VFAST Transactions on Software Engineering
https://vfast.org/journals/index.php/VTSE@ 2024, ISSN(e): 2309-3978, ISSN(p): 2411-6246
Volume 12, Number 2, April-June 2024 *pp: 174-190*

Signature Elevation Using Parametric Fusion for Large Convolutional Network for Image Extraction

Khawaja Tehseen Ahmed [1], **Nida Shahid** [1], **Syed Burhan ud Din Tahir** [2], **Aiza Shabir** [1], **Muhammad Yasir Khan** [3], **Muzaffar Hameed** [1]

[1]Department of Computer Science, Bahauddin Zakariya University, Multan, Pakistan;
[2]Department of Computer Science, Air University Multan Campus, Multan, Pakistan;
[3]Department of Computer Science, MNS University of Agriculture Multan, Pakistan

Keywords: *Digital image processing. Bag-of-Words. Parametric Fusioning. Convolutional Neural Networks. Image Extraction.*

Journal Info:
Submitted:
May 15, 2024
Accepted:
June 16, 2024
Published:
June 30, 2024

Abstract

The image acquisition process involves finding regions of interest and defining feature vectors as visual features of the image. This encompasses local and global delineations for specific areas of interest, enabling the classification of images through the extraction of high-level and low-level information. The proposed approach computes the Harris determinants and Hessian matrix after converting the input image to grayscale. Blob structuring is then performed to identify potential regions of interest that can adequately describe texture, color, and shape at different representation levels and the Harris corner detector is used to identify keypoints within these regions. Moreover, scale adaptation method is applied to the determinants of the Harris matrix and the Laplacian operator to extract scale-invariant features. Meanwhile, the input image undergoes processing through VGG-19, DenseNet, and AlexNet architectures to extract features representing diverse levels of abstraction. Furthermore, the RGB channels of the input image are extracted and their color values are computed. All extracted features local, global, and color are then integrated in feature set and encoded through a bag-of-words model to rank and retrieve images based on their shared visual characteristics. The proposed technique is tested on challenging datasets including Caltech-256, Cifar-10, and Corel-1000. The presented approach shows remarkable precision, recall and f-score rates in most of the image categories. The proposed approach leverages the complementary strengths of multiple feature extraction techniques to achieve high accuracy.

***Correspondence author email address:** nidashahid697@gmail.com
DOI: 10.21015/vtse.v12i2.1810

1 Introduction

Deep learning (DL) is a leading solution for sharing huge amounts of images and data on social networks and online. Deep learning is commonly used to solve various problems such as crowded object recognition and complex object detection. The main purpose of deep learning technology based on neural networks is to generate feature vectors through feature extraction.

Analyzing and classifying images effectively requires capturing both specific details and broader visual characteristics. Visual features may describe numerous properties of either low-level features including edges, corners, and spatial relationships or high-level features including color [1], shape [2], and texture [3–5].

Global features delineate semantic similarity between images on an abstract level by concentrating on the overall properties of the image [6]. They do not impact bridging the semantic gap. Local features work at a very deep level to extract deep image features. During the extraction procedure, each pixel's properties are computed while taking its neighbors into consideration [7]. Local features play a crucial role in mitigating the semantic gap.

However, both global and local features must be merged to obtain the finest and maximum image features. Color features can be extracted using RGB channel and color value computation. The presented methodology involves a multi-faceted approach to feature extraction, encompassing both low-level and high-level information from the input images to achieve high accuracy and robustness in image classification and retrieval. The proposed approach initially converts input images into grayscale, followed by the computation of essential metrics such as the Hessian matrix and Harris determinants, serving to identify potential regions of interest within the image, laying the groundwork for subsequent feature extraction processes. Through the utilization of blob structuring techniques, regions exhibiting distinctive texture, color, and shape characteristics are identified, facilitating a nuanced understanding of the image content. Furthermore, the integration of scale adaptation methods augments the robustness of feature extraction, enabling the extraction of scale-invariant features crucial for real-world applications. Leveraging the strengths of diverse feature extraction architectures such as VGG-19, DenseNet, and AlexNet, features representing varying levels of abstraction are extracted and amalgamated into a comprehensive feature set. The most important feature for color is that it is often understood by people in the most elementary way that humans perceive the objects [5]. A detailed color analysis allows us to extract both spatial and object data within the image. Features from colors extracted from the corresponding channels of RGB in the input image are involved; therefore, the feature representation is further enriched. These processes result in a long list of features that capture the essences of the original image, such as texture, color, shape, and scale-invariance. Then, the bag-of-words model is implemented, and the combined feature set is applied for the ranking and retrieval of the image. Therefore, a high level of efficiency and effectiveness of the image classification is achieved. To assess the performance of our proposed method, an experimentation stage is conducted on challenging datasets, including Caltech-256, Cifar-10, and Corel-1000. Results exhibited the highest precision, recall, and f-score rates across each of the categories, demonstrating the efficiency and robustness of the applied method. The deep learning approach is developed using different feature extraction techniques which bring tangible results in the image classification process with a high level of accuracy and reliability.

2 Related Work

CNNs are undoubtedly best-suited to finger discrimination and parse the complex patterns, spatial relationships, and temporal embedding in the image datasets, which in return are the crucial building blocks of deep learning networks. The remarkable part of this particular CNN is the fact that by using explosion count filters, visibility of an image is enhanced as computational time is decreased [8].

Visualization operations are applied to images or feature maps that are two-dimensional representa-

tions of a grid structure. CNNs use this method of processing images or features that rely on spatial relationships and colors. The layout of pixels in an image is called a grid, and convolutional neural networks (CNNs) use this grid-based format to extract information that has semantic value [9].

CNNs undergo optimization by employing a loss function. This function guides the network to adjust its trainable parameters and reduce error through the process of backpropagation [10]. The method proposed first uses symmetric sampling [10] to obtain images from nearby keypoints and samples them based on their symmetry. After rotational sampling and comparison, the pattern obtained by pairwise rotational sampling and comparison is added to the standard deviation method to smooth the image.

Experimenting with Gaussian results using compression, box filtering, and suboptimal methods for estimating standard deviation at different scales. The resulting feature set is scaled using a smooth image with predefined parameters for scalability. The Spatial Feature Detection Algorithm [11] is introduced, which applies Non-Optimal compression for the aggregation of pixel derivative outputs to deal with angular changes within the fragment. The key points are detected by a multiscale response function (MSRD), which is the Hessian blob detector. Colors are then added to the computed shape and object profiles that were spatially defined by L2 normalization coefficients which lift each pixel.

Object-based feature vectors support the selection of higher variance coefficients that are aggregated for the construction of a bag-of-words (BoW) model, making image search and classification more precise. The scientists set out to come up with an advanced approach for extraction of information and features from images, highlighting texture patterns, color properties, and object features. This methodology selects image signatures based on the proportion of the images having various semantic groups and the spatial distribution of the high-resolution images. They provide an alternate route to image analysis taking into consideration subtleties and thus making it easier by consuming fewer resources. These feature vectors are then ingested into PCA to gain some robustness and statistics. Deep convolution neural network (CNN) utilizes pooled object features [12] and attributes which channelize condensed color thereby providing a novel signature representation. CNNs take advantage of the grid-like topology and inherent spatial dependencies in the 2-D image space, on the grounds of which spatial relationships and tone values are correlated, which eventually aids in the acceleration of analysis. The proposed methodology [13] endeavors to enhance image quality and priming the data for subsequent feature extraction processes.

Extract rotation-varying and rotation-invariant features using features obtained from feature extraction techniques such as Fourier descriptors, moment invariants, and Zernike moments. The proposed method [14] focuses on finding interest points in an image based on regions. The difference in intensity between these points of interest and their surrounding boundaries is what distinguishes them. To train and rank these components according to the detection process, uses the Union-Find algorithm to rank pixels based on their intensity values. Multi-scale analyzes were performed to capture changes in detection and circulation at different scales. The authors use the Histogram of Directional Gradients (HOG) technique for feature extraction. This method uses a Sobel kernel filter to determine the direction and magnitude of gradients in an image. Divide images into blocks and units to efficiently capture spatial information. The proposed method [15] uses the FRK algorithm and BRISK kernel, and provides a new technique for region-based keypoint detection. This method solves high-dimensional eigenvector problems by performing dimension reduction using fracturing descriptors and PCA. BoW models can also be used to effectively compare and represent images based on visual vocabulary.

3 Methodology

3.1 Grey Scale Conversion

The process of converting a color image to grayscale values involves converting each pixel in the input image. This value represents the intensity or brightness

of a pixel, regardless of the pixel's color information. Black and white are the specific shades of gray used to create the final grayscale image, black signifies the minimum intensity (lowest brightness), while white corresponds to the maximum intensity (highest brightness).

$$I = rgb2gray[16] \quad (1)$$

Color images are converted into grey levels in equation (1) to reduce the image complexity (contrast, shading, shape, edges, shading, texture, etc.) with sightseeing colors.

3.2 Hessian Matrix Computation

Hessian matrix HM, allows efficient optimization and analysis of complex systems. The HM represents important information about the spatial curvature of intensity changes and the second derivative of the image. The selection of locations is facilitated through the utilization of the Hessian matrix [17]. The second partial derivatives of a given function make up the square matrix known as the Hessian matrix. Spot-like structures are usually identified using determinant of Hessian matrix. The HM of the function f ($a_1, a_2, a_2, ...a_n$) distinct as follows.

$$\left[\frac{\frac{\partial^2 f}{\partial a2 \partial a1}\partial^2 f}{\partial a2^2} \cdots \frac{\partial^2 f}{\partial a2 \partial an}\right][\ldots][\frac{\frac{\partial^2 f}{\partial an \partial a1 \partial^2 f}}{\partial an \partial a2} \cdots \frac{\partial^2 f}{\partial an^2}] \quad (2)$$

The off-diagonal elements $\frac{\partial^2 f}{\partial a_i \partial a_j}$ contribute to the cross-axis interactions while the diagonal elements $\frac{\partial^2 f}{\partial a_i^2}$ offer information regarding the contour of the function along each axis. Through all this, the Hessian matrix computations involve the derivations of all the second-order partial derivatives. Evaluating the HM gives valuable information which leads to an understanding of how the functions behave.

Through all this, the Hessian matrix computations involve the derivations of all the second-order partial derivatives. Evaluating the HM gives valuable information which leads to an understanding of how the functions behave.

3.3 Detect Determinants

The article introduces two well-known scale-adaptation methods: the scale adaptation Harris Laplace method and scale adaptation of Hessian Laplace.

3.3.1 Harris Laplace

HL is a coalition to blob finding technique merging Gaussian Laplacian (LoG) and the Harris corner detector. The algorithm locates the brightest and darkest areas that make up the images covering various range of scales. Applying this technology, corner detection procedures will suffice as well as efficient scale space analysis which make sure the localization gives reliable and precise results. For illustration, the HL method applies to objects of any size with great skills that keep noise at bay and some degree of light variance. Scale-invariant corner detection approach that the LH detector utilizes is designed to be based on the combination of scale and Laplacian operator.

3.3.2 Hessian Laplace

The blob detection approach based on the Hessian Laplace method operates efficiently for blobs of arbitrary scale and shape utilizing smoothed of Hessian (HM) for different scales and hierarchical representation from the scale space. It provides shot scale that is used for recognizing blobs by employing the LoG operator to execute scale-space analysis and locality information of Hessian matrix (HM).

3.4 Determinants of Harris Matrix

The Harris matrix demonstrates visual scenes' spatial structure and uses 2x2 matrices to accomplish this purpose. It is one of the factors responsible for ensuring that the corners and nuclei with high intensity fluctuations within the region do not occur.

$$HM = \left[\ [M, N], \quad [N, O]\ \right] \quad (3)$$

The elements M, N, and O whose values are obtained from the spatial derivatives of the intensity image, using the local coordinate matrix, are composing the Harris matrix (HM) as in (3), where it gets its name from. The equation (4) is used to calculate the Harris matrix determinants:

$$\text{Det}(HM) = M \times O - N^2 \quad [18] \quad (4)$$

The Det (HM) provides information about the spatial image erection and is used to measure the over-

all response of the Harris matrix. Determines whether the pixel represents a corner, edge, or flat area. The second derivative of the image is evaluated using the Laplacian operator, which provides information about the edges and local intensity changes of the image. The discrete definition of the Laplacian operator is:

$$\nabla^2 g(a,b) = g(a-1,b)+g(a+1,b)+g(a,b-1)+g(a,b+1)-4g(a,b) \quad (5)$$

Here, g(a,b) denotes the pixel intensity at the (a,b) coordinate. The Laplacian operator is utilized to compute the summation of pixel intensity variations between the central pixel and its adjacent pixels. Areas with rapid changes in intensity are highlighted, often overlapping edges or other important image features. The curvature and intensity fluctuations of picture features over various scales are described by these derivatives. The Gaussian function's second derivative with respect to local coordinates is used to do this.

$$G(a,b) = \frac{1}{2\pi\sigma^2} \times \exp\left(-\frac{a^2+b^2}{2\sigma^2}\right) \quad (6)$$

Among these, exp() represents the exponential function, (a,b) represents the spatial coordinates, and represents the standard deviation$_{SD}$ of Gaussian distribution$_{GD}$. The computation of the first derivative of a Gaussian function concerning both the x and y dimensions can be achieved through the following formula.

$$\frac{\partial G(a,b)}{\partial a} = \left(-\frac{a}{\sigma^2}\right) \times G(a,b) \quad (7)$$

$$\frac{\partial G(a,b)}{\partial b} = \left(-\frac{b}{\sigma^2}\right) \times G(a,b) \quad (8)$$

Likewise, the first derivative can be used to obtain the second derivative, also known as the Mexican Hat operator or Laplacian of Gaussian (LoG) operator.

$$\frac{\partial^2 G(a,b)}{\partial a^2} = \left(\frac{a^2-\sigma^2}{\sigma^4}\right) \times G(a,b) \quad (9)$$

$$\frac{\partial^2 G(a,b)}{\partial b^2} = \left(\frac{b^2-\sigma^2}{\sigma^4}\right) \times G(a,b) \quad (10)$$

The value of the standard deviation has a significant effect on the Gaussian second derivative. This controls the scale of the Gaussian function and determines how much information the derivative function can capture. A smaller value provides more detail, but also increases noise in the image. On the contrary, elevated values facilitate a broader exploration of the image structure, albeit with the potential drawback of attenuating finer details due to increased blurring.

The RGB color model serves as a ubiquitous framework for the representation and manipulation of color information. Employing a conservative approach, this model amalgamates diverse intensities of red, green, and blue light to formulate the desired color output. Each channel within the image acquires the intensity data for the corresponding color component at every pixel coordinate. Typically, intensity values span the range from 0 to 255, where 0 denotes the minimum color presence, and 255 signifies the maximum color saturation.

3.5 Convolutional Neural Network (CNN)

CNNs stand out as a unique category of artificial neural networks (ANNs) characterized by employing uniform and sparse interactions between layers. This architecture effectively handles the spatial correlation inherent in substantial volumes of data. The Convolutional Neural Network (CNN) [19], demonstrates efficiency, precision, and features multiple convolutional layers incorporating subsampling through nonlinear neural network activations. The processing scope of each layer is constrained to a localized segment of the preceding layer. In CNN, inputs are arranged in a grid and directed to layers that are connected in series and maintain relationships between layers [20].

The input layer serves as a layer representation, delineating the dimensions of the image as (28 × 28 × 1). The first parameter indicates the image's height, the second gives the width, and the third parameter establishes the channels' sizes. The channel size information's numeric value specifies that a channel size of 1 corresponds to a grayscale image.

Conversely, RGB values aligning with a channel size of 3 indicate a color image representation. A trained network is required to reorganize the data. The net-

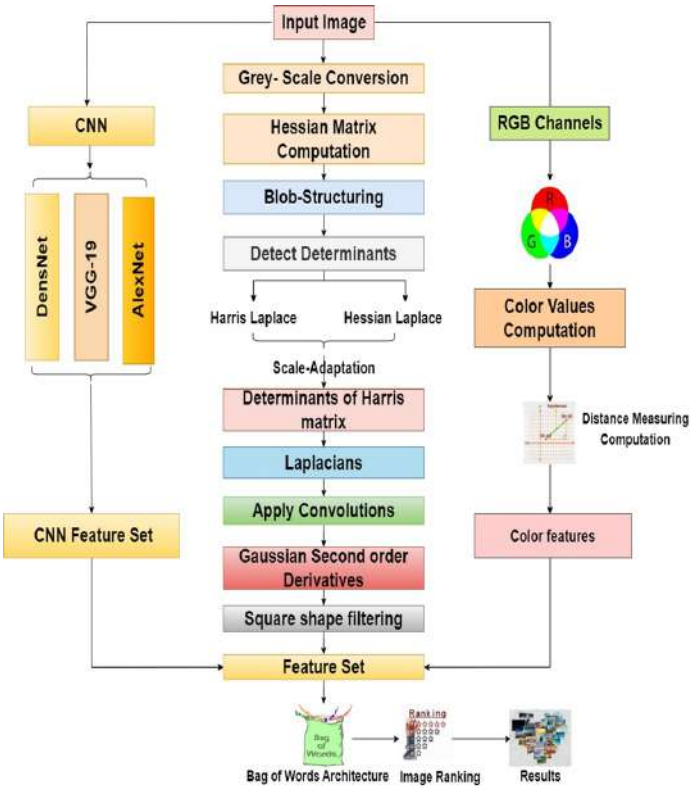


Figure 1. the proposed approach demonstrating the step-by-step image extraction process by feature extraction information

work naturally rearranges the data before training. At the beginning of each cycle, the train network can naturally reorder the data.With filterSize as the main parameter, the convolutional layer (second layer) applies multiple convolutional filters. This layer scans the image through the height and width of the applied filter. In this case, the value 3 means the filter size is 3x3.The numFilters, defines the quantity of filters employed and signifies the count of neurons associated with a specific region of the input image. The number of feature maps generated is determined by this parameter. To accurately map features to the input image, padding is added using the "Padding" parameter. Utilizing the original convolutional filter within convolution layers, with "same" padding, ensures congruence between the spatial output size and the information size.

Batch Normalization Layer normalizes the radiation and slope of the entire system and simplifies the optimization process of training the network. The ReLu layer and other nonlinear layers are separated by a BatchNormalizationLayer, accelerates the training process of the network and diminishes the likelihood of network activation. The code for this layer is generated by a successful symbolic batch normalization layer.

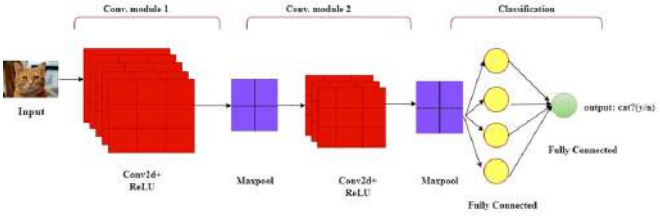


Figure 2. Convolutional Neural Networks

Batch normalization layers are observed through nonlinear activation control. Rectified Linear Units (ReLu) are the most efficient conventional drive control. The ReluLayer function generates ReLu layers. An activation function takes this input into the node's output or activates the node's weighted input. If the input is positive (+V), the optimized linear activation function produces zero, otherwise it sends the input directly. This is a universal activation function. Different types of neural networks exist. This is primarily because models that use neural networks are easier to train and generally perform better.

In the max pooling 2-D layer function, the initial parameter, pool size, denotes the quantity of maximum output values generated by processing an image rectangle as input. The application of subsampling diminishes the dimensions of the input feature grid, resulting in a loss of grid size information. Use the maxPooling2dLayer function during downsampling. In the function syntax above, the size of the square region is max pooling [2, 2]. The filter size for the training operation is determined by the "step" parameter when moving along the original length. The FullyConnectedLayer function transforms a vectorized input image of M (dimension) into a resized output image of P (dimension). At this juncture, the input data typically undergoes weight application, and bias constraints are employed to generate the output data.

The R real-valued arcs are transformed into a vector of R real-valued vectors that add to 1 by the softmax layer using an activation function. Classification probabilities are determined based on these values and used as input for classification levels. The SoftmaxLayer function is used for layer maintenance.

$$\sigma(\mathbf{c})_m = \frac{e^{c_m}}{\sum_{n=1}^{r} e^{c_n}} \tag{11}$$

In this formula: - $\sigma(\mathbf{c})_m$ denotes the m-th component of the output vector. - $\mathbf{c} = (c_1, c_2, \ldots, c_r)$ represents the input vector. - e^{c_m} is the exponential function applied to the m-th component of the input vector. - $\sum_{n=1}^{r} e^{c_n}$ is the sum of exponentials of all components of the input vector.

The variable $\mathbf{c}_m$ represents a value within the elements of an input vector, which can be either positive or negative. Each element in the given vector undergoes the general exponential function e^{c_m} resulting in an optimism level slightly above zero for negative inputs and a substantial increase for high inputs. While the probability may reside within the stationary range of (0, 1), it is not constrained within this interval. R symbolize the number of classes within the multi-classifier system.

A classification layer computes the cross-entropy loss of mutually exclusive categories to classify the problem into multiple categories. The number of classes required for this level is based on the results of the previous level. This layer calculates a loss and allocates contributions to one of the reciprocally elite categories based on the probabilities providing by the softmax activation function for respectively input.

Bag-of-words (BoW) models used to establish functional specifications and codebooks. A “feature-based histogram representation” can be used to define the BoW model. Feature descriptors allow to identify color, shape, and texture characteristics in images. Each image is then summarized by multiple spatial blocks. Patches are represented as object descriptors, which are collections of numeric vectorsFeature vectors are employed in the generation of codewords, and these codewords, in turn, contribute to the creation of codebooks. A codeword represents a collection of interconnected image patches organized into clusters within codebooks.

4 Experimentation

4.1 Datasets

In computer vision, image data sets have expanded from small amounts of complex images to large amounts of images. The process of selecting a collection of images to estimate accuracy can be a difficult process as it depends on the problem type, chosen algorithm, and application. Some datasets contain graphical representations of people, animals, and natural objects, while others contain graphical representations of structures. Some images contain different natural environments that vary in size, location, number, presence, and associated foreground and background color contrast. Some image datasets require physical properties of semantic categories for selection. Three different types of datasets are examined. The benchmark datasets such as Caltech-256 [21], Cifar-10, and Corel-1000 have been used for the experimentation. These datasets are the subject of experiments designed to improve accuracy and efficiency. Large-scale image datasets are required to test deep learning algorithms and train their ability to extract and classify images. The dataset results are displayed in numerical format.

4.1.1 Cifar-10

Cifar-10 dataset is commonly used as a benchmark, providing a huge image repository for applications like image search. The dataset encompasses images spanning diverse semantic categories such as, trucks, vehicles, frogs, dogs, horses, deer, birds, cars, and airplanes. Comprising a total of 50,000 images, it is partitioned into 10 distinct categories [14], with each category comprising 5,000 images. Each image possesses a resolution of 32 x 32 RGB pixels, as illustrated in figure 3.

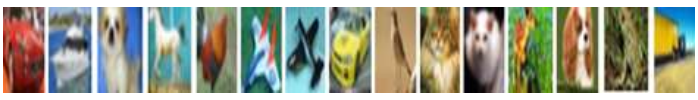


Figure 3. Cifar-10 dataset shows different sample image belongs to each category [22]

4.1.2 Corel-1000

The reference dataset Corel-1000 [23] serves as the basis for image search and classification, encompassing a diverse array of image semantic categories and content. The Corel-1000 dataset features a broad spectrum of image categories, ranging from intricate objects set against simple backgrounds. Comprising 1,000 images, this collection includes depictions of elephants, Native Americans, coastal environments, vehicles in diverse colors, various structures, assorted types of flowers, dinosaurs in different settings, ponies against varied backgrounds, mountains, and more. Each category is comprised of 100 images, with resolutions of 256 x 384 or 384 x 256 pixels, as depicted in figure 4.

Figure 4. Corel-1000 dataset shows 10 sample images per category.

4.1.3 Caltech-256

The Caltech-256 dataset, comprising 30,607 images categorized into 257 groups, stands as a sizable repository. This collection encompasses a diverse range of scenes and objects, serving as a valuable resource for experimentation and discovery. To assess the accuracy of each class within the Caltech-256 dataset, independent experiments were conducted. This allows us to comprehensively evaluate a method's performance across multiple categories and provide specific details about how well a method performs on different types of objects and images. In this experiment, 15 different image categories were used, including swan, tomato, bulldozer, tree, butterfly, airplane, boxing gloves, kettle, bonsai, clock, bag, teddy bear, spider, and pole [21]. This holds significance for the dataset since each image category exhibits a distinct texture pattern, along with unique background and foreground elements, as depicted in figure 5.

Figure 5. Caltech-256 dataset featuring 15 sample images, each representing 15 distinct categories [14]

4.2 Evaluation of Precision, Recall and F-measure

4.2.1 Precision and Recall

Precision is a widely used method that compares the total number of observations ($H_{rel(i)}$) with the corresponding or correct search results ($H_{(ret(i)}$) received at a given search threshold. A clear definition of the accuracy calculation formula is given below.

$$\text{Precision} = \frac{H_{\text{rel}(i)}}{H_{\text{ret}(i)}} \tag{12}$$

Recall is a measure of how accurately a model identifies positive events among all true positive events in a dataset. It is calculated as the true positive and the cumulative ratio of true positives to false negatives.

$$\text{Recall} = \frac{H_{\text{rel}(i)} + H_{\text{ret}(i)}}{H_{\text{rel}(i)}} \tag{13}$$

4.2.2 F-Measure

Utilizing Equation (14), an equal weighting is applied to precision (p) and recall (q) in the two metrics, resulting in the computation of the F-score as the harmonic mean of these two metrics.

$$F = \frac{2 \times p \times q}{p + q} \tag{14}$$

Equation (14) replaces p (intermediate precision) and q (completeness) with F.

4.3 Experimental Results

The proposed approach uses color images that are taken from benchmark datasets as its input. These

color images are processed by a convolutional neural network (CNN), though, interestingly, they are first converted to greyscale by the system. The efficiency and accuracy of the image retrieval system are significantly influenced by the careful selection of appropriate image datasets.

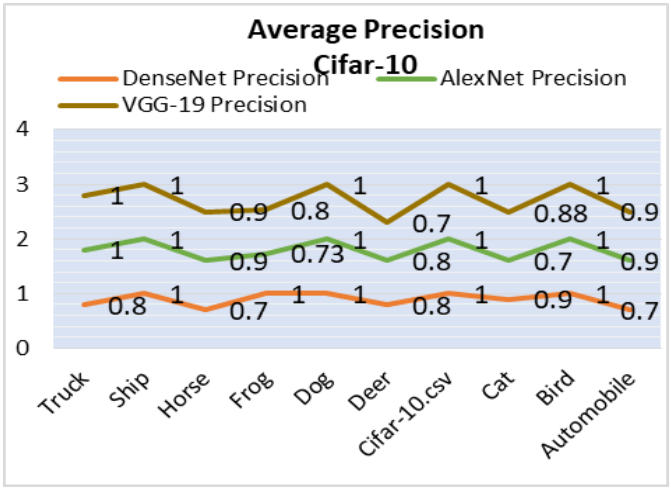
Therefore, a thoughtful and deliberate choice is crucial. Many datasets are customized to fit the needs of particular tasks and the nature of the endeavor [24, 25]. Depending on the domain they are analyzing, researchers frequently choose specific image classifications. The dependability of the results is closely related to aspects of the images, including scale, overlap, occlusion, cluttering, object placement, color composition, and consistency. We performed experimentation on three distinct benchmark datasets, namely Cifar-10, Corel-1000, and Caltech-256, to assess the accuracy of the proposed technique.

4.3.1 Experimentation on large benchmark Cifar-10

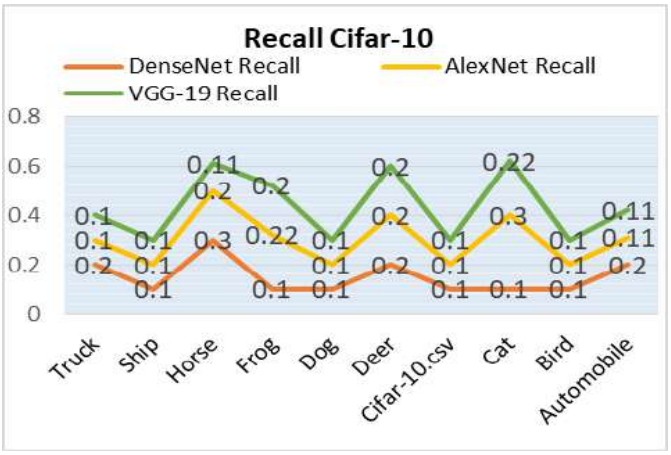
We tested our proposed method on large benchmark dataset cifar-10. Many semantic groups are represented in the cifar-10 dataset, including vehicle, truck, automobile, ship, frog, cat, dog, horse, deer, etc. This dataset contains 5,000 images per category. The proposed method demonstrates elevated average precision (AP) across a majority of the Cifar-10 categories.

The proposed method by applying CNN features achieves accurate classification of images. Precise image classification across diverse semantic categories including cats, cars, ships, dogs, horses, birds, deer, frogs, airplanes, and trucks, is achieved through the utilization of Gaussian second derivatives, RGB color channels, and advanced deep learning capabilities employing DenseNet, VGG-19, and AlexNet models [26]. On the cifar-10 dataset, the proposed approach achieves an average precision of more than 95%. Figure 6 shows a graphical representation of the proposed approach on the cifar-10 dataset.

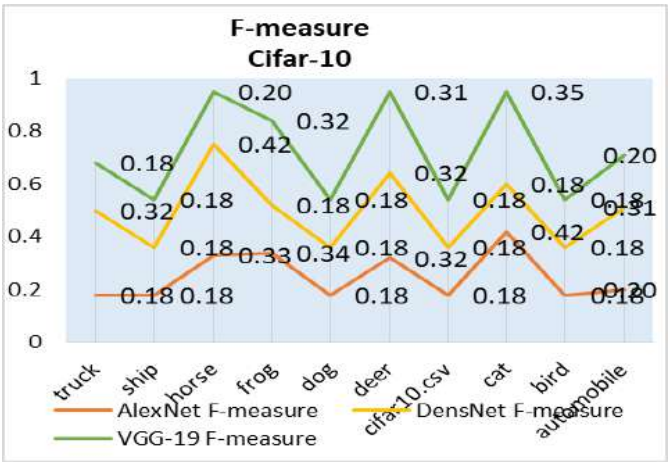
The proposed method covers all ten categories of the cifar-10 dataset. The proposed method achieved 100% precision in certain categories, including bird,



(a) **Average Precision**



(b) **Recall**



(c) **F-Score**

Figure 6. (a) shows the Average Precision (AP), **(b)** shows Recall, **(c)** shows f-score of Cifar-10 dataset with DenseNet, AlexNet, and VGG-19 architecture

Table 1 AP, Recall and F-score with DenseNet, AlexNet, and VGG-19 of Cifar-10 Dataset

Cifar-10 with DenseNet, AlexNet, and VGG-19									
	DenseNet			AlexNet			VGG-19		
Category	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Truck	0.8	**0.2**	**0.32**	**1**	0.1	0.18	**1**	0.1	0.18
Ship	**1**	0.1	0.18	**1**	0.1	0.18	**1**	0.1	0.18
Horse	0.7	**0.3**	**0.42**	**0.9**	0.2	0.33	**0.9**	0.11	0.20
Frog	**1**	0.1	0.18	0.73	0.22	**0.34**	0.8	**0.2**	0.32
Dog	**1**	0.1	0.18	**1**	0.1	0.18	**1**	0.1	0.18
Deer	**0.8**	0.2	**0.32**	**0.8**	0.2	**0.32**	0.7	0.2	0.31
Cifar-10	**1**	0.1	0.18	**1**	0.1	0.18	**1**	0.1	0.18
Cat	**0.9**	0.1	0.18	0.7	**0.3**	**0.42**	0.88	0.22	0.35
Bird	**1**	0.1	0.18	**1**	0.1	0.18	**1**	0.1	0.18
Automobile	0.7	**0.2**	**0.31**	**0.9**	0.11	0.20	**0.9**	0.11	0.20

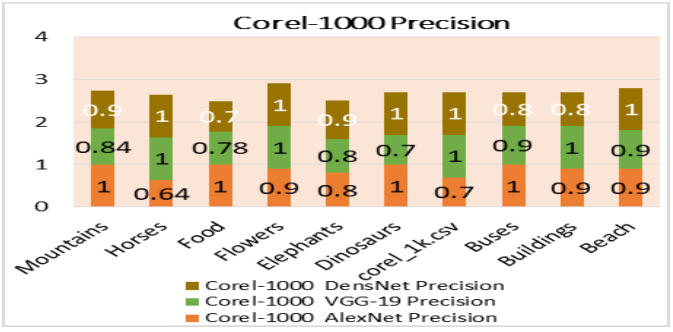
dog, and ship with DenseNet, AlexNet, and VGG-19 architecture. The proposed method also produced better AP performance, as shown in figure 6a and in table 1, which shows the highest AP for all categories of the cifar-10 dataset, Figure 6b, shows the recall ratio and Figure 6c shows the f-score of the cifar-10 dataset with numerical representation in table 1.

The proposed method shows 90% AP in automobile with AlexNet and VGG-19, whereas the cat category shows with DenseNet. The deer category exhibits an average precision (AP) result surpassing 80% with DenseNet and AlexNet architectures. In the DenseNet architecture, the horse category achieves AP rates exceeding 70%. Within the AlexNet architecture, the cat, deer, and frog categories attain AP rates surpassing 70%, while the deer category achieves AP rates exceeding 70% in the VGG-19 architecture. The automobile category shows the 0.2 Average Recall rate with DenseNet. The cat category shows highest recall rate 0.3 with AlexNet. In DenseNet the deer and truck category stated significant performance with 0.2 Average Recall. The category frog recall rate is 0.2 with VGG-19 and the horse highest recall rate0.3 is with DenseNet. The bird, dog, and ship categories demonstrate noteworthy performance with 0.1 Average Recall rate across the DenseNet, AlexNet, and VGG-19 architectures.

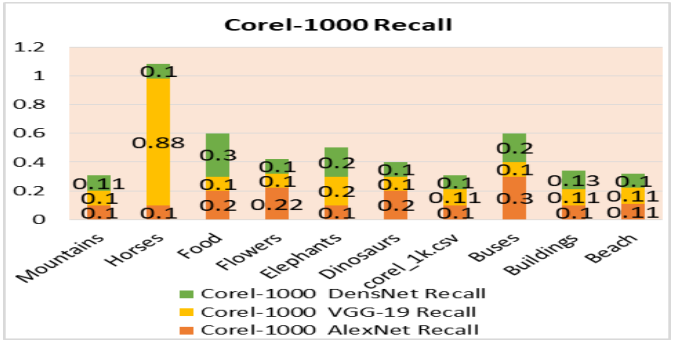
The proposed method shows the 0.31 f-measure for automobile with DenseNet, whereas for bird, dog and ship 0.18 f-measure with DenseNet, AlexNet and VGG-19. In cat category f-score is 0.42 with AlexNet, whereas deer and truck has 0.32 f-score with DenseNet. The category frog has 0.34 with AlexNet and horse has 0.42 with DenseNet architecture.

4.3.2 Experimentation on benchmark Corel-1000 dataset

One large image collection used for image retrieval and classification applications is the Corel-1000 dataset [27]. This dataset encompasses diverse image categories, featuring images with unadorned foregrounds and backgrounds, catering to the representation of complex and cluttered objects. The 1000-image collection is divided into 10 classes, each of which represents a different semantic group, such as buildings, food, flowers, buses, elephants, mountains, horses, beaches, or native people. The graphical depiction of the suggested approach on the Corel-1000 dataset is shown in figure 7.



(a) **Average Precision**



(b) **Recall**

Table 2 Precision, Recall and F-score with AlexNet, VGG-19, and DenseNet of Corel-1000 Dataset

Corel-1000 with AlexNet, VGG-19, and DenseNet									
	AlexNet			VGG-19			DenseNet		
Category	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
Mountains	**1**	**0.1**	0.18	0.84	0.1	**0.89**	0.9	0.11	0.20
Horses	0.64	0.1	**0.44**	1	**0.88**	0.18	**1**	0.1	0.18
Food	**1**	0.2	0.18	0.78	0.1	**0.83**	0.7	**0.3**	0.42
Flowers	0.9	**0.22**	0.20	**1**	0.1	0.18	**1**	0.1	0.18
Elephants	0.8	0.1	**0.44**	0.8	**0.2**	0.18	**0.9**	**0.2**	0.33
Dinosaurs	**1**	**0.2**	0.18	0.7	0.1	**0.31**	**1**	0.1	0.18
corel_1k.csv	0.7	0.1	**0.31**	**1**	0.11	0.18	**1**	0.1	0.18
Buses	**1**	**0.3**	0.18	0.9	0.1	0.20	0.8	0.2	**0.32**
Buildings	0.9	0.1	0.20	**1**	0.11	0.18	0.8	**0.13**	**0.22**
Beach	0.9	0.11	**0.20**	0.9	0.11	0.20	**1**	0.1	0.18

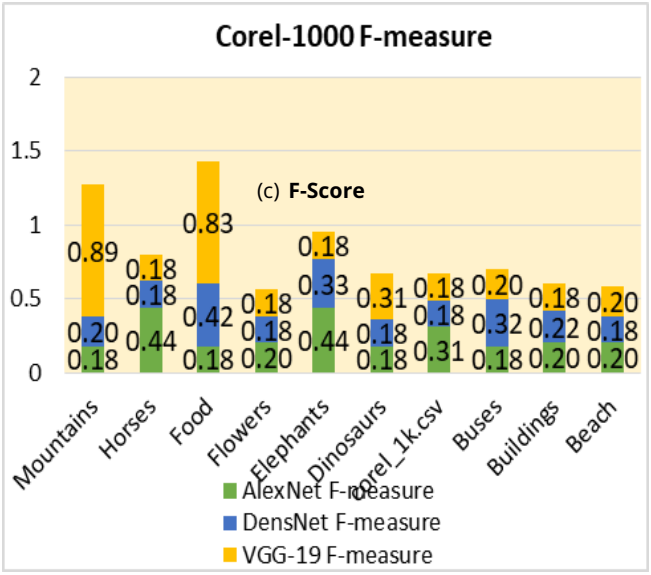
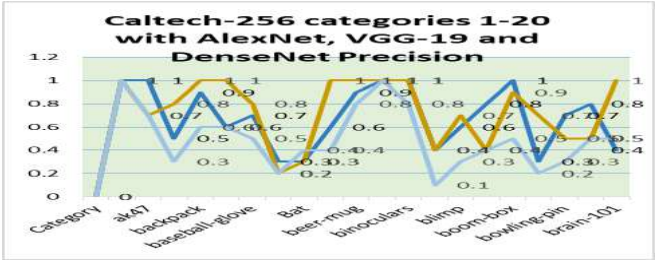


Figure 7. (a) Show the Average Precision (AP), **(b)** shows Recall, **(c)** shows f-score of Corel-1000 Dataset with AlexNet, VGG-19, and DenseNet architecture

The proposed approach processes all ten Corel-1000 dataset categories. Furthermore, the proposed technique excels in certain categories such as beach, building, buses, dinosaurs, flowers, food, elephants, horses, and mountains. Notably, in flower and horse categories demonstrates 100% AP ratio with DenseNet. Furthermore, horses in VGG-19 exhibit 100% AP ratio, and for AlexNet, food, mountain, dinosaurs, and buses also achieve 100% AP ratio, as shown in Figure 7a that shows the highest AP for all the categories of corel-1000 dataset, Figure 7b shows the recall ratio and Figure 7c shows the f-score.

Table 2 represents the numerical values of these categories name, precision rate, recall rate and f-score. 90% Average Precision ratio for elephant in DenseNet, for beach and building in AlexNet and in VGG-19 shows 90% for beach and buses. The presented method demonstrates notable recall results on the Corel-1000 dataset across DensNet, AlexNet, and VGG-19 architectures. The building category stated significant performance with 0.13 recall rate, buses shows highest recall rate 0.3 with AlexNet, and dinosaurs with 0.2 recall rate. The category elephant shows 0.2 recall rate with VGG-19 and DenseNet. In AlexNet category flower with 0.22, food recall rate 0.3 with DesneNet and horse 0.88 recall rate with VGG-19 architecture.

The proposed method shows the 0.20 f-measure in beach with AlexNet, whereas in building 0.22 and in buses 0.32 with DenseNet. The VGG-19 has significant f-score 0.31 in dinosaur's category, whereas elephant has 0.44 and flower has 0.20 with AlexNet. The food category has 0.83 f-score with VGG-19, food has 0.44 with AlexNet and in mountain f-score is 0.89 with VGG-19.



(a) **Shows the top 20 categories precision of Caltech-256 dataset**

4.3.3 Experimentation on large benchmark Caltech-256 dataset

The Caltech-256 dataset, comprising 257 diverse categories and exceeding 30,000 images, exhibits heightened complexity compared to Caltech-101, owing to its varied image categories and content intricacies. Rigorous evaluation is conducted on each individual category within the Caltech-256 dataset to ascertain the precision of classification. The experimental setup involves the use of fifteen distinct image categories, encompassing items such as a bonsai, a clock, a bag, a teddy bear, a cactus, an airplane, a boxing glove, a teapot, a spider, a billiard ball, a swan, a tomato, a bulldozer, a tree, and a butterfly [21]. The significance of each category in the dataset is underscored by its compositional pattern and the presence of foreground and background objects [28].

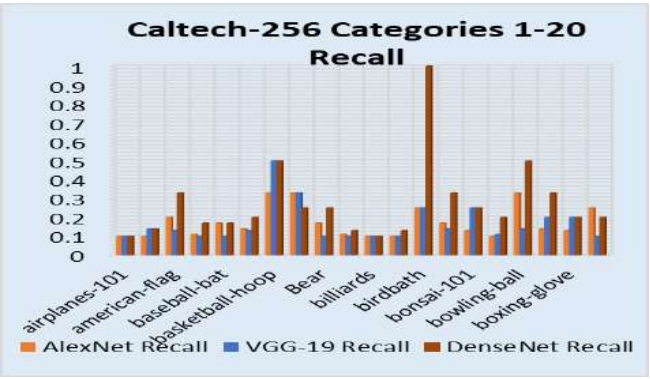
Caltech-256 demonstrates notable performance across the majority of its categories. Distinguished by its voluminous nature with 256 categories, the graphical representation of the Caltech-256 dataset is stratified into the top 20 categories, average 5 categories, and below-average 3 categories, as delineated below.

Graphical representation in figure 8a illustrates the top 20 categories, while table 3 provides the corresponding numerical values, including category names and precision rates. Notably, the categories Airplanes, backpack, baseball-bat, bear, beer-mug, billiards, binoculars, and brain exhibit a 100% Average Precision (AP) when employing the VGG-19 architecture.

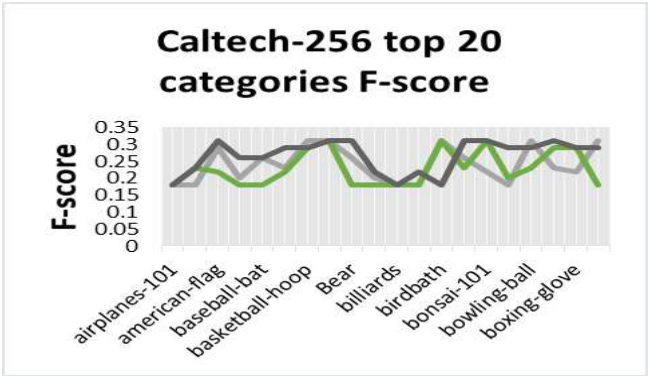
Figure 8b represents the graphical representation of the top 20 categories, while table 3 offers the numerical values pertaining to these categories, encompassing their respective names and recall rates. The highest recall rate is 0.33 in most of the categories American-flag with DenseNet, bat with AlexNet and VGG-19, bowling-ball with AlexNet, and bowling-pin with DenseNet architecture.

Figure 8c represents the top 20 categories, and table 3 supplements this presentation with numerical values denoting category names and corresponding F-score rates. Notably, a predominant number of categories demonstrate the maximal F-score 0.29 and 0.31.

Figure 9a depicts the average precision for the 5 categories, while Table 4 provides the corresponding numerical details including category names and precision rates. Figure 9b visually represents the average recall for the same categories, with Table 4 again supplementing this with numerical values for category names and their respective recall rates. Finally, Figure 9c shows the graphical representation of the average F-score for the categories, complemented by Table 4 which details category names and their corresponding F-score rates.

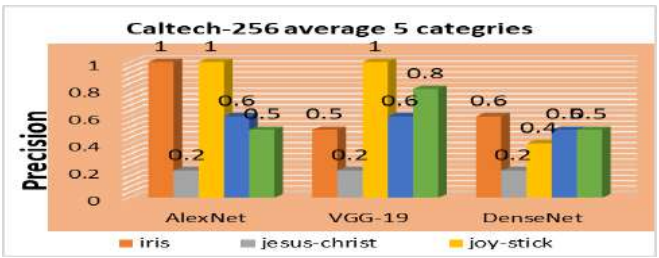


(b) **Shows the recall of top 20 categories of Caltech-256 dataset**



(c) **Shows the F-score of top 20 categories of Caltech-256 dataset**

Figure 8. (a),(b), (c) shows average precision, recall and f-score of top 20 categories of caltech-256 dataset with AlexNet, DenseNet andVGG-19



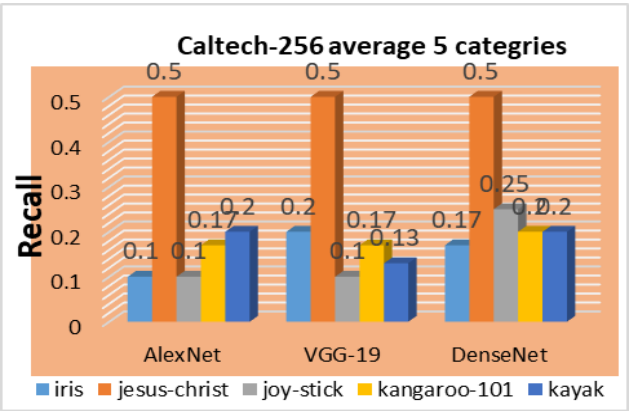
(a) **Average Precision**

Table 3 Precision, Recall and F-score with AlexNet, VGG-19, and DenseNet of Caltech-256 dataset top 20 categories

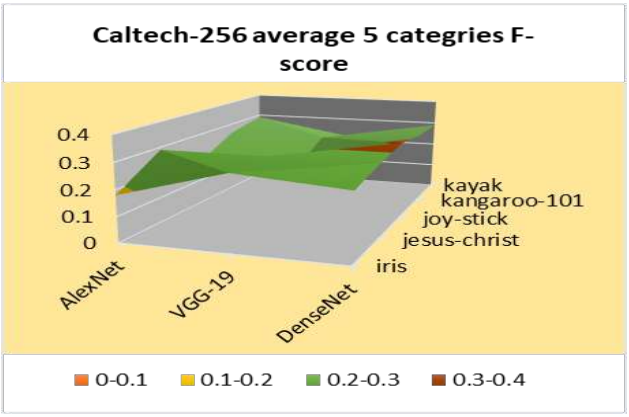
Caltech-256 Dataset top 20 categories									
	AlexNet			VGG-19			DenseNet		
Category	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
airplanes-101	**1**	0.1	0.18	**1**	0.1	0.18	**1**	0.1	0.18
ak47	**1**	0.1	0.18	0.7	0.14	0.23	0.7	0.14	0.23
american-flag	0.5	0.2	0.29	0.8	0.13	0.22	0.3	0.33	0.31
backpack	0.9	0.11	0.2	1	0.1	0.18	0.6	0.17	0.26
baseball-bat	0.6	0.17	0.26	1	0.1	0.18	0.6	0.17	0.26
baseball-glove	0.7	0.14	0.23	0.8	0.13	0.22	0.5	0.2	0.29
basketball-hoop	0.3	0.33	0.31	0.2	0.5	0.29	0.2	0.5	0.29
Bat	0.3	0.33	0.31	0.3	0.33	0.31	0.4	0.25	0.31
Bear	0.6	0.17	0.26	1	0.1	0.18	0.4	0.25	0.31
beer-mug	0.9	0.11	0.2	1	0.1	0.18	0.8	0.13	0.22
billiards	1	0.1	0.18	1	0.1	0.18	1	0.1	0.18
binoculars	1	0.1	0.18	1	0.1	0.18	0.8	0.13	0.22
birdbath	0.4	0.25	0.31	0.4	0.25	0.31	0.1	1	0.18
blimp	0.6	0.17	0.26	0.7	0.14	0.23	0.3	0.33	0.31
bonsai-101	0.8	0.13	0.22	0.4	0.25	0.31	0.4	0.25	0.31
boom-box	1	0.1	0.18	0.9	0.11	0.2	0.5	0.2	0.29
bowling-ball	0.3	0.33	0.31	0.7	0.14	0.23	0.2	0.5	0.29
bowling-pin	0.7	0.14	0.23	0.5	0.2	0.29	0.3	0.33	0.31
boxing-glove	0.8	0.13	0.22	0.5	0.2	0.29	0.5	0.2	0.29
brain-101	0.4	0.25	0.31	1	0.1	0.18	0.5	0.2	0.29

Table 4 AP, Recall and F-score with AlexNet, VGG-19, and DenseNet of Caltech-256 dataset average 5 categories

Caltech-256 Dataset average 5 categories									
	AlexNet			VGG-19			DenseNet		
Category	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
iris	**1**	0.1	0.18	0.5	**0.2**	**0.29**	0.6	0.17	0.26
jesus-christ	0.2	**0.5**	**0.29**	0.2	**0.5**	**0.29**	0.2	**0.5**	**0.29**
joy-stick	**1**	0.1	0.18	**1**	0.1	0.18	0.4	**0.25**	**0.31**
kangaroo-101	0.6	0.17	0.26	0.6	0.17	0.26	0.5	**0.2**	**0.29**
kayak	0.5	0.2	0.29	0.8	0.13	0.22	0.5	0.2	0.29



(b) **Recall**



(c) **F-Score**

Figure 9. a Shows the Precision, **b** shows the Recall and **c** shows the F-score of average 5 categories of Caltech-256 dataset with AlexNet, DenseNet, and VGG-19

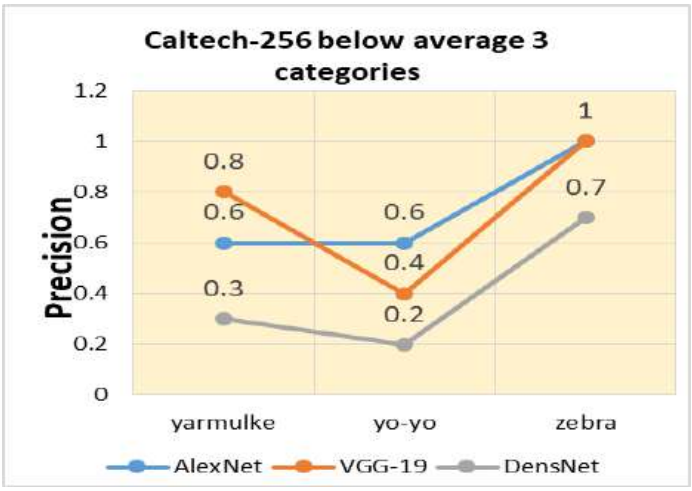
Table 5 Precision, Recall and F-score with AlexNet, VGG-19, and DenseNet of Caltech-256 below average 3 categories

Caltech-256 Dataset below average 3 categories

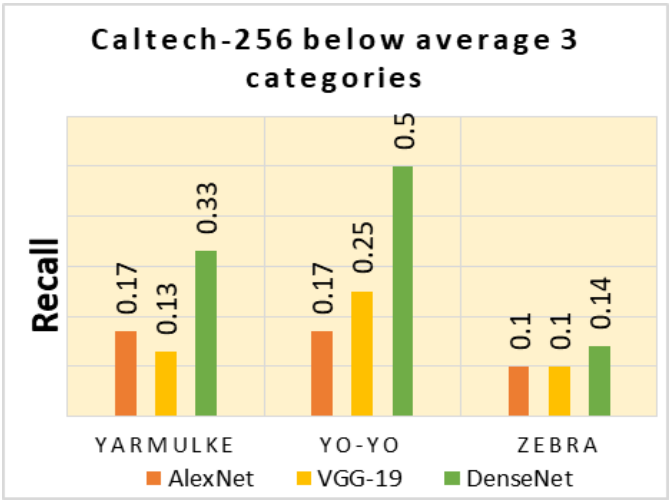
	AlexNet			VGG-19			DensesNet		
Category	Precision	Recall	F-score	Precision	Recall	F-score	Precision	Recall	F-score
yarmulke	0.6	0.17	0.26	0.8	0.13	0.22	0.3	0.33	0.31
yo-yo	0.6	0.17	0.26	0.4	0.25	0.31	0.2	0.5	0.29
zebra	1	0.1	0.18	1	0.1	0.18	0.7	0.14	0.23

Notably, the "iris" category achieves 100% precision when using the AlexNet architecture, while "joy-stick" achieves 100% precision with both AlexNet and VGG-19. Interestingly, the "jesus-christ" category exhibits the highest recall at 0.5 across all three CNN architectures. "Iris" attains a recall of 0.2 with VGG-19, and "joy-stick" achieves a recall of 0.25 with DenseNet. Noteworthy findings include "iris" achieving an F-score of 0.29 with VGG-19, "jesus-christ" attaining an F-score of 0.29 across all three CNN architectures, and "joy-stick" exhibiting an F-score of 0.31 with DenseNet. Additionally, the "kangaroo" category registers an F-score of 0.29 with DenseNet.

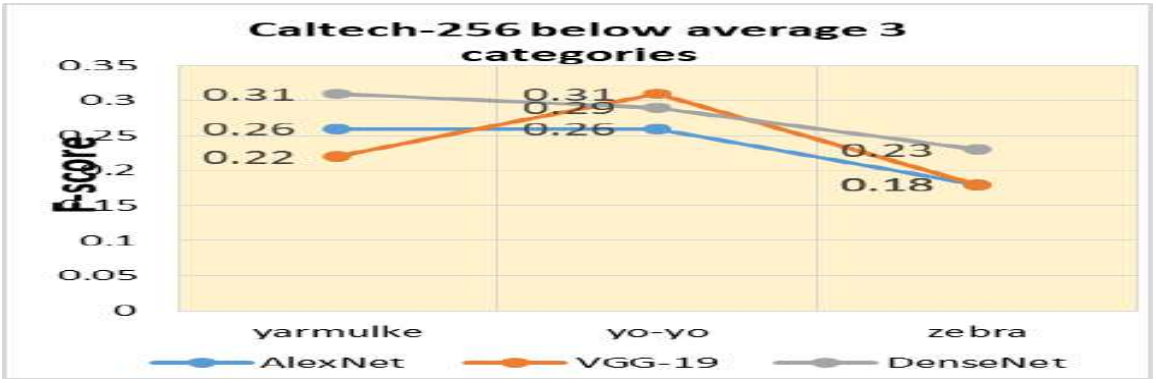
The graphical representation of Caltech-256 dataset is divided into below average 3 categories, as presented underneath. In figure 10a, 10b, and 10c, a graphical representation is provided for the below average 3 categories and table 5 represents the numerical values of these categories name, precision rate, recall rate and f-score. The category yarmulke and yo-yo shows 60% precision rate with AlexNet whereas yarmulke shows 80 % with DenseNet. The category zebra shows 100 % precision rate with AlexNet and VGG-19. The category yarmulke shows 0.33 recall, yo-yo shows recall rate 0.5 and zebra shows 0.14 recall with DenseNet. The category yarmulke shows 0.31 f-score with DenseNet, yo-yo shows f-score 0.31 with VGG-19, whereas zebra shows 0.23 f-score with DenseNet.



(a) **Average Precision**



(b) **Recall**



(c) **F-Score**

Figure 10. a Shows average precision, **b** shows Recall and **c** shows F-score of below average 3 categories of Caltech-256 dataset AlexNet, DenseNet, and VGG-19

The graphical representation of Caltech-256 dataset is divided into below average 3 categories, as presented underneath. In figure 10a, 10b, and 10c, a graphical representation is provided for the below average 3 categories and table 5 represents the numerical values of these categories name, precision rate, recall rate and f-score. The category yarmulke and yo-yo shows 60% precision rate with AlexNet whereas yarmulke shows 80 % with DenseNet. The category zebra shows 100 % precision rate with AlexNet and VGG-19. The category yarmulke shows 0.33 recall, yo-yo shows recall rate 0.5 and zebra shows 0.14 recall with DenseNet. The category yarmulke shows 0.31 f-score with DenseNet, yo-yo shows f-score 0.31 with VGG-19, whereas zebra shows 0.23 f-score with DenseNet.

In Figure 11, the Average Precision (AP) scores for the Caltech-256 Dataset are depicted, presenting the performance of DenseNet, AlexNet, and VGG-19 architectures. All used CNNs have highest average precision of 1.0 on some categories (car-side-101, faces-easy-101, frying-pan, Harp, leopards-101, Mars, Photocopier, motorbikes etc.)

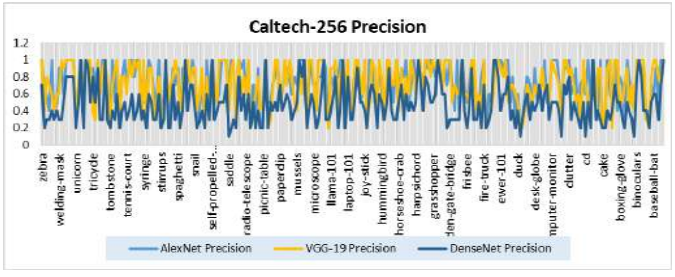


Figure 11. AP of Caltech-256 Dataset with AlexNet, VGG-19, and DenseNet

In Figure 12, the recall ratio for the Caltech-256 Dataset are depicted, presenting the performance of DenseNet, AlexNet, and VGG-19 architectures. CNNs architectures has highest recall rate 0.33 on some categories (computer- mouse, windmill, superman, ladder etc.)

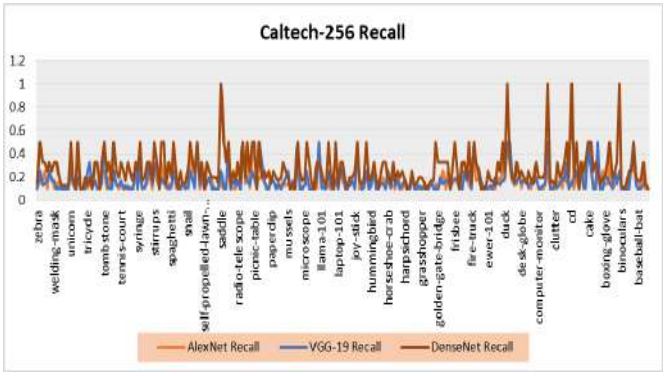


Figure 12. Caltech-256 Recall rate with AlexNet, VGG-19, and DensNet

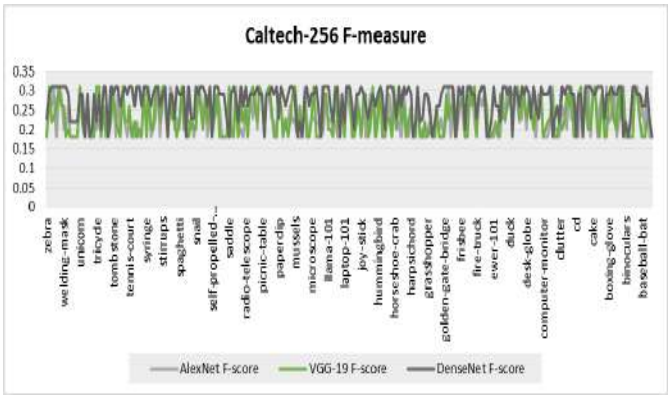


Figure 13. Caltech-256 dataset f-measure with DenseNet, AlexNet and VGG-19

In Figure 13, f-score for the Caltech-256 Dataset are depicted, presenting the performance of DenseNet, AlexNet, and VGG-19 architectures. All used CNNs have highest f-measure of 0.31 on most of the categories.

5 Conclusion

A new method for image extraction using feature vector details and points of interest is proposed. To classify images, the proposed model first converts the input image into grayscale and computes the Hessian matrix and Harris determinants. Blob structuring is then performed to identify potential regions of interest that can adequately describe texture, color, and shape at different representation levels and the Harris corner detector is used to identify keypoints within these regions. Moreover, scale adaptation method is applied to the determinants of the Harris matrix and the Laplacian operator to extract scale-invariant. Meanwhile, the input image undergoes processing through VGG-19, DenseNet, and AlexNet architectures to extract features representing diverse levels of abstraction. Furthermore, the RGB channels of the input image are extracted and their color values are computed. All extracted features local, global, and color subsequently passed through bag-of-words model to rank and retrieve images based on their shared visual characteristics. Combining high-level and low-level information improves the proposed model's ability to distinguish between foreground and background objects. The results show that this technique can successfully distinguish between background and foreground objects in various datasets such as Corel-1000, Cifar-10, and Caltech-256. Proposed method provides high precision and recall for most image categories.

Author Contributions

Khawaja Tehseen Ahmed: Conceptualization, Methodology, Supervision. **Nida Shahid**: Methodology, Writing - Original draft preparation, Supervision. **Syed Burhan ud Din Tahir**: Writing - Original draft preparation. **Aiza Shabir**: Visualization. **Muhammad Yasir Khan**: Writing, Reviewing and Editing. **Muzaffar Hameed**: Writing, Reviewing and Editing.

Compliance with Ethical Standards

The authors declare no conflict of interest. This article does not involve studies with human participants or animals conducted by any of the authors. Informed consent was obtained from all individual participants included in the study.

References

[1] S. R. Dubey, S. K. Singh, and R. K. Singh, "Local neighbourhood-based robust colour occurrence descriptor for colour image retrieval," *IET Image Process.*, vol. 9, no. 7, pp. 578–586, 2015.

[2] S. Hamad, A. Iqbal, S. Naz, N. ul, M. Imran, and B. Al-Haqbani, "Content-based image retrieval using texture color shape and region," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, 2016.

[3] M. Verma and B. Raman, "Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval," *Multimed. Tools Appl.*, vol. 77, no. 10, pp. 11843–11866, 2018.

[4] S. R. Dubey, S. K. Singh, and R. K. Singh, "Boosting local binary pattern with bag-of-filters for content based image retrieval," in *2015 IEEE UP Sect. Conf. Electr. Comput. Electron. (UPCON 2015)*, 2016.

[5] K. T. Ahmed, S. A. H. Naqvi, A. Rehman, and T. Saba, "Convolution, approximation and spatial information based object and color signatures for content based image retrieval," in *2019 Int. Conf. Comput. Inf. Sci. (ICCIS 2019)*, 2019.

[6] A. Alzu'bi, A. Amira, and N. Ramzan, "Content-based image retrieval with compact deep convolutional features," *Neurocomputing*, vol. 249, pp. 95–105, 2017.

[7] K. T. Ahmed, S. Jaffar, M. G. Hussain, S. Fareed, A. Mehmood, and G. Y. U. S. Choi, "Maximum response deep learning using markov, retinal & primitive patch binding with googlenet & vgg-19 for large image retrieval," *IEEE Access*, vol. 9, 2021.

[8] N. Sharma, V. Jain, and A. Mishra, "An analysis of convolutional neural networks for image classification," *Procedia Comput. Sci.*, vol. 132, no. Iccids, pp. 377–384, 2018.

[9] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network," *Appl. Intell.*, vol. 51, no. 2, pp. 854–864, 2021.

[9] A. Abbas, M. M. Abdelsamea, and M. M. Gaber, "Classification of covid-19 in chest x-ray images using detrac deep convolutional neural network," *Appl. Intell.*, vol. 51, no. 2, pp. 854–864, 2021.

[10] C. S. et al., "Going deeper with convolutions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015.

[11] K. T. Ahmed, H. Afzal, M. R. Mufti, A. Mehmood, and G. S. Choi, "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*, vol. 8, pp. 90351–90379, 2020.

[12] M. N. A. et al., "Colour features extraction techniques and approaches for content-based image retrieval (cbir) system," *J. Mater. Sci. Chem. Eng.*, vol. 09, no. 07, pp. 29–34, 2021.

[13] G. Kumar, "A detailed review of feature extraction in image processing systems," pp. 5–12, 2014.

[14] K. Tehseen, A. Muhammad, and A. Iqbal, "Region and texture based effective image extraction," *Cluster Comput.*, vol. 21, no. 1, pp. 493–502, 2018.

[15] K. Tehseen, S. Ummesafi, and A. Iqbal, "Content based image retrieval using image features information fusion," vol. 51, no. November 2018, pp. 76–99, 2019.

[16] MathWorks, "rgb2gray." https://www.mathworks.com/help/matlab/ref/rgb2gray.html#d126e141594. Accessed: 2021-09-15.

[17] H. Bay and A. Ess, "Speeded-up robust features (surf)," vol. 110, pp. 346–359, 2008.

[18] M. Saadetoğlu and Ş. M. Dinsev, "Inverses and determinants of n× n block matrices," *Mathematics*, vol. 11, no. 17, p. 3784, 2023.

[19] N. Ma, X. Zhang, H. T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 11218 LNCS, pp. 122–138, 2018.

[20] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.

[21] K. Kanwal, K. T. Ahmad, R. Khan, N. Alhusaini, and L. Jing, "Deep learning using isotroping, laplacing, eigenvalues interpolative binding, and convolved determinants with normed mapping for large-scale image retrieval," *Sensors (Switzerland)*, vol. 21, no. 4, pp. 1–39, 2021.

[22] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images." https://www.semanticscholar.org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/5d90f06bb70a0a3dced62413346235c02b1aa086. Accessed: 2020-12-02.

[23] D. Bauso, *Game Theory with Engineering Applications*. 2016.

[24] K. Kanwal, K. T. Ahmad, R. Khan, A. T. Abbasi, and J. Li, "Deep learning using symmetry, fast scores, shape-based filtering and spatial mapping integrated with cnn for large scale image retrieval," *Symmetry (Basel)*, vol. 12, no. 4, p. 612, 2020.

[25] A. Naeem, T. Anees, K. T. Ahmed, R. A. Naqvi, S. Ahmad, and T. Whangbo, "Deep learned vectors' formation using auto-correlation, scaling, and derivations with cnn for complex and huge image retrieval," *Complex Intell. Syst.*, vol. 9, no. 2, pp. 1729–1751, 2023.

[26] F. O. Giuste and J. C. Vizcarra, "Cifar-10 image classification using feature ensembles," *arXiv preprint arXiv:2002.03846*, 2020.

[27] A. Shakarami and H. Tarrah, "An efficient image descriptor for image classification and cbir," *Optik*, vol. 214, p. 164833, 2020.

[28] M. Bansal, M. Kumar, M. Sachdeva, and A. Mittal, "Transfer learning for image classification using vgg19: Caltech-101 image data set," *Journal of ambient intelligence and humanized computing*, pp. 1–12, 2023.