% using GoogLeNet for tropical fruits dataset.

Table 4. Average precision, F-measure, and ARP ratios of FTVL Tropical Fruits dataset.

	FTVL Tropical Fruits Dataset (Average Precision, F-Measure, and ARP)								
Category		**ResNet-50**			**VGG-19**			**GoogLeNet**	
Tahiti Lime	1.00	0.18	0.90	0.86	0.20	0.86	0.80	0.22	0.80
Cashew	0.95	0.19	0.98	0.90	0.20	0.88	0.87	0.20	0.84
Agata Potato	1.00	0.18	0.98	0.92	0.19	0.89	0.90	0.20	0.86
Diamond Peach	0.95	0.19	0.98	0.89	0.20	0.89	0.82	0.21	0.85
Granny Smith Apple	0.95	0.19	0.97	0.90	0.20	0.89	0.75	0.23	0.83
Asterix Potato	0.90	0.20	0.96	0.88	0.20	0.89	0.70	0.24	0.81
Nectarine	1.00	0.18	0.96	0.90	0.20	0.89	0.86	0.20	0.81
Fuji Apple	0.85	0.21	0.95	0.78	0.22	0.88	0.70	0.24	0.80
Watermelon	1.00	0.18	0.96	0.98	0.18	0.89	0.90	0.20	0.81
Honeydew Melon	0.90	0.20	0.95	0.77	0.22	0.88	0.82	0.21	0.81
Spanish Pear	1.00	0.18	0.95	0.98	0.18	0.89	0.88	0.20	0.82
Plum	0.90	0.20	0.95	0.88	0.20	0.89	0.80	0.22	0.82
Kiwi	0.98	0.20	0.95	0.78	0.22	0.88	0.76	0.22	0.81
Onion	1.00	0.18	0.95	0.90	0.20	0.88	0.88	0.20	0.82
Orange	0.95	0.19	0.95	0.88	0.20	0.88	0.80	0.22	0.82

4.2.10. Performance on the Caltech-256 Dataset

The Caltech-256 database [71] comprises of more than thirty thousand images which are allocated to 257 varied categories. Fifteen different image categories are used to perform experimentation including bonsai, wrist watch, back-pack, teddy-bear, cactus, airplane, boxing gloves, teapot, spider, billiards, swan, tomato, bulldozer, tree, and butterfly. All image categories in database are essential due to their texture patterns and background and foreground objects. Sample images of Caltech-256 dataset are shown in Figure 22.

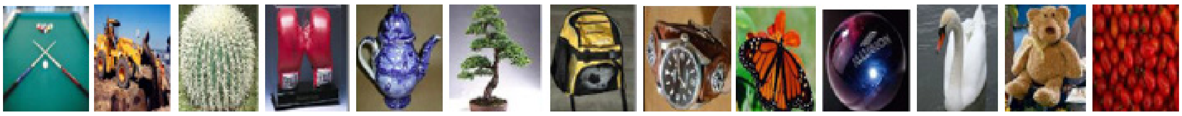
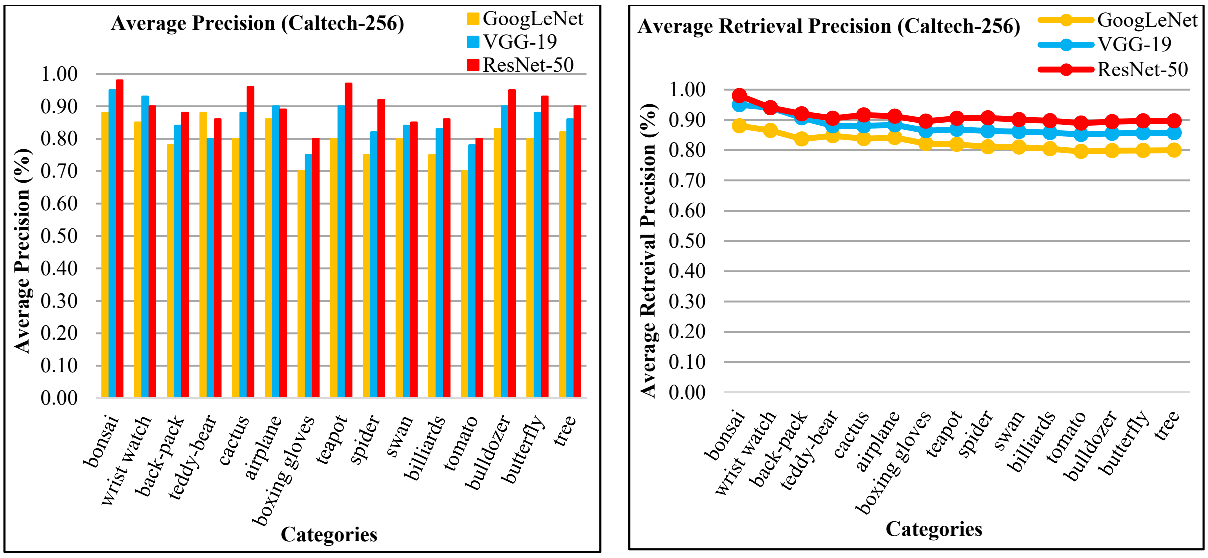


Figure 22. Various sample images of Caltech-256 dataset [71].

The AP rates of Caltech-256 database is shown in Figure 23a. The presented approach provides the highest results for most of the images of Caltech-256 categories with similar shape and color. The spatial mapping, L2 norm, RGB coefficients, and scale spacing with CNN features make it possible to effectively classify images of Caltech-256. It is noted that presented method reports high precision rates using ResNet-50 in tropical fruits for most of the categories. The superiority of the presented technique is to differentiate objects that are based on color, shape, and texture which plays important role for classification of Caltech-256. However, the presented approach shows amazingly high rates using ResNet-50 in 13/15 Caltech-256. The proposed approach reports highest AP ratios using VGG-19 in wrist watch and airplane categories of Caltech-256. It is observed that the presented technique shows significant AP ratio using GoogLeNet in teddy bear category. The presented approach provides outstanding recall rates for all categories of Caltech-256. Moreover, the presented approach reports 90% mAP using ResNet-50, 86% using VGG-19, and 80% using GoogLeNet. In Figure 23b, the presented method shows above 89% ARP using ResNet-50, above 82% ARP rates using VGG-19, and above 75% ARP rates using GoogLeNet for many categories. The presented approach shows outstanding ARP ratios for

Caltech-256 dataset using L2 norm, color coefficient, and multi-scale filtering to effectively and efficiently image classification and indexing.



(**a**) (**b**)

Figure 23. (**a**) Average precision for Caltech-256 dataset; (**b**) Average retrieval precision of Caltech-256 database.

4.2.11. Results of the FTVL Tropical Fruits, Cifar-10, Corel-1000, and 17-Flowers Datasets with State-of-the-Art Methods

To check the accuracy and efficiency of the presented method, it compares with state-of-the-art methods. The comparison is performed with existing methods provided outstanding performance.

The FTVL fruit dataset [63] is used for experimentations because of its cropped object, illumination differences, pose variations, and partial occlusions; and Figure 24 depicts AP rates for FTVL tropical fruits dataset, the proposed method is compared with existing methods including CBRFF [63] and with existing research techniques in [78]. The presented method shows significant average precision results in most of the image categories of FTVL tropical fruits dataset.

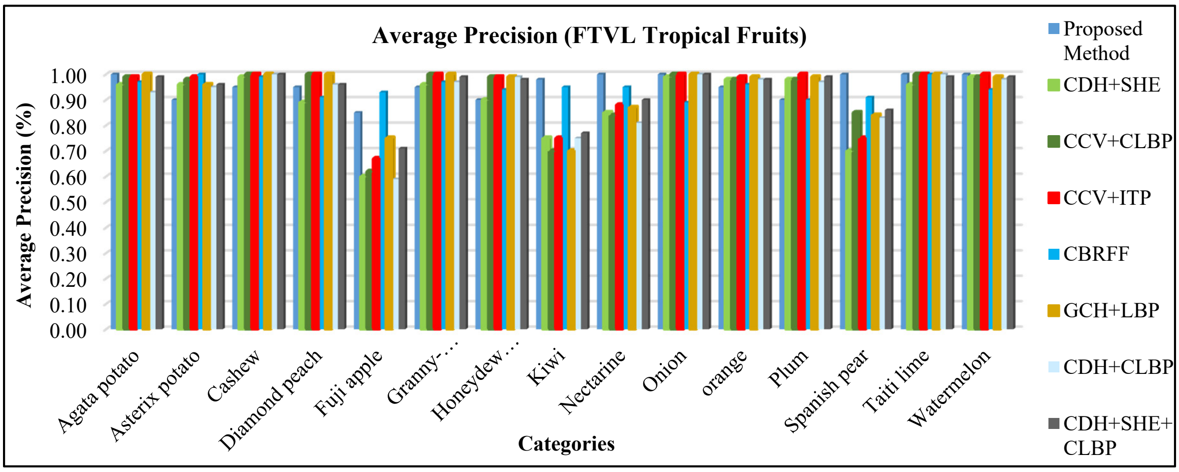


Figure 24. AP results of presented approach versus existing research methods for FTVL tropical fruits dataset.

The AP ratios of the proposed approach technique in assessment with state-of-the-art methods are presented in Table 5. Some techniques CDH +SEH illustrate low accuracy rate due to the missing nature for their technique in the cropped objects. Some methods incorporate the textural attribute and provide AP but with mixed, ambiguous images. However, the objects with the same color and shape are still hard to recognize for them. The presented method is takes color coordinates into consideration using shape and textural properties to report 0.96 mAP.

Table 5. Comparison of presented method mAP results with CBRFF [63] and other state of the art methods [78] of FTVL Tropical Fruits dataset.

FTVL Tropical Fruits Dataset Vs. Existing Methods–Mean Average Precision (mAP)								
	Proposed Method	**CDH + SEH**	**CCV + CLBP**	**CCV + LTP**	**GCH + LBP**	**CDH + CLBP**	**CDH + SHE + CLBP**	**CBRFF [63]**
Average	0.96	0.90	0.93	0.93	0.94	0.91	0.94	0.95

The mean average precision results are graphically shown in comparison with existing research methods for cifar-10 dataset of the presented approach and other existing research approaches are shown in Figure 25. The proposed approach shows remarkable mAP results over state of the art methods for cifar-10 dataset. The existing methods including ETRCI [76], IRSCTS [79], FFCDIR [80], MSCBIR [81], CBIRCT [82], SPRCNN [83], and PBOMLA [84]. The mean average precision of the presented approach as compared with existing research methods is graphically presented in Figure 25.

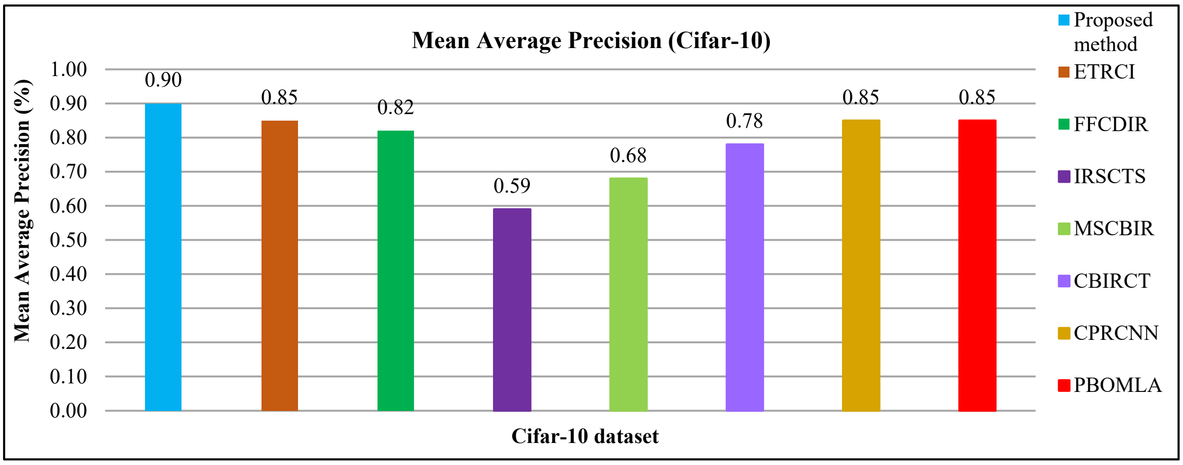


Figure 25. AP results of presented approach versus existing research methods for Cifar-10 dataset.

17-Flowers database is used for shape, texture and color features experimentation. The comparative results of the proposed approach with existing research methods for mean average precision metric are shown in Table 6. The fine-grained [85] technique provides improved ratios by extracting deeper shape and color information. The research method in [86] performs the spatial matching and calculates the differences depending upon the algorithmic criteria and results in low precision due to colors and shapes. The other existing research methods [86–88] include linear coding and return mean average precision results. This approach is not capable to perform deep analysis on shape and texture features. The proposed approach returns outstanding AP in most of the flower categories using spatial texture and color patterns along with shape details. The mean average precision results for the proposed approach versus state-of-the-art approaches in Table 6.

Table 6. Comparison of presented method mAP results versus state of the art methods of 17-Flowers dataset.

17-Flowers Dataset vs. Existing Methods—Mean Average Precision (mAP)									
Average	Proposed Method	LCDS [85]	SCSPM [86]	SCS W/B [86]	LCLCIC [87]	JLVCD [88]	JLVCD W/B [88]	FCMOC [89]	DLCSPC [90]
	0.84	0.72	0.52	0.62	0.65	0.69	0.71	0.67	0.59

Experimentation is performed to check the accuracy and effectiveness of the presented approach, average precision results for Corel-1000 dataset are shown in comparison with state-of-the-art research approaches. The challenging research methods used for the comparison are ETRCI [76], DISR [73], CBIF [63], FFECIR [91], MDLBP [75], MNSIR [81], ENNSR [92], CRHOG [93], MCFGM [94], and EIRCTS [79]. In Figure 26, a graphical illustration of AP rates for the presented approach as compared with state-of-the-art research methods. The presented technique shows the highest AP results performance in most of the image categories of Corel-1000 dataset. The proposed approach reports better AP rates in the categories—such as buildings, flowers, mountains, and food.

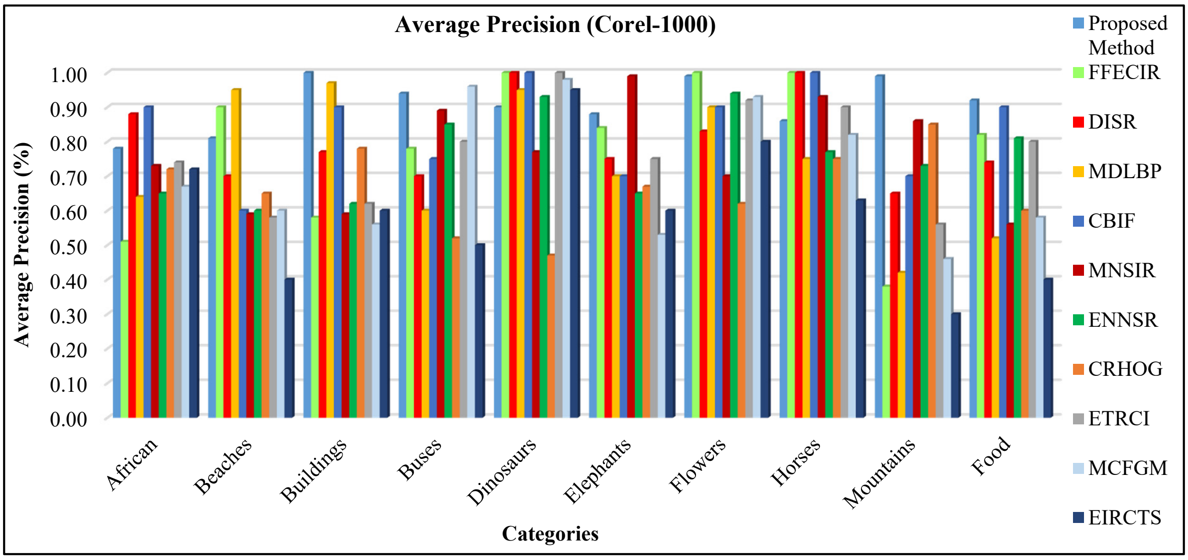


Figure 26. AP results of presented approach versus existing research methods for Corel-1000 dataset.

Figure 27 shows mAP results of the presented approach in comparison with state-of-the-art research methods. The proposed approach reports highest 0.91 mAP. CBIF [63] shows second highest 0.84 mAP. MNSIR [81], ENNSR [92] report 0.76 mAP. ETRCI [76], DISR [73], FFECIR [91], MDLBP [75], CRHOG [93], and MCFGM [94] show mean average precision ratios between 0.66 and 0.80. EIRCTS [79] provides the lowest 0.59 mAP. The EIRCTS [75] covers the textural aspects therefore mAP rates are comparatively lower than the other approaches.

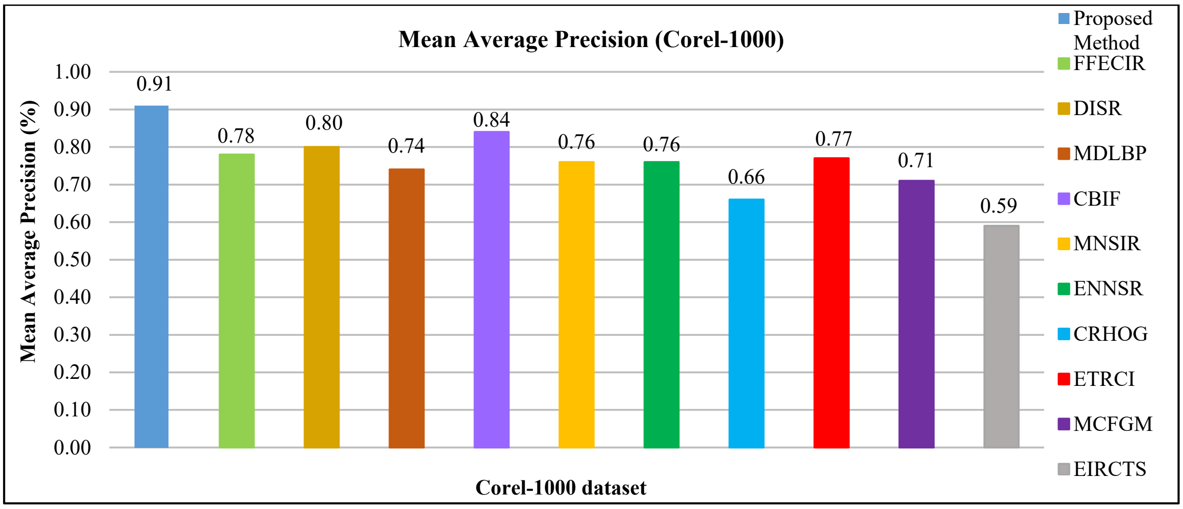


Figure 27. mAP results of presented approach versus existing research methods for Corel-1000 dataset.

5. Conclusions

This research presents a novel interactional fusion of GoogLeNet, VGG-19, and ResNet-50 with an innovative salient anchor collection and detection framework to enhance the image retrieval accuracy over large datasets. This innovative deep learning solution is a breakthrough in framework capsulation, features binding, primitive features fusion with layered technology, deep features orientation, and CNN typed effects on revealed local and global signatures. The remarkable outcomes with extensive experimentation from benchmark architectures endorsed the superiority of the presented approach with ALOT (250), Corel-10000, Cifar-10, Oxford Buildings, FTVL Tropical Fruits, 17-Flowers, Cifar-100, Fashion (15), Corel-1000, and Caltech-256 highly recognizable datasets. An extension to this research is the fusion of several formulations.

Author Contributions: Conceptualization, K.K. and K.T.A.; Methodology, K.K.; Writing—Original draft preparation, K.K.; Investigation, K.K.; Data curation, K.K., R.K. and N.A.; Formal analysis, K.K.; validation, K.K., L.J. and K.T.A.; resources, L.J.; visualization, K.K., K.T.A. and L.J.; Software, K.K.; supervision, L.J.; experimentation, K.K.; investigation, K.K.; implementation, K.K. and K.T.A.; writing-review and editing, L.J. and K.T.A.; Project administration, L.J.; funding acquisition, L.J. All authors have read and agreed to the published version of the manuscript.

Funding: This paper is supported by the Chinese Academy of Sciences through the Strategic Priority Research Program (A Class) under Grant XDA19020102.

Conflicts of Interest: There no conflict of interest.

References

1. Zhu, X.; Bain, M. B-CNN: Branch convolutional neural network for hierarchical classification. *arXiv* **2017**, arXiv:1709.09890.
2. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–8 December 2012.
3. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
4. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015.
5. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
6. Chetlur, S.; Wool, C.; Vandermersch, P.; Cohen, J.; Tran, J.; Catanzaro, B.; Shelhamer, E. cuDNN: Efficient primitives for deep learning. *arXiv* **2014**, arXiv:1410.0759.
7. Wang, Y.; Wei, G.-Y.; Brooks, D. A systematic methodology for analysis of deep learning hardware and software platforms. In Proceedings of the Machine Learning and Systems, Austin, TX, USA, 2–4 March 2020; pp. 30–43.

8. Canziani, A.; Paszke, A.; Culurciello, E. An analysis of deep neural network models for practical applications. *arXiv* **2016**, arXiv:1605.07678.
9. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef]
10. Borji, A.; Cheng, M.; Jiang, H.; Li, J. alient object detection: A benchmark. *IEEE Trans. Image Process.* **2015**, *24*, 5706–5722. [CrossRef]
11. Abdel-Hamid, O.; Mohamed, A.; Jiang, H.; Deng, L.; Penn, G.; Yu, D. Convolutional neural networks for speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1533–1545. [CrossRef]
12. Johnson, R.; Zhang, T. Semi-supervised convolutional neural networks for text categorization via region embedding. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 919–927.
13. Zhou, W.; Li, H.; Tian, Q. Recent advance in content-based image retrieval: A literature survey. *arXiv* **2017**, arXiv:1706.06064.
14. Kim, Y. Convolutional neural networks for sentence classification. *arXiv* **2014**, arXiv:1408.5882.
15. Gudivada, V.N.; Raghavan, V.V. Content based image retrieval systems. *Computer* **1995**, *28*, 18–22. [CrossRef]
16. Lowe, D.G. Object recognition from local scale-invariant features. In Proceedings of the 7th IEEE International Conference on Computer Vision, Corfu, Greece, 25 September 1999.
17. Massaoudi, M.; Bahroun, S.; Zagrouba, E. Video summarization based on local features. *arXiv* **2017**, arXiv:8086943461.
18. Abubakar, F.M. A study of region-based and contour-based image segmentation. *Signal Image Process.* **2012**, *3*, 15.
19. Friedman, N.; Russell, S. Image segmentation in video sequences: A probabilistic approach. *arXiv* **2013**, arXiv:1302.1539.
20. Kamdi, S.; Krishna, R. Image segmentation and region growing algorithm. *Int. J. Comput. Technol. Electron. Eng. (IJCTEE)* **2012**, *2*, 103–107.
21. Liu, J.; Leung, S. A splitting algorithm for image segmentation on manifolds represented by the grid based particle method. *J. Sci. Comput.* **2013**, *56*, 243–266. [CrossRef]
22. Dhanachandra, N.; Manglem, K.; Chanu, Y.J. Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. *Procedia Comput. Sci.* **2015**, *54*, 764–771. [CrossRef]
23. Amoda, N.; Kulkarni, R.K. Image segmentation and detection using watershed transform and region based image retrieval. *Int. J. Emerg. Trends Technol. Comput. Sci.* **2013**, *2*, 89–94.
24. Szczypiński, P.; Klepaczko, A.; Pazurek, M.; Daniel, P. Texture and color based image segmentation and pathology detection in capsule endoscopy videos. *Comput. Methods Programs Biomed.* **2014**, *113*, 396–411. [CrossRef]
25. Rasoulian, A.; Rohling, R.; Abolmaesumi, P. Lumbar spine segmentation using a statistical multi-vertebrae anatomical shape+pose model. *IEEE Trans. Med. Imaging* **2013**, *32*, 1890–1900. [CrossRef]
26. Tang, M.; Djelouah, A.; Perazzi, F.; Boykov, Y.; Schroers, C. Normalized cut loss for weakly-supervised cnn segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
27. Kadir, A.; Nugroho, L.; Susanto, A.; Santosa, P. Leaf classification using shape, color, and texture features. *arXiv* **2013**, arXiv:1401.4447.
28. Lin, S.; Crotty, K.M.; Vazquez, N. Shape Feature Extraction and Classification. U.S. Patent 7,668,376 B2, 23 February 2010.
29. Riaz, F.; Hassan, A.; Rehman, S.; Qamar, U. Texture classification using rotation-and scale-invariant gabor texture features. *IEEE Signal Process. Lett.* **2013**, *20*, 607–610. [CrossRef]
30. Rippel, O.; Snoek, J.; Adams, R.P. Spectral representations for convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 11 June 2015.
31. Szegedy, C.; Zaremba, W.; Sutskever, L.; Bruna, J.; Erhan, D.; Goodfellow, L.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
32. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, L.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
33. Kuo, C.-H.; Chou, Y.-H.; Chang, P.-C. Using deep convolutional neural networks for image retrieval. *Electron. Imaging* **2016**, *2016*, 1–6. [CrossRef]
34. Chan, T.-H.; Jia, K.; Gao, S.; Lu, J.; Zeng, Z.; Ma, Y. CANet: A simple deep learning baseline for image classification? *IEEE Trans. Image Process.* **2015**, *24*, 5017–5032. [CrossRef] [PubMed]
35. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing for image retrieval via image representation learning. In Proceedings of the Twenty-eighth AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014.
36. Lin, K.; Yang, H.; Hsiao, J.; Chen, C. Deep learning of binary hash codes for fast image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015.
37. Gianfelici, F.; Biagetti, G.; Crippa, P.; Turchetti, C. Novel KLT algorithm optimized for small signal sets [speech processing applications]. In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05), Philadelphia, PA, USA, 18–23 March 2005.
38. Kanwal, K.; Ahmad, K.; Khan, R.; Abbasi, A. Deep Learning Using Symmetry, FAST Scores, Shape-Based Filtering and Spatial Mapping Integrated with CNN for Large Scale Image Retrieval. *Symmetry* **2020**, *12*, 612. [CrossRef]
39. Lavinia, Y.; Vo, H.H.; Verma, A. Fusion based deep CNN for improved large-scale image action recognition. In Proceedings of the 2016 IEEE International Symposium on Multimedia (ISM), San Jose, CA, USA, 11–13 December 2016.

40. Abdallah, F.B.; Feki, G.; Ammar, A.; Amar, C. Multilevel deep learning-based processing for lifelog image retrieval enhancement. In Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 7–10 October 2018.
41. Yamamoto, S.; Nishimura, T.; Akagi, Y.; Takimoto, Y.; Inoue, T.; Toda, H. Pbg at the ntcir-13 lifelog-2 lat, lsat, and lest tasks. In Proceedings of the NTCIR-13, Tokyo, Japan, 5–8 December 2017.
42. Sapijaszko, G.; Mikhael, W.B. An Overview of Recent Convolutional Neural Network Algorithms for Image Recognition. In Proceedings of the 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Canada, 5–8 August 2018.
43. Mateen, M.; Wen, J.; Song, S.; Huang, Z. Fundus image classification using VGG-19 architecture with PCA and SVD. *Symmetry* **2019**, *11*, 1. [CrossRef]
44. Liu, L.; Chen, J.; Fieguth, P.; Zhao, G.; Chellappa, R.; Pietikäinen, M. From BoW to CNN: Two decades of texture representation for texture classification. *Int. J. Comput. Vis.* **2019**, *127*, 74–109. [CrossRef]
45. Srinivas, S.; Sarvadevabhatla, R.; Mopuri, K.; Prabhu, N.; Kruthiventi, S.; Babu, R. A taxonomy of deep convolutional neural nets for computer vision. *Front. Robot. AI* **2016**, *2*, 36. [CrossRef]
46. Varga, D.; Szirányi, T. Fast content-based image retrieval using convolutional neural network and hash function. In Proceedings of the 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Budapest, Hungary, 9–12 October 2016.
47. Sezavar, A.; Farsi, H.; Mohamadzadeh, S. Content-based image retrieval by combining convolutional neural networks and sparse representation. *Multimed. Tools Appl.* **2019**, *78*, 20895–20912. [CrossRef]
48. Burghouts, G.J.; Geusebroek, J.-M. Material-specific adaptation of color invariant features. *Pattern Recognit. Lett.* **2009**, *30*, 306–313. [CrossRef]
49. Guo, Z.; Zhang, L.; Zhang, D. Rotation invariant texture classification using binary filter response pattern (BFRP). In Proceedings of the International Conference on Computer Analysis of Images and Patterns, Munster, Germany, 2–4 September 2009.
50. Varma, M.; Zisserman, A. Classifying images of materials: Achieving viewpoint and illumination independence. In Proceedings of the European Conference on Computer Vision, Copenhagen, Denmark, 28–31 May 2002.
51. Varma, M.; Zisserman, A. Texture classification: Are filter banks necessary? In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 18–20 June 2003.
52. Basu, M. Gaussian-based edge-detection methods-a survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2002**, *32*, 252–260. [CrossRef]
53. Mainali, P.; Yang, Q.; Lafruit, G.; Van Gool, L.; Lauwereins, R. Robust low complexity corner detector. *IEEE Trans.Circuits Syst. Video Technol.* **2011**, *21*, 435–445. [CrossRef]
54. Deriche, R. Recursively Implementating the Gaussian and Its Derivatives. Ph.D. Thesis, Institut National de Recherche en Informatique et en Automatique (INRIA), Le Chesnay-Rocquencourt, France, 1993.
55. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vision Image Underst.* **2008**, *110*, 346–359. [CrossRef]
56. Tomasi, C.; Kanade, T. *Detection and Tracking of Point Features*; CMU-CS-91-132; School of Computer Science—Carnegie Mellon University: Pittsburgh, PA, USA, 1991.
57. Birchfield, S. KLT: An Implementation of the Kanade-Lucas-Tomasi Feature Tracker. 2007. Available online: http://www.ces.clemson.edu/~{}stb/klt/ (accessed on 2 December 2020).
58. Sinha, S.N.; Frahm, J.; Pollefeys, M.; Genc, Y. GPU-based video feature tracking and matching. In Proceedings of the EDGE, Workshop on Edge Computing Using New Commodity Architectures, Chapel Hill, NC, USA, 23–26 May 2006.
59. Harris, C.; Stephens, M. A combined corner and edge detection. In Proceedings of the Fourth Alvey Vision Conference, Manchester, UK, 31 August–2 September 1988.
60. Mainali, P.; Yang, Q.; Lafruit, G.; Lauwereins, R.; Van Gool, L. Lococo: Low complexity corner detector. In Proceedings of the 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, USA, 14–19 March 2010.
61. Koenderink, J.J. The structure of images. *Biol. Cybern.* **1984**, *50*, 363–370. [CrossRef]
62. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
63. Ahmed, K.T.; Ummesafi, S.; Iqbal, A. Content based image retrieval using image features information fusion. *Inf. Fusion* **2019**, *51*, 76–99. [CrossRef]
64. Biagetti, G.; Crippa, P.; Falaschetti, L.; Orcioni, S.; Turchetti, C. Multivariate direction scoring for dimensionality reduction in classification problems. In Proceedings of the International Conference on Intelligent Decision Technologies, Puerto de la Cruz, Spain, 15–17 June 2016.
65. Calonder, M.; Lepetit, V.; Strecha, C.; Fua, P. Brief: Binary robust independent elementary features. In Proceedings of the European Conference on Computer Vision, Crete, Greece, 5–11 September 2010.
66. Arora, S.; Bhaskara, A.; Ge, R.; Ma, T. Provable bounds for learning some deep representations. In Proceedings of the International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
67. Zhang, J.; Yu, J.; Tao, D. Local deep-feature alignment for unsupervised dimension reduction. *IEEE Trans. Image Process.* **2018**, *27*, 2420–2432. [CrossRef]

68. Krizhevsky, A.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. *Comput. Sci.* Available online: https://www.semanticscholar.org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/5d90f06bb70a0a3dced62413346235c02b1aa086 (accessed on 2 December 2020).
69. Rostamzadeh, N.; Hosseini, S.; Boquet, T.; Stokowiec, W.; Zhang, Y.; Jauvin, C.; Pal, C. Fashion-gen: The generative fashion dataset and challenge. *arXiv* **2018**, arXiv:1806.08317.
70. Philbin, J.; Sivic, J.; Zisserman, A. Geometric latent dirichlet allocation on a matching graph for large-scale image datasets. *Int. J. Comput. Vis.* **2011**, *95*, 138–153. [CrossRef]
71. Griffin, G.; Holub, A.; Perona, P. *Caltech-256 Object Category Dataset*; 7694; 19 April 2007. Available online: https://authors.library.caltech.edu/7694/ (accessed on 11 January 2021).
72. Ahmed, K.T.; Irtaza, A.; Iqbal, M.A. Fusion of local and global features for effective image extraction. *Appl. Intell.* **2017**, *47*, 526–543. [CrossRef]
73. Ahmed, K.T.; Ummesafi, S.; Iqbal, A. Deep Image Sensing and Retrieval Using Suppression, Scale Spacing and Division, Interpolation and Spatial Color Coordinates With Bag of Words for Large and Complex Datasets. *IEEE Access.* **2020**, *8*, 90351–90379. [CrossRef]
74. Kandefer, M.; Shapiro, S. An F-measure for context-based information retrieval. *Commonsense* **2009**, 79–84.
75. Dubey, S.R.; Singh, S.K.; Singh, R.K. Multichannel decoded local binary patterns for content-based image retrieval. *IEEE Trans. Image Process.* **2016**, *25*, 4018–4032. [CrossRef] [PubMed]
76. Shrivastava, N.; Tyagi, V. An efficient technique for retrieval of color images in large databases. *Comput. Electr. Eng.* **2015**, *46*, 314–327. [CrossRef]
77. Zhou, Y.; Zeng, F.Z.; Zhao, H.M.; Murray, P.; Ren, J. Hierarchical visual perception and two-dimensional compressive sensing for effective content-based color image retrieval. *Cogn. Comput.* **2016**, *8*, 877–889. [CrossRef]
78. Dubey, S.R.; Jalal, A.S. Fruit and vegetable recognition by fusing colour and texture features of the image using machine learning. *Int. J. Appl. Pattern Recogn.* **2015**, *2*, 160–181. [CrossRef]
79. Wang, X.-Y.; Yu, Y.-J.; Yang, H.-Y. An effective image retrieval scheme using color, texture and shape features. *Comput. Stand. Interfaces* **2011**, *33*, 59–68. [CrossRef]
80. Raghuwanshi, G.; Tyagi, V. Feed-forward content based image retrieval using adaptive tetrolet transforms. *Multimed. Tools Appl.* **2018**, *77*, 23389–23410. [CrossRef]
81. ElAlami, M.E. A new matching strategy for content based image retrieval system. *Appl. Soft Comput.* **2014**, *14*, 407–418. [CrossRef]
82. Lin, C.-H.; Chen, R.-T.; Chan, Y.-K. A smart content-based image retrieval system based on color and texture feature. *Image Vis. Comput.* **2009**, *27*, 658–665. [CrossRef]
83. Zeiler, M.D.; Fergus, R. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv* **2013**, arXiv:1301.3557.
84. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012.
85. Gao, S.; Tsang, I.W.-H.; Ma, Y. Learning category-specific dictionary and shared dictionary for fine-grained image categorization. *IEEE Trans. Image Process.* **2013**, *23*, 623–634. [PubMed]
86. Yang, J.; Yu, K.; Gong, Y.; Huang, T. Linear spatial pyramid matching using sparse coding for image classification. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami Beach, FL, USA, 20–25 June 2009.
87. Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; Gong, Y. Locality-constrained linear coding for image classification. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
88. Zhou, N.; Fan, J. Jointly learning visually correlated dictionaries for large-scale visual recognition applications. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 715–730. [CrossRef]
89. Gehler, P.; Nowozin, S. On feature combination for multiclass object classification. In Proceedings of the 2009 IEEE 12th International Conference on Computer Vision, Kyoto, Japan, 27 September–4 October 2009.
90. Kong, S.; Wang, D. A dictionary learning approach for classification: Separating the particularity and the commonality. In Proceedings of the European Conference on Computer Vision, Florence, Ialy, 7–13 October 2012.
91. Walia, E.; Pal, A. Fusion framework for effective color image retrieval. *J. Vis. Commun. Image Represent.* **2014**, *25*, 1335–1348. [CrossRef]
92. Irtaza, A.; Jaffar, M.; Aleisa, E.; Choi, T. Embedding neural networks for semantic association in content based image retrieval. *Multimed. Tools Appl.* **2014**, *72*, 1911–1931. [CrossRef]
93. Pan, S.; Sun, S.; Yang, L.; Duan, F.; Guan, A. Content retrieval algorithm based on improved HOG. In Proceedings of the 2015 3rd International Conference on Applied Computing and Information Technology/2nd International Conference on Computational Science and Intelligence, Okyama, Japan, 12–16 July 2015.
94. Xiao, Y.; Wu, J.; Yuan, J. mCENTRIST: A multi-channel feature generation mechanism for scene categorization. *IEEE Trans. Image Process.* **2013**, *23*, 823–836. [CrossRef]