Efficient Image Descriptor Generation Using CNN Architectures for Enhanced Image Retrieval

[1]Muhammad Huzaifa Rashid*, [2]Muhammad Haroon, [3]Muhammad Tanveer Meeran, [4]Rana Muhammad Nadeem , [5]Sadia Latif

[1]Department of Computer Science, NFC Institute of Engineering and Technology, Multan, Pakistan.
[2] School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, 710048, China.
[3]Faculty of Computer Science and Mathematics, Universiti Malaysia Terengganu, Malaysia.
[4]Department of Computer Science, Govt. Graduate College Burewala, Pakistan.
[5]Department of Computer Science, Bahauddin Zakaria University, Multan, Pakistan.
**Corresponding Author: Muhammad Huzaifa Rashid. Email: huzaifarashid6447@yahoo.com*

DOI: https://doi.org/

Keywords (Image descriptors, GoogleNet, Feature extraction, Image retrieval, Image classification).

Article History
Received on 08 July 2025
Accepted on 22 July 2025
Published on 04 August 2025

Copyright @Author
Corresponding Author: *
Muhammad Huzaifa Rashid

Abstract

Machine learning algorithms are widely employed in image classification tasks to extract and represent discriminative features from images. In this study, we present an efficient approach for generating image descriptors using Convolutional Neural Network (CNN) architectures, including GoogleNet, Inception V3, and DenseNet-201. These networks are leveraged to capture both texture and object-level features, which are further encoded through three color channels to enhance image retrieval performance while maintaining an optimal response time. When images are processed through the hierarchical layers of the CNNs, distinctive feature representations (signatures) are produced. These signatures are subsequently used to construct a new matrix that effectively encodes spatial relationships, color attributes, and latent patterns, thereby providing a more comprehensive representation of image content. The proposed CNN-based method was evaluated on four benchmark datasets: Corel-1K, CIFAR-10, 17-Flowers, and ZuBuD. Among the tested architectures, DenseNet-201 achieved the best performance on the CIFAR-10 dataset, which contains images of diverse categories and varying sizes, demonstrating superior accuracy compared to GoogleNet and Inception V3.

INTRODUCTION

Databases are increasingly being used to store digital images from various organizations. With the widespread use of digital cameras and imaging applications, the volume of image data stored online in libraries and databases has expanded rapidly. As these image collections grow, retrieving specific images becomes challenging, especially in the absence of textual metadata. In the field of computer vision, numerous techniques have been developed to identify image attributes such as color, texture, segmented regions, keypoints, and descriptive keywords through sampling. To address these challenges, **Content-Based Image Retrieval (CBIR)** systems have been introduced. CBIR focuses on retrieving relevant images based on visual content rather than relying solely on textual annotations. It extracts low-level features such as shape, color, and texture directly from the image, allowing the system to classify and assign semantic meaning to the images automatically. These features are stored in a database, and similarity comparisons are conducted to retrieve images that match a given query. To bridge the gap between low-level visual features and high-level semantic concepts, machine learning algorithms are employed, enabling more effective image categorization and enhancing CBIR performance.

The core components of a CBIR system include **feature extraction** and **similarity measurement**, both of which are user-dependent. While low-level features encompass visual attributes like shape and texture, high-level features reflect semantic understanding. A key challenge in CBIR is the semantic gap—the discrepancy between low-level features and human visual perception—which often affects system performance. To mitigate this issue, deep learning models are integrated with machine learning techniques, enhancing the system's ability to learn complex patterns from image data.

To further improve CBIR robustness and efficiency, a combination of color and object-based features is used. The retrieval process begins with a query image, from which texture, color, and other descriptors are extracted following preprocessing. Principal Component Analysis (PCA) is then applied to refine these features. To enable fast and efficient retrieval from large datasets, a Bag of Words (BoW) model is constructed using the extracted features, which helps in indexing and identifying relevant images. These descriptors effectively capture image patterns such as edges and corners, which are essential for object recognition and retrieval.

In this study, the proposed approach is evaluated on several publicly available datasets, including ALOT, CIFAR-10, Fashion-MNIST, and Oxford 102 Flowers. Various CNN architectures such as GoogleNet, VGG19, AlexNet, DenseNet-201, and NASNetLarge are applied to subsets of these datasets. The performance is assessed based on the system's ability to capture color, texture, and descriptive features, with a focus on maximizing retrieval accuracy and minimizing response time.

1.1. Content-Based Image Retrieval

Content-Based Image Retrieval (CBIR) is a technique that focuses on analyzing the visual characteristics of images such as color, shape, texture, and spatial features rather than relying on keywords or labels. In CBIR systems, the actual content of the image is examined to identify and retrieve similar images from a database based on feature similarity. This approach plays a significant role in various fields, including defense, healthcare, agriculture, architecture, and education.

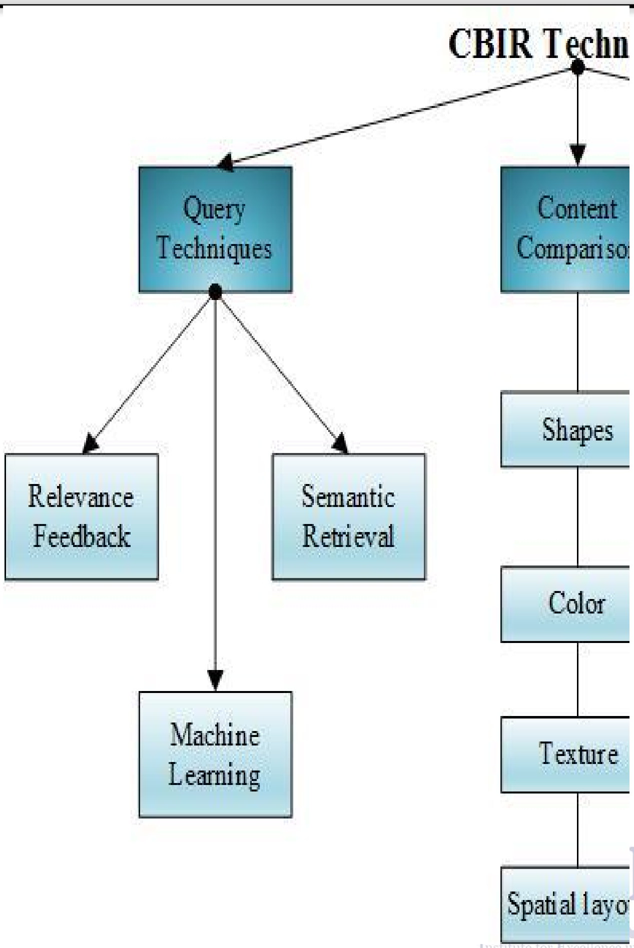

Figure 1: CBIR Techniques [1]

Related work

Recent years have seen a dramatic increase in digital image usage. Retrieving specific images via queries from large datasets has become particularly challenging, prompting researchers to develop robust Content-Based Image Retrieval (CBIR) techniques. These methods extract visual features such as shape, color, corners, edges, noise, and texture for efficient image retrieval. A hybrid approach combines Convolutional Neural Networks (CNNs) with CBIR to enhance retrieval accuracy. Textures are sampled based on neighboring key points, and standard deviation is used for smoothing and pattern comparison. Features are reduced through Principal Component Analysis (PCA) and ResNet is leveraged for feature generation. Additionally, color coordinates are integrated into CNN architectures, resulting in high-precision image features used for smoothing, scaling, and sampling [8]. To accelerate training, one study integrates spatio-temporal and multi-resolution information, improving performance over traditional models like UCF- 101. The proposed architecture better preserves connections over time, surpassing slow- fusion networks and reducing error rates [9]. Comparative studies using the Normalized Difference Vegetation Index (NDVI) involve CNNs to assess single- and multi-sensor scenarios over time. While initial results showed both cross-sensor and temporal dependencies, gaps remain in generalization across diverse datasets [10]. Several studies focus on combining color and object-based features using BoW indexing. PCA is applied to discard redundant features, and mixed representations (RGB + grayscale) yield strong precision and recall across datasets [11]. Feature extraction methods such as MORAVEC corners, covariance-based edge scoring, and binary pattern encoding are applied to datasets like ImageNet, Caltech, and Corel. Traditional descriptors such as HOG, SIFT, and SURF perform well on some datasets but struggle in others due to limited attributes [12].

BiCBIR, a dual-stage system, first retrieves images using color and texture, and then refines results based on shape and color comparison, delivering fast and accurate retrieval. Authors propose further enhancing it by integrating CNNs [13]. Experiments on varied images (diverse shapes, textures, backgrounds) harness multiple CNNs combined into eigenvalue-based feature mappings. This yields improvements in retrieval rate and efficiency across large datasets [14]. Research aiming to optimize features like texture, segmentation, interest points, and keywords employs symmetric scoring (FAST), smoothing, and reduction techniques to scale for large datasets. Results suggest potential benefits when compared with ResNet [38][8]. A triple-network approach uses semantic feature extraction to cluster similar images and separate dissimilar ones. Euclidean distance measures similarity, and feature dimensionality is minimized to improve retrieval performance [15]. A general CBIR framework optimized for domains like forensic fingerprinting, facial recognition, and digital libraries uses a seven-step feature selection, extraction, clustering, and similarity measurement pipeline. Algorithms such as EAAQ, SEPAM, MYOLO, and ERDO enhance body-part segmentation and overall retrieval accuracy [16]. Enhanced color histograms—including Color Coherent Vector (CCV), hybrid and angular/annular histograms have been developed to better represent color distribution and improve search precision in CBIR systems [3, 33]. CRB- CNN, a CNN-based image retrieval model, simultaneously extracts convolutional features and applies compact pooling to reduce dimensionality. This strategy decreases storage and accelerates retrieval while maintaining performance [17]. CNN-based systems have been adapted for diagnosing malaria from blood smear images, though further work is required to close accuracy gaps caused by parasitic artifacts [18].

Texture feature extraction combined with BoW and language models enables texture-based retrieval, using CNNs to capture feature distributions from image inputs [19]. DNNs classify medical images (e.g., diabetic diagnostics) using models like VGGNet, PCA, GMM, AlexNet, and SIFT-based features, delivering fast and accurate results [6]. Signature- based CBIR techniques exploit shape, color, and interest-point derivatives clustered via coefficient-based methods. Features are converted into Bag-of-Words and ranked, yielding reliable precision and recall metrics [20]. Visual saliency is integrated into CBIR by using dual-stream CNNs: one network extracts divergent features, while the other reinforces salient content. The auxiliary channel aids the main stream for enhanced retrieval quality [21]. Techniques that minimize intra-class feature distance using entropy-optimized deep networks boost descriptive feature retrieval and gap minimization between relevant/irrelevant images [22]. CBIR systems using VGG16 and ResNet50 fuse color, texture, and shape features from pre-trained models to support image retrieval across satellite and remote sensing datasets [23]. Blend of seven local/global detectors (e.g., RGBLBP, MSER, HoG, SURF, SIFT, LBP) followed by PCA and L2-normalization enhances retrieval performance across multiple datasets [24]. Hybrid models combining ANN, SVM, and genetic algorithms (like GCCL) improve classification outcomes and retrieval accuracy in CBIR [25]. Comparative studies of color spaces (RGB, TUV, HSV) paired with octree-based indexing show adaptive retrieval performance based on color quantization techniques [26].

2. Methods and materials

This study focuses on the fundamental stages of feature detection, key point matching, structural analysis, and the use of image descriptors. These components are essential for readers aiming to understand and replicate the methodology. One of the key challenges in feature extraction lies in balancing two primary objectives: generating high-quality descriptions and maintaining computational efficiency.

2.1. Continuous Scale-Space Representation

To identify stable key points across various scales, saliency-based criteria are employed. This involves detecting intersection points within both the image and its corresponding scale dimension. Achieving scale invariance requires locating high-quality key points beyond the traditional image plane by identifying local maxima in scale space. In the subsequent step, saliency scores are assigned to these scale-based key points. Rather than sampling the scale axis at fixed intervals, advanced detectors are utilized to estimate the optimal scale for each key point in a continuous scale-space framework. This approach enhances precision and robustness in the detection process. The complete process is visually summarized in the accompanying figure.

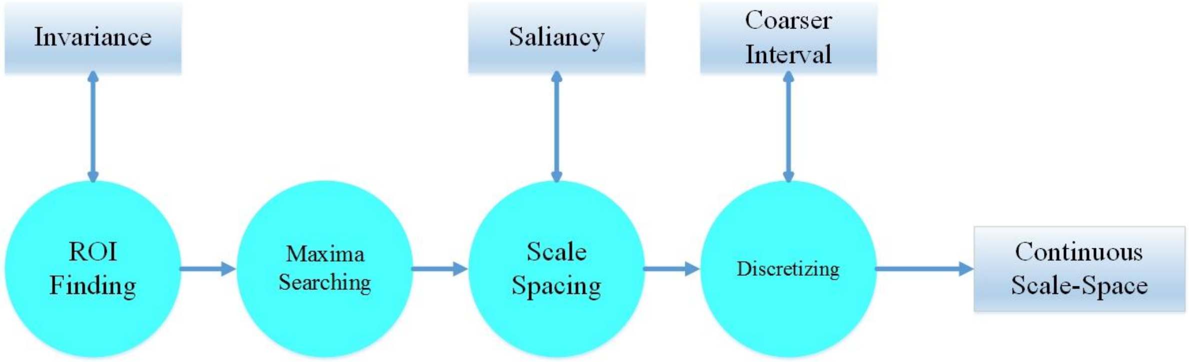

Figure 2: Continuous Scale Space

2.2. Pyramid Layer Construction

Following the continuous scale-space representation, a pyramid structure is formed. This pyramid is composed of multiple octaves and inter-octaves, typically with a total of four layers (n = 4). Each octave is generated from the original image through downsampling, where each new octave contains half the resolution of its predecessor. Inter-octave layers, which exist between two consecutive layers cic_ici and ci+1c_{i+1}ci+1 , are produced by scaling the original image by a factor of 1.5. Subsequent inter-octaves are derived by continuing the down sampling process. For feature detection within these layers, corner detection techniques such as FAST (Features from Accelerated Segment Test) and AGAST (Adaptive and Generic Accelerated Segment Test) are utilized. The FAST detector uses a 9–16-pixel mask—meaning that among a 16-pixel circle around a candidate pixel, at least 9 pixels must be either consistently brighter or darker than the center pixel to qualify as a key point. FAST detection is applied across both octave and inter-octave layers, with thresholding used to identify joint regions. Non-Maximum Suppression (NMS) is then used to isolate key points by evaluating their saliency relative to eight neighboring pixels within the same layer. A pixel is retained as a key point if its score is the maximum in its vicinity, and higher than those in the layers directly above and below. Only square-shaped images with equal side lengths are used, with a maximum side length difference of 2 pixels. At the layer boundaries, interpolation is applied to manage score variations across neighboring regions.

2.3. Descriptor Sampling

Descriptor sampling involves selecting points on concentric circular regions around each detected key point. These sample points are used to extract grayscale intensity values from the image, which are then compared to generate a binary descriptor. This binary comparison approach ensures efficient key point matching. Inspired by the human visual system, particularly the retina's ability to detect and distinguish features, the Fast Retina Key point (FREAK) method is employed to enhance object detection within images. Each key point is defined by its position and scale (sub-pixel accuracy), and its binary descriptor is formed by encoding the results of brightness comparisons between sampled pairs of points. Orientation and rotation-invariance are ensured through the use of normalized descriptors. The descriptor pattern mimics that of the DAISY descriptor, with sample locations evenly distributed along circular paths centered on the key point. Gaussian smoothing is applied with a defined standard deviation to reduce aliasing effects during sampling. The sampling pattern is then scaled and rotated according to the key point's orientation to maintain consistency across transformations. For efficient rotation and scale normalization, the sampling pattern is aligned with the key point's dominant direction. Binary descriptors perform localized intensity comparisons, and methods like BRIEF (Binary Robust Independent Elementary Features) are implemented to encode these comparisons. Pairwise sampling is preferred over single-point comparisons, as it enables faster and more reliable matching. Finally, hamming distance is used to measure dissimilarity between descriptors by counting the number of differing bits, facilitating quick and efficient feature matching.

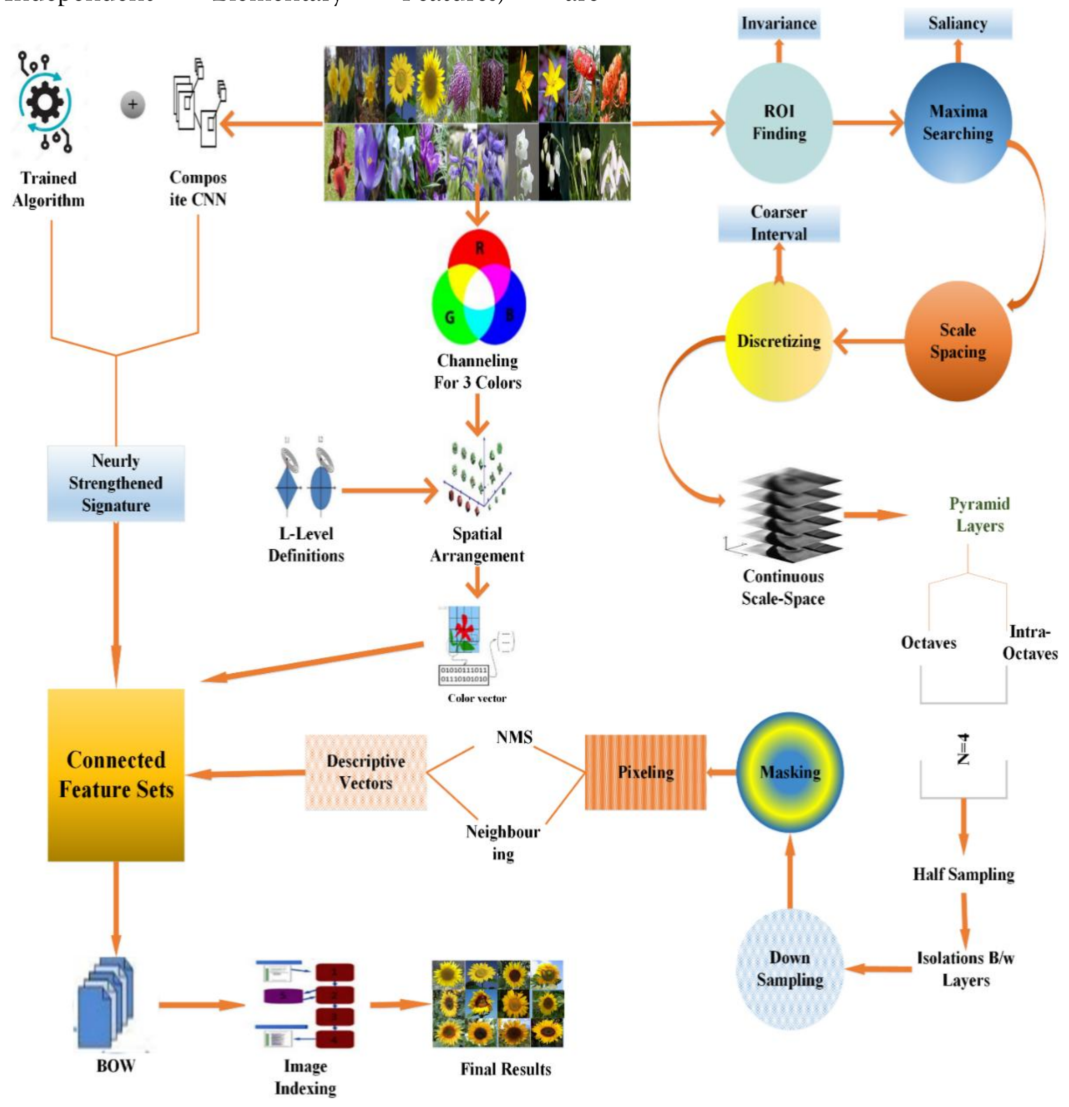

Figure 3: The proposed method includes a step-by-step demonstration of the process

2.4. Spatial Color Feature Extraction

The image retrieval process begins with analyzing the image to extract similarity-based features. Since the input consists of color images, distinguishing between relevant and irrelevant images is based on their feature similarity. Two main types of features are considered: local features, such as image segmentation and edge detection, and global features, which include color and texture information.

In the case of color features, the process involves calculating the first and second-order derivatives, as well as the area associated with each pixel. These computed values are stored in a structured array. Additionally, the RGB (Red, Green, Blue) color values for each pixel are recorded. Edges within the image are detected and utilized to form a feature vector, which contributes to identifying and comparing visual content effectively.

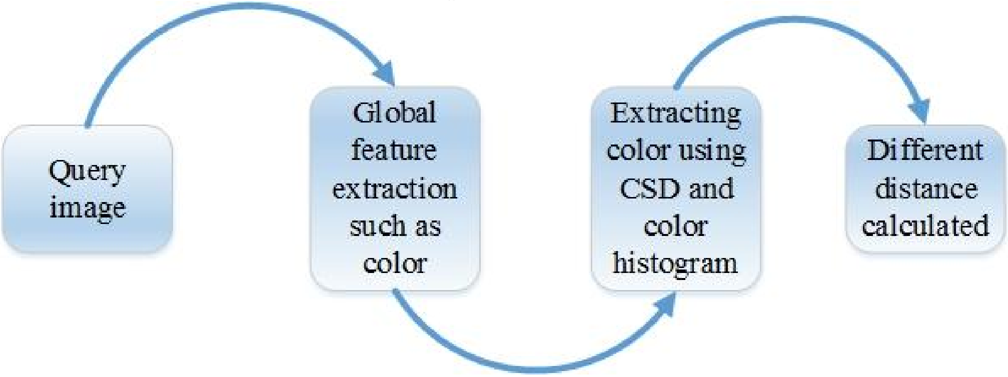
Query image
Global feature extraction such as color
Extracting color using CSD and color histogram
Different distance calculated

Figure 4: Color Features Extraction

Our framework outlines the key phases of feature extraction, keypoint matching, structural representation, and descriptor formulation. These elements are essential for readers to understand and replicate the method. One of the central challenges lies in balancing detailed feature representation with computational efficiency. Color is a critical visual attribute in CBIR systems. It aids in distinguishing relevant from irrelevant images with minimal error while maintaining compact feature representations. CBIR methods often utilize various color spaces (e.g., RGB, HSV) to encode pixel information. Color histograms are computed across image regions to summarize color distributions. Images are segmented into grids, and histograms are generated for each sub-region. Histogram bins, typically normalized within a range (e.g., 0.0 to 0.01), represent the frequency of pixel colors. A Color Structure Descriptor is also applied, which quantifies how often specific colors appear in structured elements of the image. Alongside color histograms, edge-based descriptors and color moments are extracted to build the overall color feature vector. Similarity between a query image and database images is computed using metrics such as the Euclidean distance between color feature vectors.

Image Indexing Using Bag-of-Words (BoW)

After extracting CNN-based feature signatures, color descriptors, and texture details, we employ a Bag-of-Words (BoW) model for image indexing. Each image is represented as a fixed-length vector, where each visual word corresponds to a cluster of similar descriptors (e.g., SIFT). Visual words are quantized into histograms, and an inverted index is created to enable efficient image retrieval. For each image, the number of occurrences of each visual word is tallied, and ranking is based on similarity between histogram vectors. While BoW does not explicitly encode spatial or color information, augmenting it with descriptive and color vectors, combined with CNN features, improves retrieval accuracy significantly.

3. Results and implementation

To assess the performance and efficiency of the proposed CBIR system, several publicly available benchmark datasets were utilized. These datasets vary in complexity, image type, and object diversity, ensuring a comprehensive evaluation. The four datasets employed are:

- ✓ *CIFAR-10*: Contains 10 distinct classes
- ✓ *Oxford 17-Category Flower Dataset*: Includes 17 categories of flowers
- ✓ *Corel-1K*: Comprises 10 classes of natural images
- ✓ *ZuBuD (Zurich Buildings Database)*: Consists of 250 different building categories

Each dataset offers unique visual attributes such as color composition, texture, shape, and object complexity. The retrieval performance varies across datasets due to their inherent differences in class structure and feature diversity.

3.1. Image Processing

The CBIR system begins with the ingestion of color images, which are initially transformed into grayscale format to standardize input and reduce complexity. Feature extraction is carried out using deep learning-based models such as GoogleNet, Inception v3, and DenseNet-201. The selected models are trained on the aforementioned datasets to extract discriminative features. After feature extraction, the Bag-of-Words (BoW) method is employed to generate feature vectors and index images efficiently. This indexing process facilitates the classification of key visual characteristics, including texture, color, shape, and object identity.

3.2. Evaluation metrics

The effectiveness of image retrieval is measured using two key metrics:

$$\text{MAP} = \frac{\sum_{p=1}^{i} E(p)*rel(p)}{r} \quad \text{Equation (1)}$$

- *Precision*: Represents the proportion of relevant images retrieved out of all retrieved images.
- *Recall*: Measures the proportion of relevant images retrieved out of the total relevant images present in the dataset.

Mathematically:

- *Precision* = (Number of Relevant Retrieved Images) / (Total Retrieved Images)
- *Recall* = (Number of Relevant Retrieved Images) / (Total Relevant Images in Dataset)

These metrics provide insight into the system's accuracy and retrieval capability across various categories. Each category's precision and recall values are computed individually.

Average Retrieval Precision (ARP) is used to evaluate the retrieval performance for each image category across different datasets. The ARP is computed using the average of individual category-wise precision scores. A bar graph is used to visually present ARP results, where each bar represents the number of correctly retrieved images for a specific category. The x-axis indicates the categories, and the y-axis shows the corresponding average precision scores.

The ARP is calculated for each of the following datasets:

- *CIFAR-10*
- *17-Flowers*
- *Corel-1K*
- *ZuBuD*

To evaluate the overall retrieval performance, Mean Average Precision (MAP) is computed. MAP is derived from the ARP values of all categories across a dataset and reflects the system's general ability to return relevant images. MAP is defined as the mean of the average precision scores calculated for all queries. It offers a unified metric to compare performance across different image datasets.

In Equation (1) defines the calculation of average precision, denoted by **E(p)**, for the top *p* retrieved images. The term **rel(p)** represents the relevance of the retrieved image: it is assigned a value of 1 if the retrieved image matches the query image, otherwise it is 0. The variables **r** and **i** refer to the retrieved images and their respective indices at the time of retrieval.

3.3. System Configuration and Experimental Results

The experiments were conducted on a Dell Core i5 system with 8 GB RAM. MATLAB R2020b was used for implementing and evaluating the CBIR framework. For feature extraction and classification, deep learning models such as GoogleNet**,** Inception v3, and DenseNet-201 were employed using MATLAB's Deep Learning Toolbox. A range of benchmark image datasets were tested to validate system performance.

The Oxford 17-Flowers dataset was utilized to evaluate the retrieval system's effectiveness. The performance metrics calculated include Average Precision (AP)**,** Average Recall (AR)**,** Average Retrieval Precision (ARP)**,** Average Retrieval Recall (ARR)**,** Mean Average Precision (MAP)**, and** Mean Average Recall (MAR). This dataset includes 17 distinct flower categories such as: *Bluebell, Buttercup, Coltsfoot, Cowslip, Crocus, Daffodil, Daisy, Dandelion, Fritillary, Iris, Lily of the Valley, Pansy, Snowdrop, Sunflower, Tigerlily, Tulip*, and Below is a category-wise summary: *Windflower*. Each category contains 80 images, resulting in a total of 1360 images in the dataset. The dimensions of the images vary across the dataset. For each category, common visual attributes like color**,** shape, and object structure were considered. The system consistently delivered high average precision scores across most categories. A figure included in this section illustrates sample images from various flower categories in the 17-Flowers dataset.

Performance Evaluation on the 17-Flowers Dataset

The following table summarizes the performance of three deep learning architectures GoogleNet, Inception v3, and DenseNet-201 on the 17-Flowers dataset. Each model was tested using the top 20 retrieved images per query. The evaluation metrics include Precision and Recall for each flower category.

GoogleNet showed moderate performance across most categories, with particularly high precision for *Fritillary* (0.95) and *Sunflower* (0.75).

Inception v3 consistently outperformed GoogleNet in several categories, achieving perfect precision (1.0) for *Fritillary* and high scores for *Pansy* (0.95) and *Sunflower* (0.85).

DenseNet-201 delivered the best overall performance, attaining full precision (1.0) in both *Fritillary* and *Windflower* categories and high precision in others such as *Dandelion*, *Sunflower*, and *Daisy* (all at 0.95).

Table 1: Precision and recall represented in tabular form for 17-flowers dataset.

17-flowers			
Flower Category	**GoogleNet (Precision / Recall)**	**Inception v3 (Precision / Recall)**	**DenseNet-201 (Precision / Recall)**
Bluebell	0.65 / 0.04	0.85 / 0.03	0.55 / 0.05
Buttercup	0.50 / 0.05	0.55 / 0.05	0.80 / 0.03
Coltsfoot	0.45 / 0.06	0.65 / 0.04	0.95 / 0.03
Cowslip	0.30 / 0.08	0.40 / 0.06	0.55 / 0.05
Crocus	0.45 / 0.06	0.55 / 0.05	0.70 / 0.04
Daffodil	0.40 / 0.06	0.45 / 0.06	0.45 / 0.06
Daisy	0.45 / 0.06	0.70 / 0.04	0.95 / 0.03
Dandelion	0.50 / 0.05	0.70 / 0.04	0.95 / 0.03
Fritillary	0.95 / 0.03	1.00 / 0.03	1.00 / 0.03
Iris	0.55 / 0.05	0.65 / 0.04	0.80 / 0.03
Lily of the Valley	0.45 / 0.06	0.65 / 0.04	0.85 / 0.03
Pansy	0.70 / 0.04	0.95 / 0.03	0.80 / 0.03
Snowdrop	0.30 / 0.08	0.55 / 0.05	0.50 / 0.05
Sunflower	0.75 / 0.03	0.85 / 0.03	0.95 / 0.03
Tigerlily	0.40 / 0.06	0.90 / 0.03	0.50 / 0.05
Tulip	0.30 / 0.08	0.35 / 0.07	0.40 / 0.06
Windflower	0.45 / 0.06	0.55 / 0.05	1.00 / 0.03

Figure 5: 17-Flowers dataset showing different sample images of categories

The average precision for each category within the 17-Flowers dataset is illustrated using a bar graph. This visualization highlights the classification effectiveness of three deep learning models: **GoogleNet**, **Inception v3**, and **DenseNet-201**. These models use convolutional neural networks (CNNs) combined with feature mapping, image scaling, and integration techniques to accurately classify images. The precision scores are scaled to a maximum of 1.0. Among all models, **DenseNet-**

201 demonstrated the highest accuracy, achieving **100% average precision** for *Fritillary* and *Windflower*. Additionally, *Coltsfoot*, *Daisy*, *Dandelion*, and *Sunflower* categories recorded **95% precision** using DenseNet-201. Several other categories maintained average precision values around **70%**. In contrast, **GoogleNet** showed lower performance, particularly on the **Corel-1K** dataset, where precision dropped to just above **30%** in certain categories. With **Inception v3**, over half of the categories from Corel-1K achieved more than **70% average precision**. The comparative results of average precision for Corel-1K are depicted in Figure 6.

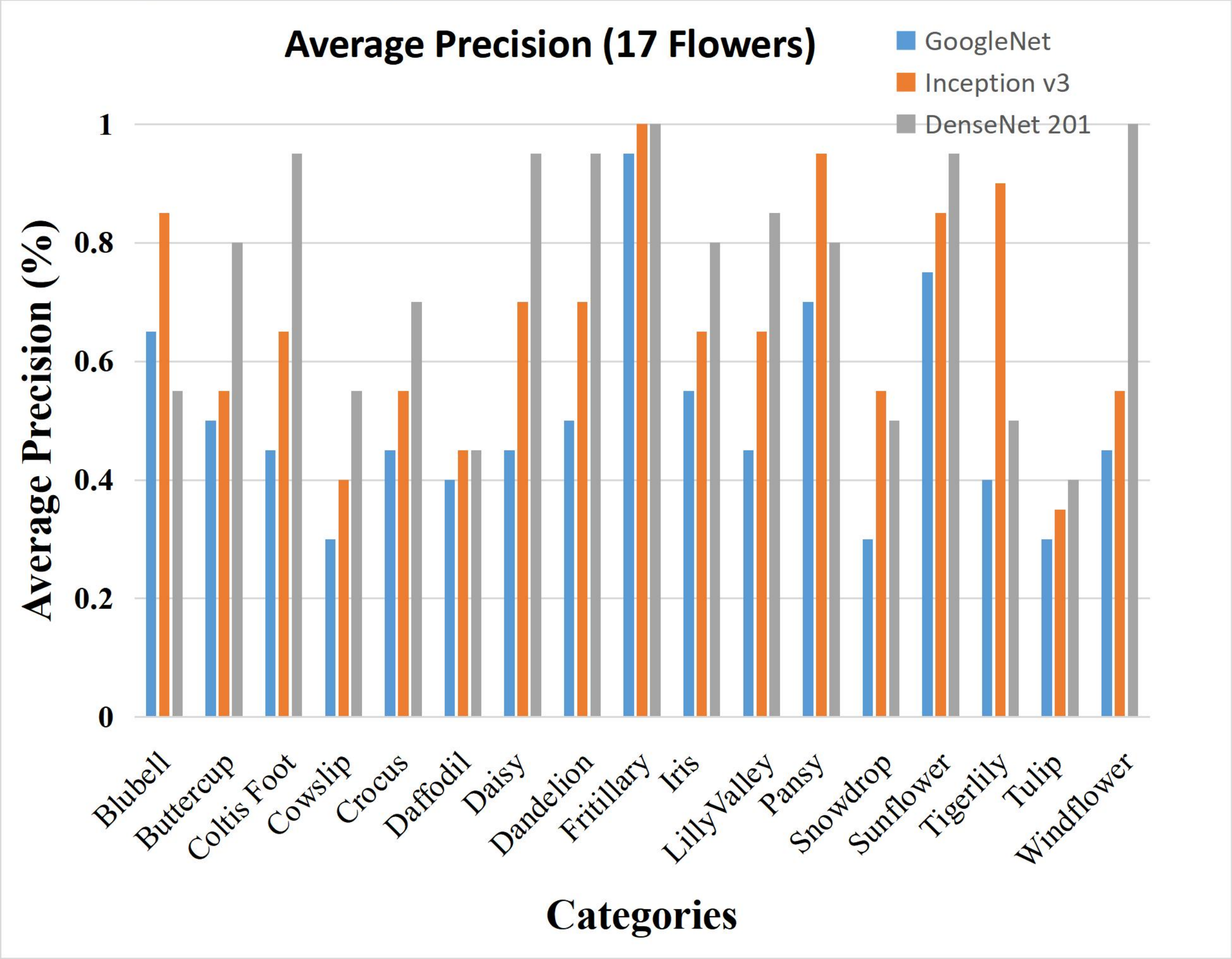

Figure 6: Precision rate for categories of 17-flowers dataset

In Figure 7 presents the average recall for all 17 categories in the 17-Flowers dataset, assessed using **GoogleNet**, **Inception v3**, and **DenseNet-201**. The bar graph illustrates category-wise variations, offering insights into the retrieval performance across different flower types.

Among the models, **GoogleNet** exhibited strong recall performance in categories such as *Cowslip*, *Snowdrop*, and *Tulip*, with recall rates reaching nearly **80%**. However, the majority of categories recorded recall values at or below **60%**. Specifically, categories like *Coltsfoot*, *Crocus*, *Daffodil*, *Daisy*, *Lily of the Valley*, *Tigerlily*, and *Windflower* showed an average recall of approximately 60% with GoogleNet.

DenseNet-201, on the other hand, reported lower recall scores for many categories, with nearly half of them falling at or below 30%, despite its overall strong precision performance. Inceptionv3 demonstrated moderate results, with average recall values mostly ranging between 30% and 60%, and a general average around 40%. The detailed average recall statistics across all categories are visualized in Figure 7.

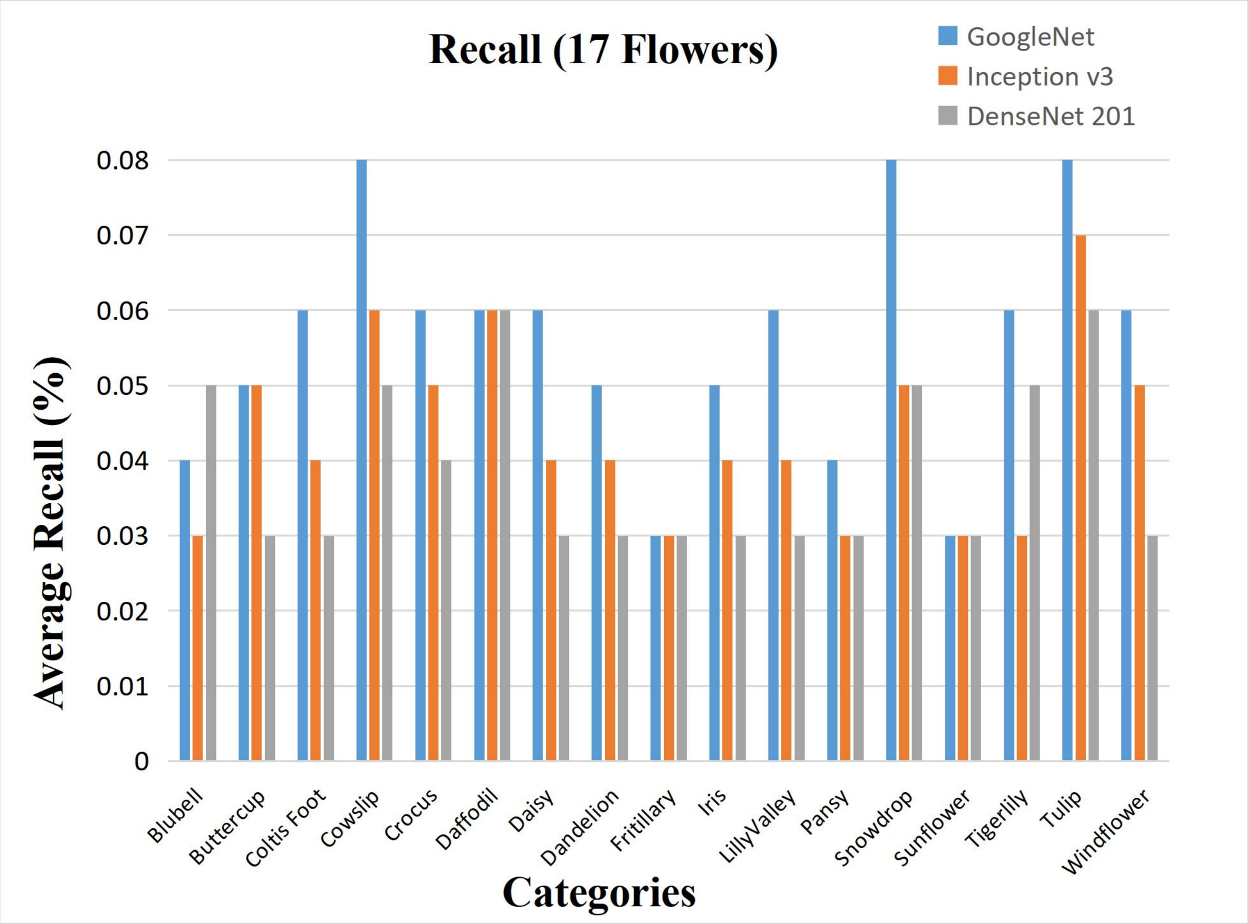

Figure 7 :Recall rate for categories in 17-flowers dataset

Figure 8 illustrates the **Average Retrieval Precision (ARP)** across all categories of the 17-Flowers dataset, using three CNN-based feature extraction models: **GoogleNet**, **Inception v3**, and **DenseNet-201**. Among these, **DenseNet-201** delivered the most consistent performance, achieving an ARP of approximately **76%** across the majority of flower categories. The highest individual precision was recorded for the *Bluebell* category, reaching close to **90%** when using **Inception v3** as the feature extractor. **GoogleNet**, comparatively, exhibited the lowest ARP, falling below **80%** for most categories. In contrast, **DenseNet-201** maintained ARP values within the **80% to 90%** range, highlighting its superior performance in accurate image retrieval within this dataset.

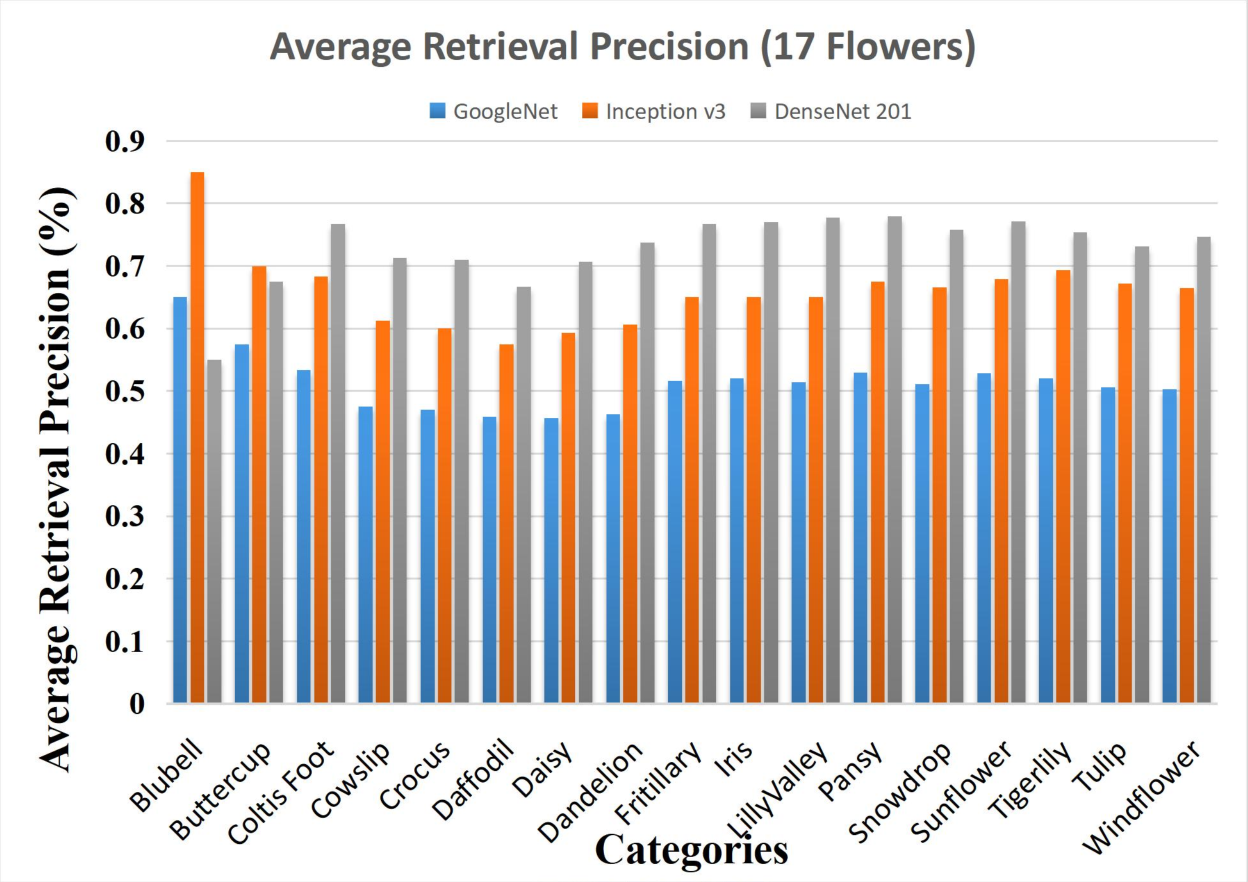

Figure 8: Average Retrieval precision rate for categories of 17 flowers in 17-flowers dataset.

Figure 9 presents the **Mean Average Precision (MAP)** results for the 17-Flowers dataset. The evaluation shows that **DenseNet-201** achieved the highest MAP, reaching approximately **96%**. **Inception v3** followed with a MAP of around **86%**, while **GoogleNet** recorded the lowest performance with a MAP of about **70%**.

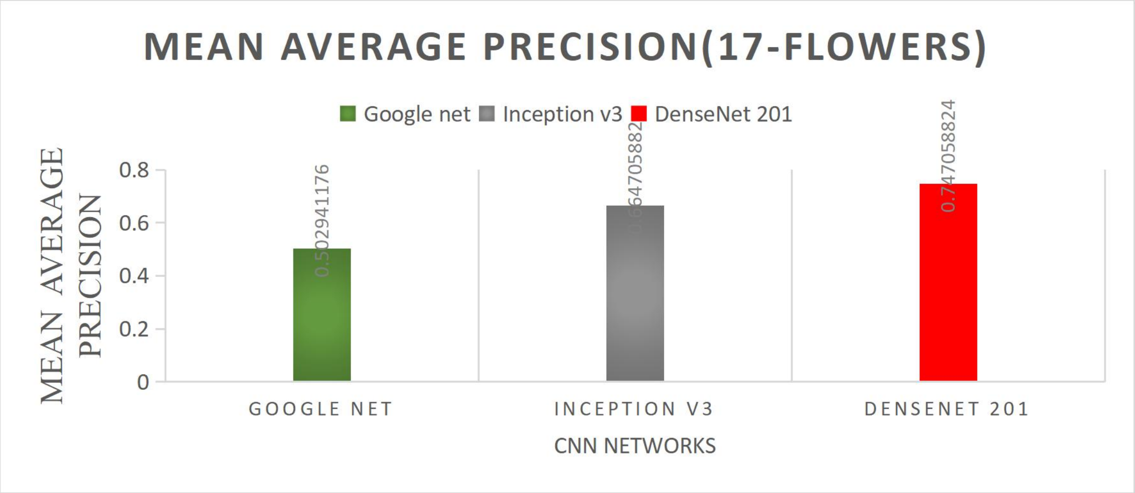

Figure 9: Mean Average Precision Rate For 17-Flowers Dataset.

Figure 10 illustrates a line graph depicting the **average retrieval recall** performance across various categories of the 17-Flowers dataset. The recall values range between **60% and 90%**, indicating moderate to high retrieval effectiveness. The highest recall—approximately **96%**—was achieved using **GoogleNet**, with the **daisy** category performing the best. **DenseNet-201** demonstrated a strong recall for the **bluebell** category, exceeding **85%**, while **Inception v3** achieved its peak recall with the **daffodil** category. However, Inception v3 reported the **lowest recall for bluebell**. The graph also reveals that **DenseNet-201's overall recall is relatively low**, which implies it offers **higher precision** in retrieval tasks.

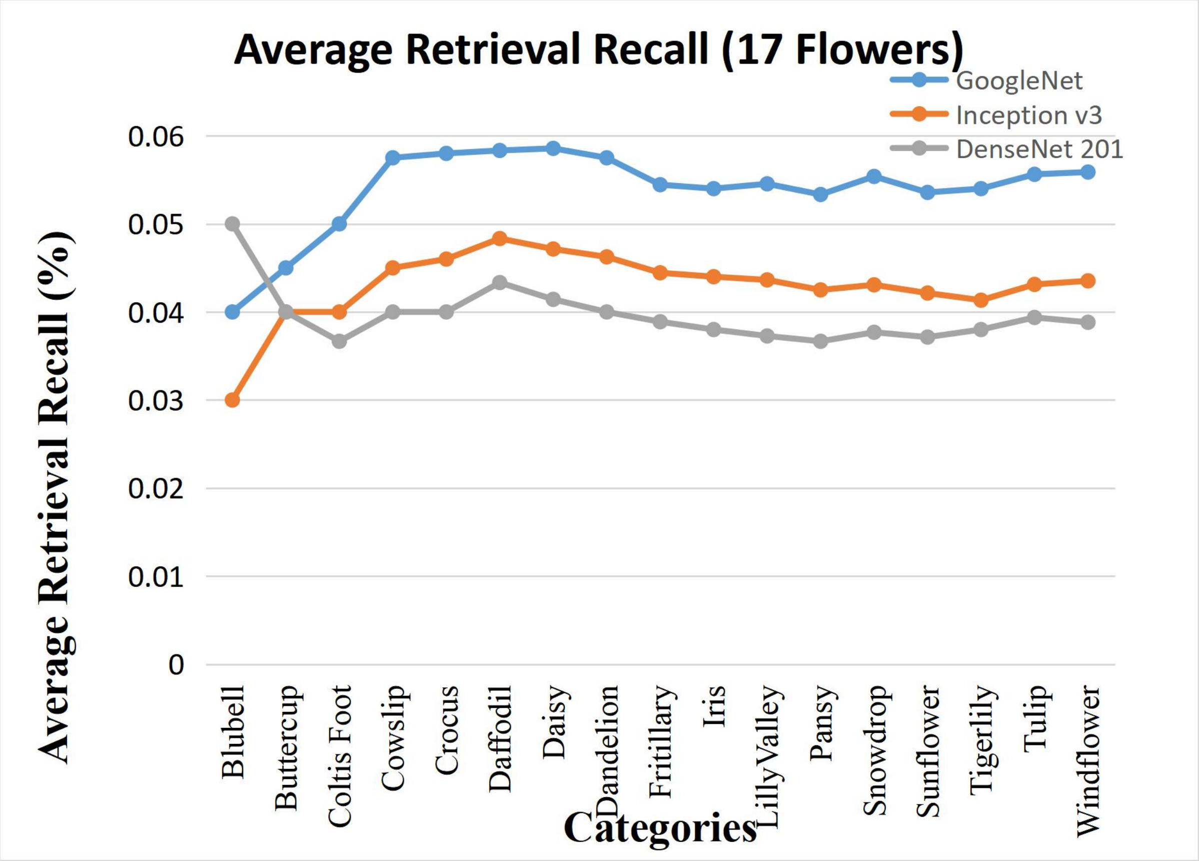

Figure 10: Average Retrieval Recall rate for categories of 17-flowers dataset.

The graphical analysis of the 17-Flowers dataset shows that GoogleNet achieves the highest mean average recall (MAR), while DenseNet-201 records the lowest MAR among the evaluated models.

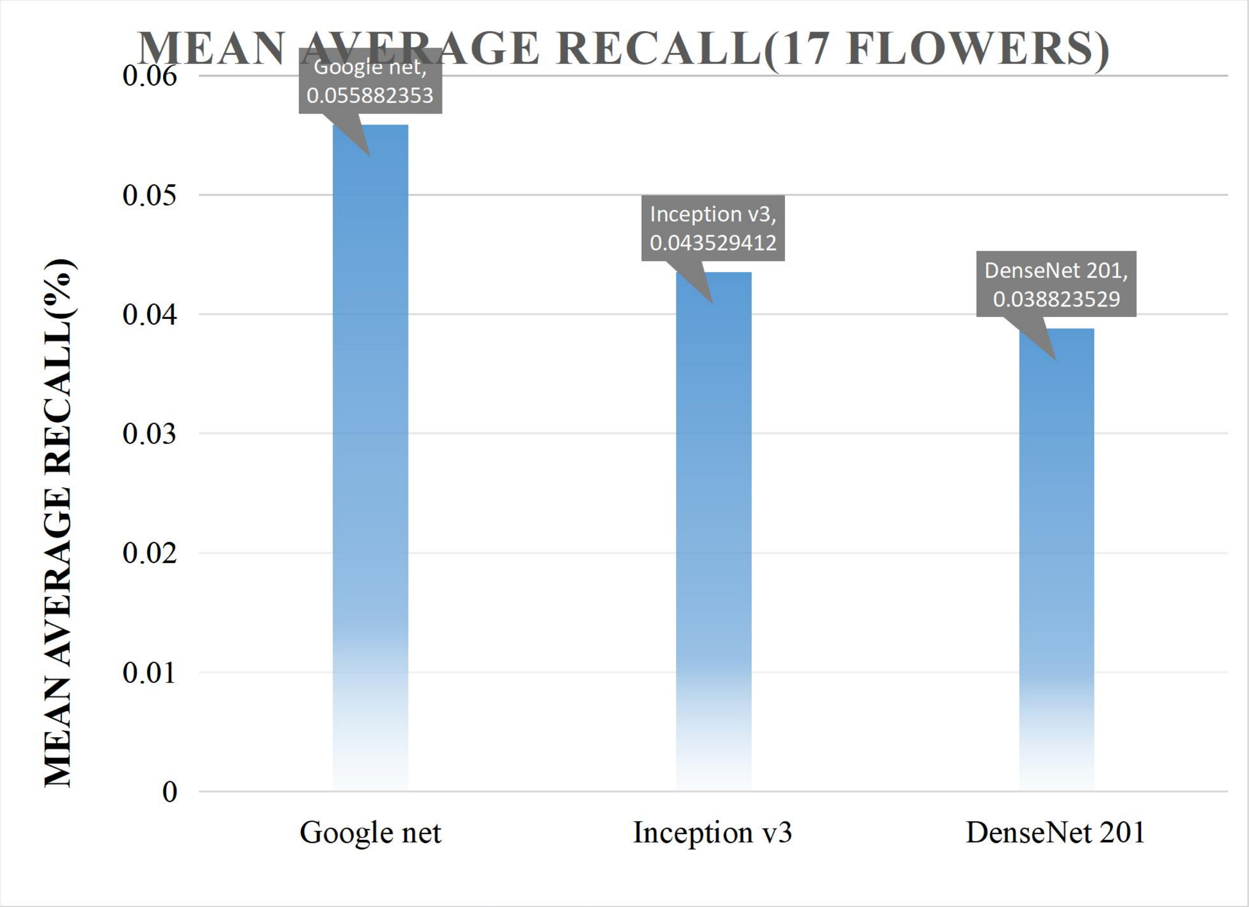

Figure 11: Mean Average Recall Rate For Categories Of 17-Flowers Dataset.

Table 2 presents the average precision and recall values for various image categories in the CIFAR-10 dataset. The results show that categories such as **ship, horse, and dog achieved up to 90% average precision**, while several others exceeded 85% **precision**. The CIFAR-10 dataset, used for this evaluation, consists of **60,000 color images** across **10 distinct classes** with consistent image dimensions. The categories include **automobile, bird, cat, deer, dog, frog, horse, ship, and truck**.

Table 2: Value of precision and recall represented in tabular form for 10 categories of cifar-10 dataset.

Category	GoogleNet			Inception v3			DenseNet-201		
	Top-20 Images	Precision	Recall	Top-20 Images	Precision	Recall	Top-20 Images	Precision	Recall
Automobile	20	1.00	0.23	20	1.00	0.32	20	1.00	0.15
Bird	20	1.00	0.23	20	1.00	0.32	20	1.00	0.20

Cat	13	0.25	0.03	16	0.80	0.31	20	1.00	0.25
Deer	17	0.20	0.23	17	0.85	0.30	20	1.00	0.35
Dog	18	0.21	0.31	20	1.00	0.23	20	1.00	0.30
Frog	17	0.20	0.23	19	0.95	0.32	20	1.00	0.20
Horse	14	0.40	0.041	20	1.00				

The experimental outcomes were visualized using bar graphs to facilitate a clearer understanding of the results. Deep learning-based feature extraction techniques were applied, using Convolutional Neural Networks (CNNs) for sampling and scaling across the CIFAR-10 dataset, which includes categories such as automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. As illustrated in Figure 12, the average precision for each class was analyzed. DenseNet-201 demonstrated the highest precision among the models, followed by Inception v3, while GoogleNet showed the comparatively lowest performance in terms of precision.

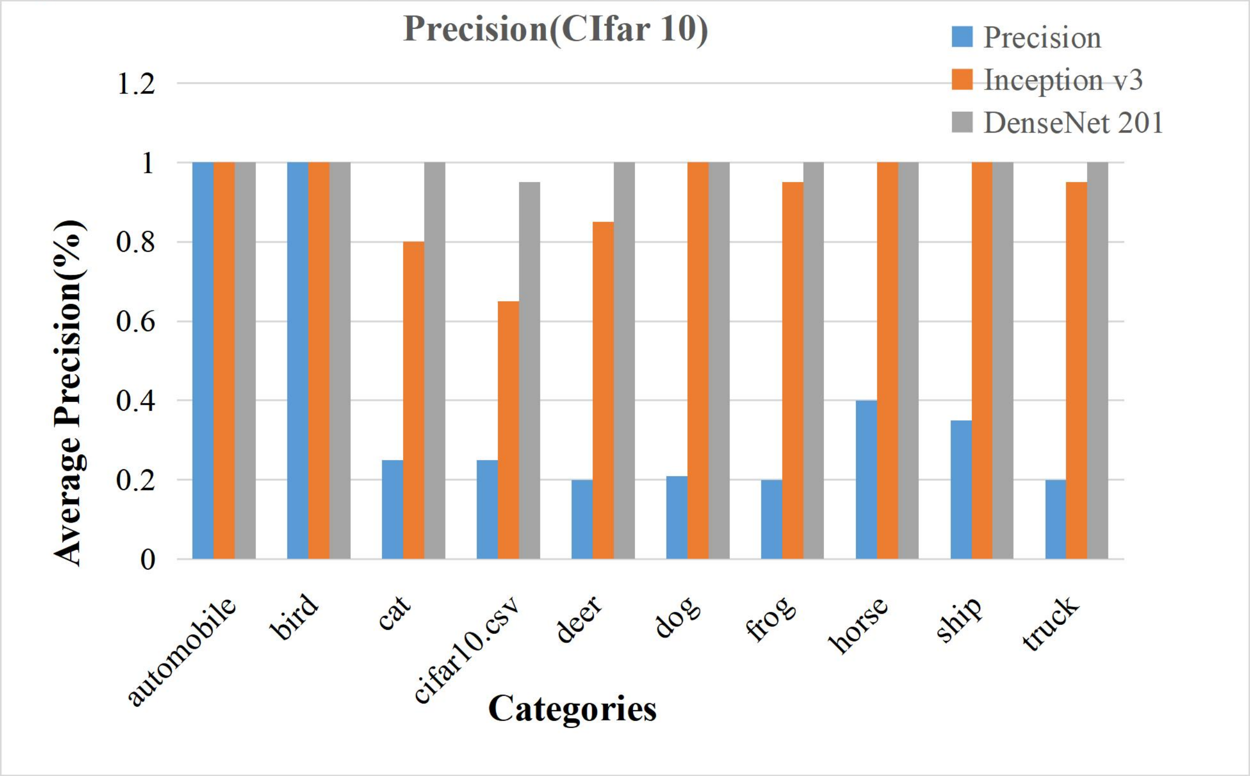

Figure 12: Cifar-10 Dataset Average Precision Rate

Figure 13 presents the average recall values for the ten classes in the CIFAR-10 dataset, analyzed using GoogleNet, Inception v3, and DenseNet-201. The bar graph illustrates variations across different categories, aiding in understanding the recall performance of each class. GoogleNet achieved notably high recall in categories such as ship, cat, CIFAR10.CSV, and horse—though these were still under 50%. For categories like automobile, bird, deer, frog, dog, and truck, the recall exceeded 70%

with GoogleNet's CNN-based feature extraction. DenseNet-201, on the other hand, showed lower recall across most classes, often below 30%, though approximately half the categories reached up to 60%. Inception v3 produced recall values generally ranging from 80% to 90%, with an overall average near 50%. The results are visually summarized in Figure 13.

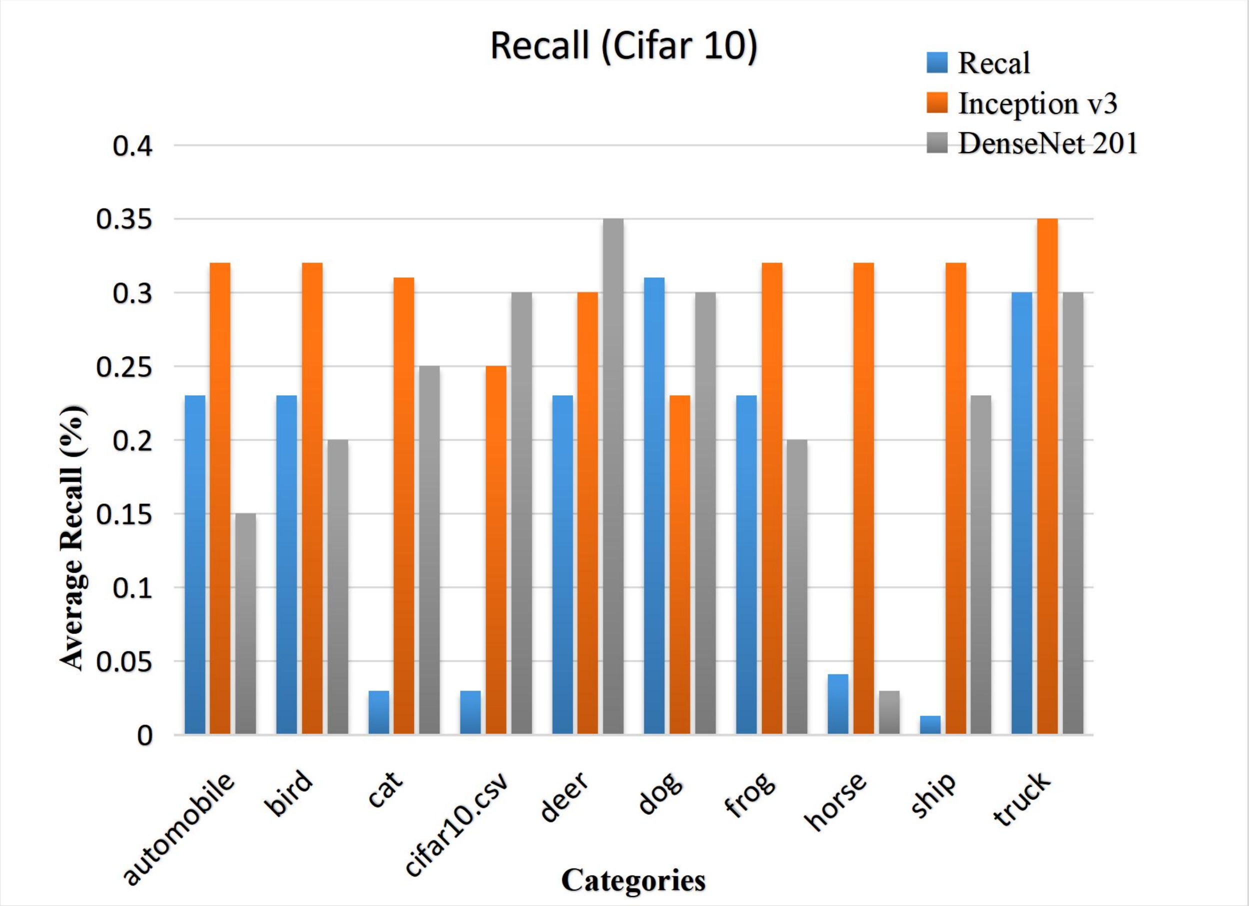

Figure 13: Average Recall rate for 10 categories in Cifar-10 dataset.

As shown in Figure 14, the highest average retrieval precision across most CIFAR-10 categories is achieved using the DenseNet-201 feature extractor, reaching nearly 99%. Specifically, the automobile and bird categories show particularly strong performance, with DenseNet-201 delivering average retrieval precision rates exceeding 80%.

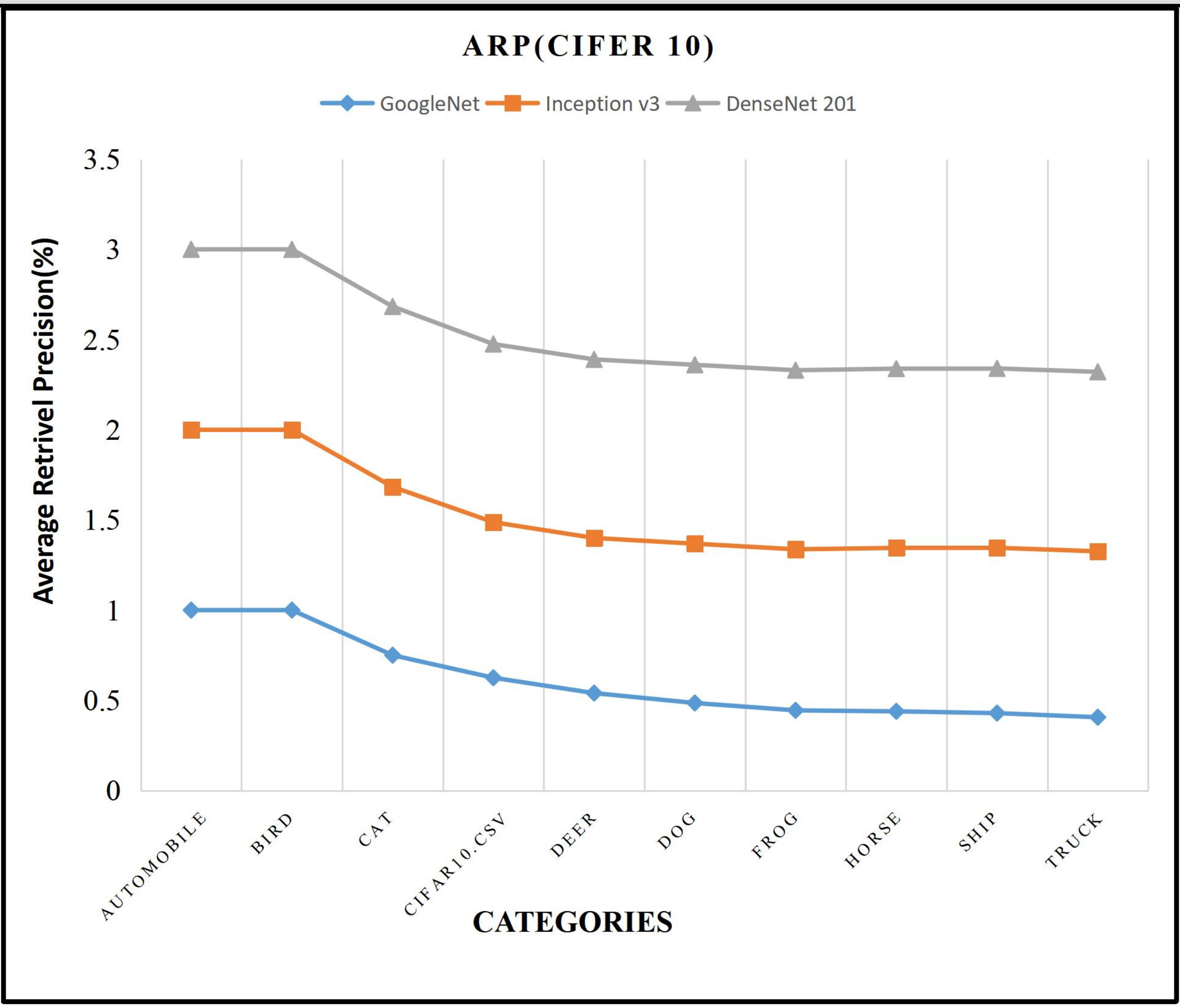

Figure 14: Cifar-10 Dataset Average Retrieval Precision Rate

Figure 15 illustrates the mean average precision (MAP) results for the CIFAR-10 dataset. DenseNet-201 achieves the highest MAP at 100%, followed by Inception v3 with 92%. In comparison, GoogleNet demonstrates a relatively lower MAP performance than the other two CNN architectures.

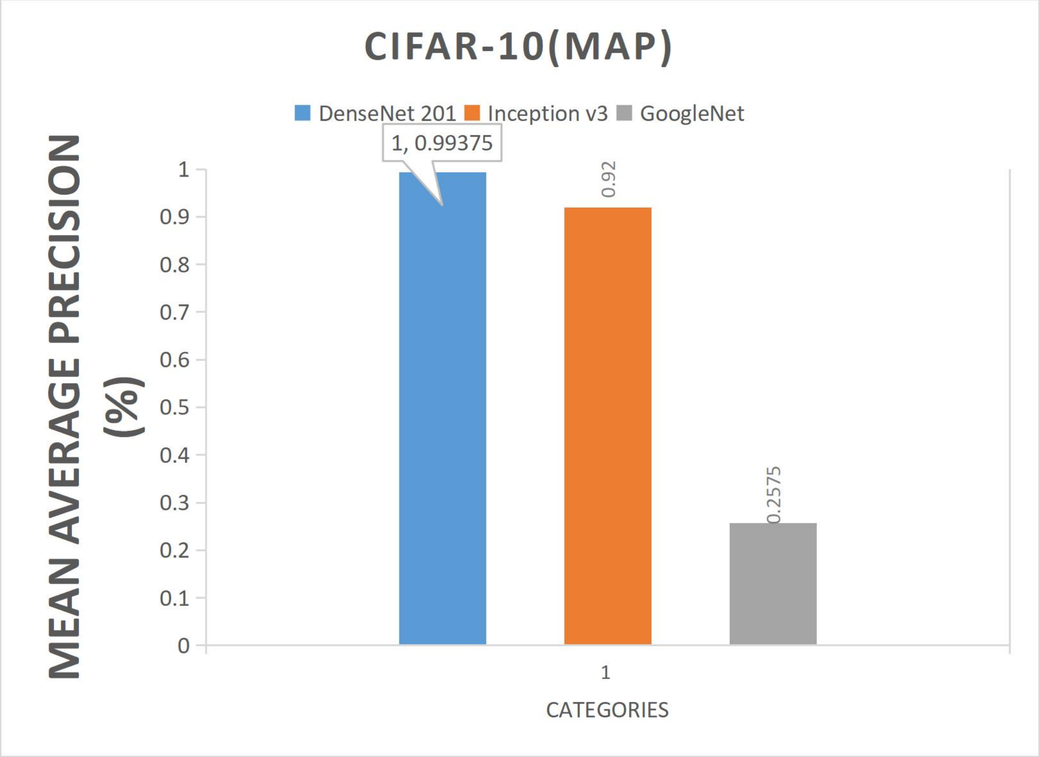

Figure 15: Cifar-10 Mean Average Precision.

Figure 16 presents the average retrieval recall across various CIFAR-10 categories. DenseNet-201 achieves the highest recall, reaching approximately 95% for most classes, with the frog category showing the best performance. Inception v3 demonstrates its highest recall for the bird category, exceeding 75%, while GoogleNet also records its peak recall for the bird class.

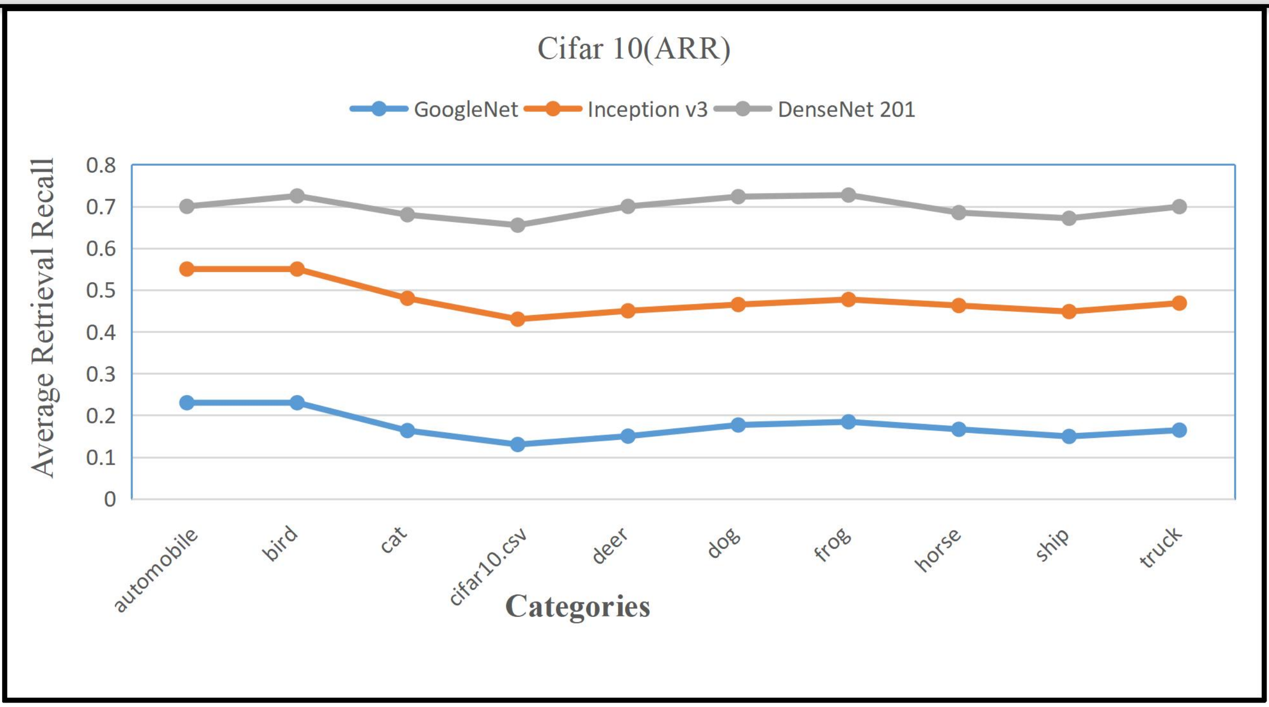

Figure 16: Average Retrieval Recall Rate For Cifar-10 Dataset

Figure 17 illustrates the mean average recall (MAR) performance across the CIFAR-10 dataset. DenseNet-201 achieved the highest MAR at 92%, followed by Inception v3 with 62%, while GoogleNet recorded the lowest recall performance at 42%.

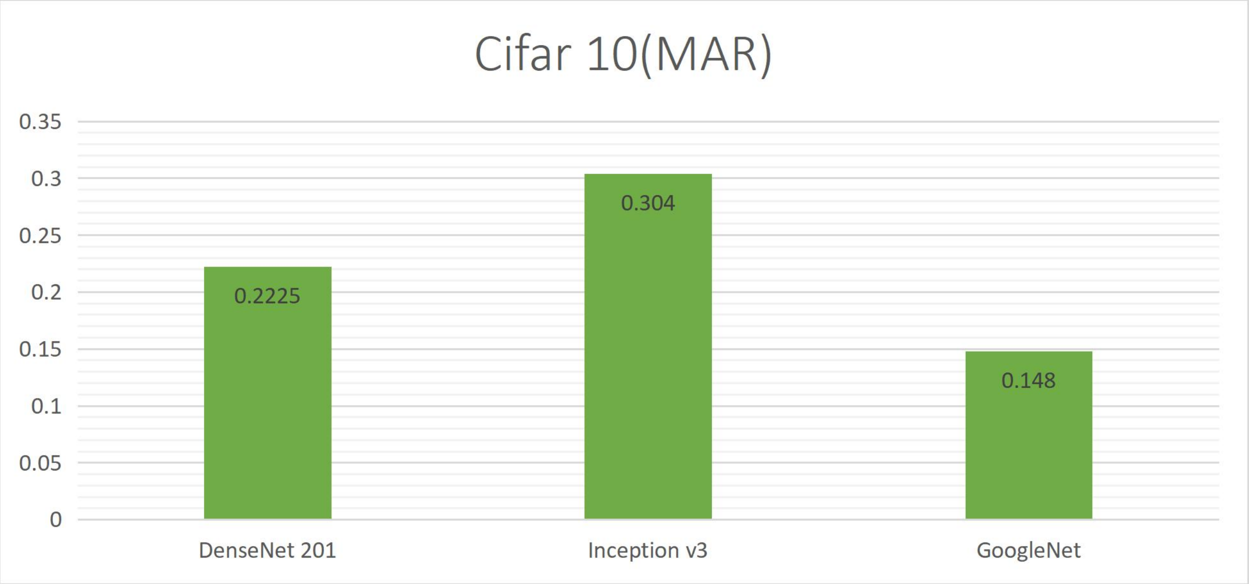

Figure 17: Cifar-10 Mean Average Recall Rate.

An additional experiment was conducted using the widely known Corel-1K dataset, which includes 1,000 images distributed evenly across 10 distinct categories: beach, buildings, buses, corel-1k, dinosaurs, elephants, flowers, horses, food, and mountains. Each category contains approximately 100 images of similar type. The evaluation focused on measuring average precision and recall using different convolutional neural networks (CNNs), including GoogleNet, Inception v3, and DenseNet-201. Among these, DenseNet-201 delivered the most effective performance, achieving 100% similarity in nearly half of the categories. Another 40% of the categories also showed perfect similarity. In contrast, GoogleNet demonstrated the lowest similarity results. The high efficiency of DenseNet-201 is attributed to its strong performance in categories such as dinosaurs, elephants, and horses—all of which are animal-based classes sharing similar visual features like four-legged structure and body shape.

Table 3: Precision and Recall represented in tabular form for Corel-1000 dataset.

Category	GoogleNet (Top 20, Precision, Recall)	Inception v3 (Top 20, Precision, Recall)	DenseNet-201 (Top 20, Precision, Recall)
Beach	18, 0.90, 0.03	20, 1.00, 0.03	18, 0.90, 0.03
Buildings	12, 0.60, 0.04	15, 0.75, 0.03	12, 0.60, 0.04
Buses	14, 0.70, 0.04	13, 0.65, 0.04	16, 0.80, 0.03
Corel_1k.csv	12, 0.60, 0.04	19, 0.95, 0.03	19, 0.95, 0.03
Dinosaurs	20, 1.00, 0.03	20, 1.00, 0.03	20, 1.00, 0.03
Elephants	14, 0.70, 0.04	17, 0.85, 0.03	20, 1.00, 0.03
Flowers	20, 1.00, 0.03	20, 1.00, 0.03	20, 1.00, 0.03
Food	18, 0.90, 0.03	17, 0.85, 0.03	20, 1.00, 0.03
Horses	20, 1.00, 0.03	20, 1.00, 0.03	20, 1.00, 0.03
Mountains	13, 0.65, 0.04	14, 0.70, 0.04	15, 0.75, 0.03

Figure 18 illustrates the average precision scores for various categories within the Corel_1K dataset using a bar chart. This visual representation effectively highlights the classification accuracy achieved through deep learning techniques. Feature extraction was carried out using CNN models, supported by image scaling and filtering methods to enhance mapping and integration. DenseNet-201 demonstrated exceptional performance, achieving perfect (100%) precision in several categories including *dinosaurs*, *flowers*, *elephants*, *food*, and *horses*. Categories like *beach* and *buses* also achieved high precision rates of 90% and 95%, respectively, using the same network. Remaining categories mostly recorded average precision rates of around 70%. Conversely, GoogleNet yielded the lowest average precision values, though still above 60% for all categories.

Overall, more than half of the dataset categories achieved over 70% precision, confirming the effectiveness of DenseNet-201 in feature-based image classification for the Corel_1K dataset.

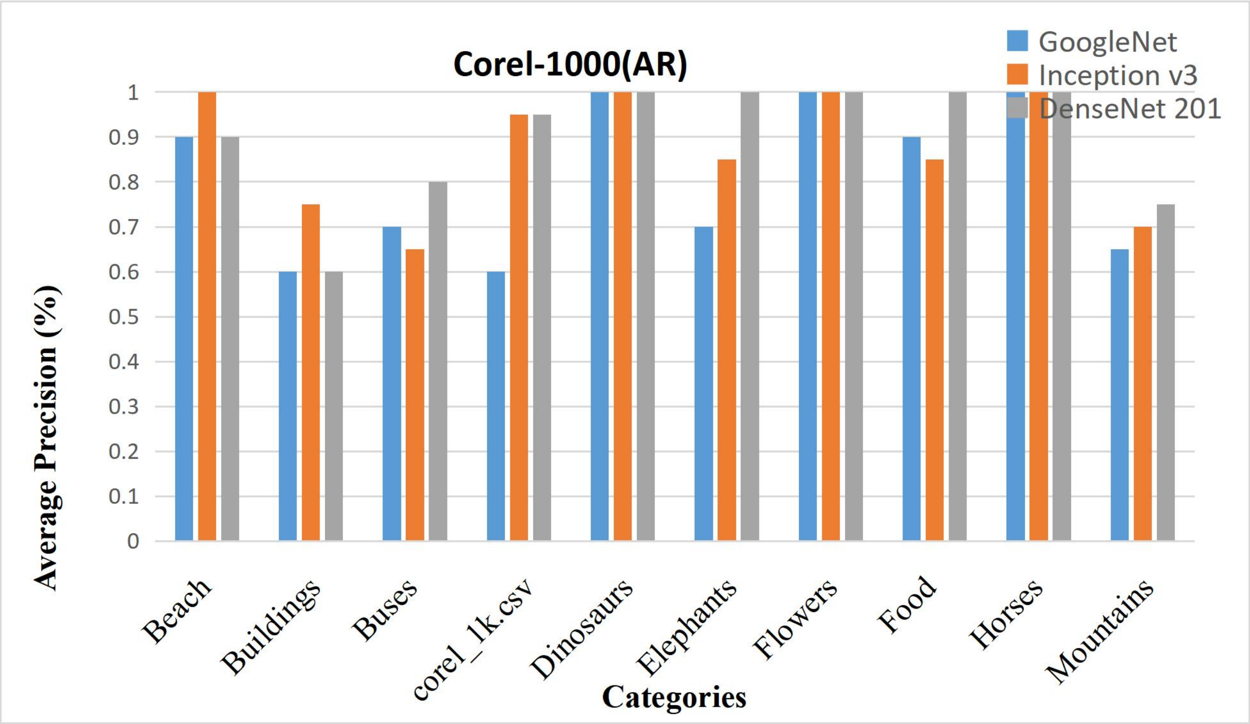

Figure 18: Average precision rate for 10 categories in Corel_1k dataset.

Figure 19 presents the average recall values for the ten categories within the Corel-1K dataset, evaluated using three convolutional neural network models: GoogleNet, Inception v3, and DenseNet-201. The bar graph illustrates category-wise variations in recall, highlighting the effectiveness of each model across different classes. GoogleNet achieved notably high recall rates for *buildings*, *elephants*, and *mountains*, each approaching approximately 75%. However, the majority of categories using this model exhibited recall values at or below 50%. In contrast, Inception v3 demonstrated superior performance in categories such as *beach*, *dinosaurs*, *flowers*, and *mountains*, with recall values reaching or exceeding 80%. DenseNet-201 also produced competitive results, especially in categories like *buses*, *dinosaurs*, *elephants*, *flowers*, and *horses*, many of which achieved recall scores near or above 50%. Inception v3 consistently delivered recall rates between 80% and 90% for several categories, while the overall average recall across models remained around the 50% mark. These results, visualized in Figure 4.16, offer insight into each model's retrieval performance across the Corel-1K dataset.

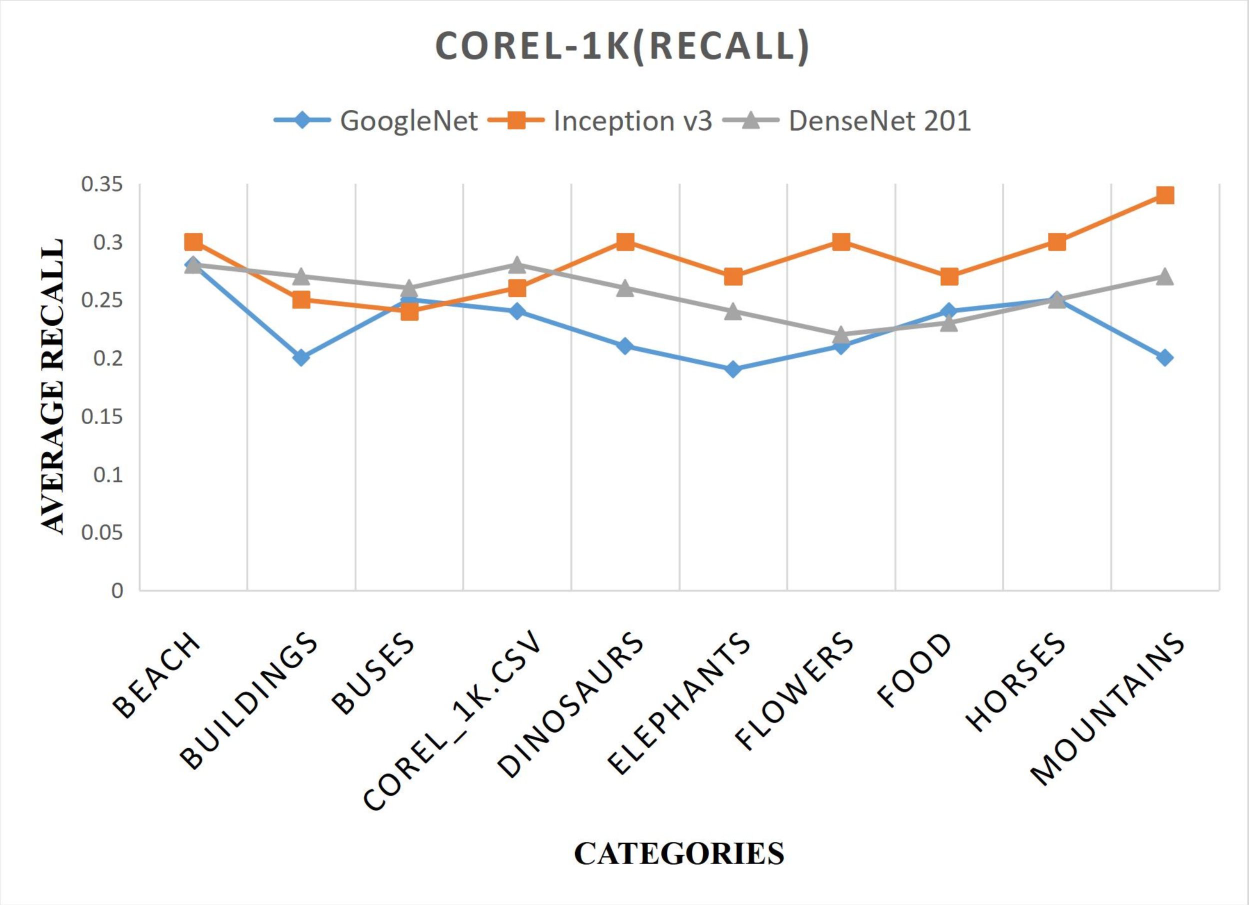

Figure 19: Average Recall rate for 10 categories in corel-1K dataset

Figure 20 illustrates the average retrieval precision (ARP) across various categories of the Corel-1K dataset using different CNN-based feature extractors. Inception v3 demonstrated the highest ARP, reaching approximately 99% for the majority of categories. Notably, the *beach* category achieved over 80% ARP with Inception v3, indicating its strong feature representation capability. The *buses* category showed the lowest ARP within the Inception v3 results, although it still maintained an 80% precision rate. When comparing GoogleNet and DenseNet-201, both models yielded similar ARP values for *beach* and *buildings*. Additionally, feature extraction for the *elephant* category produced equivalent ARP scores using both DenseNet-201 and Inception v3, highlighting their comparable performance for certain classes.

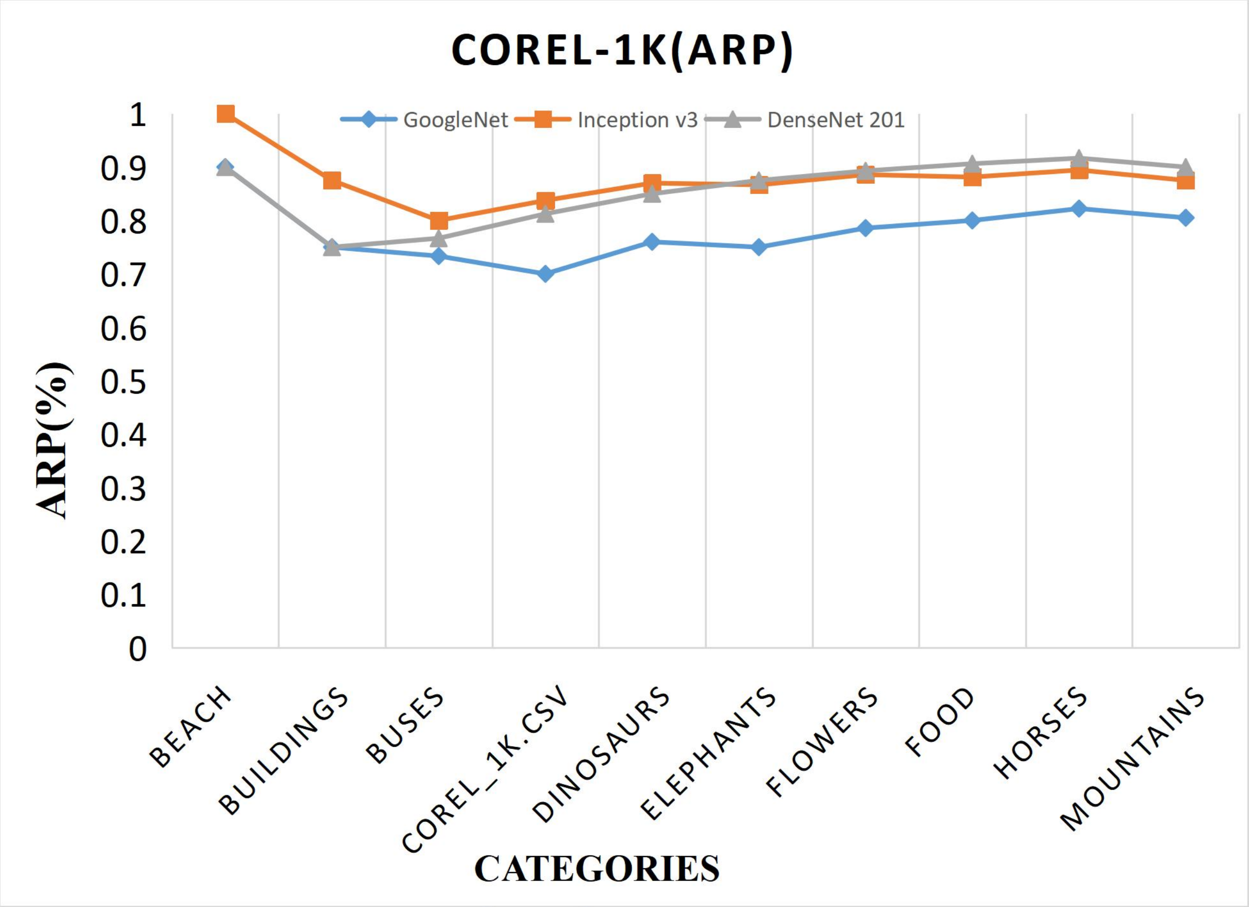

Figure 20: Average Retrieval Precision rate for 10 categories in Corel-1K dataset

Figure 21 presents a line graph illustrating the efficiency of average retrieval recall across various categories within the Corel-1000 dataset. The recall values generally range between 60% and 90%, indicating strong retrieval performance overall. The highest recall is observed for the *buses* category, particularly when using DenseNet-201 as the feature extractor, achieving over 90% retrieval recall. Similarly, Inception v3 demonstrates strong performance for the *mountains* category. Conversely, the lowest recall using Inception v3 is found in the *beach* category. The graph also reveals that GoogleNet exhibits consistently lower recall rates, which suggests that while fewer relevant items are retrieved, its precision may be relatively higher.

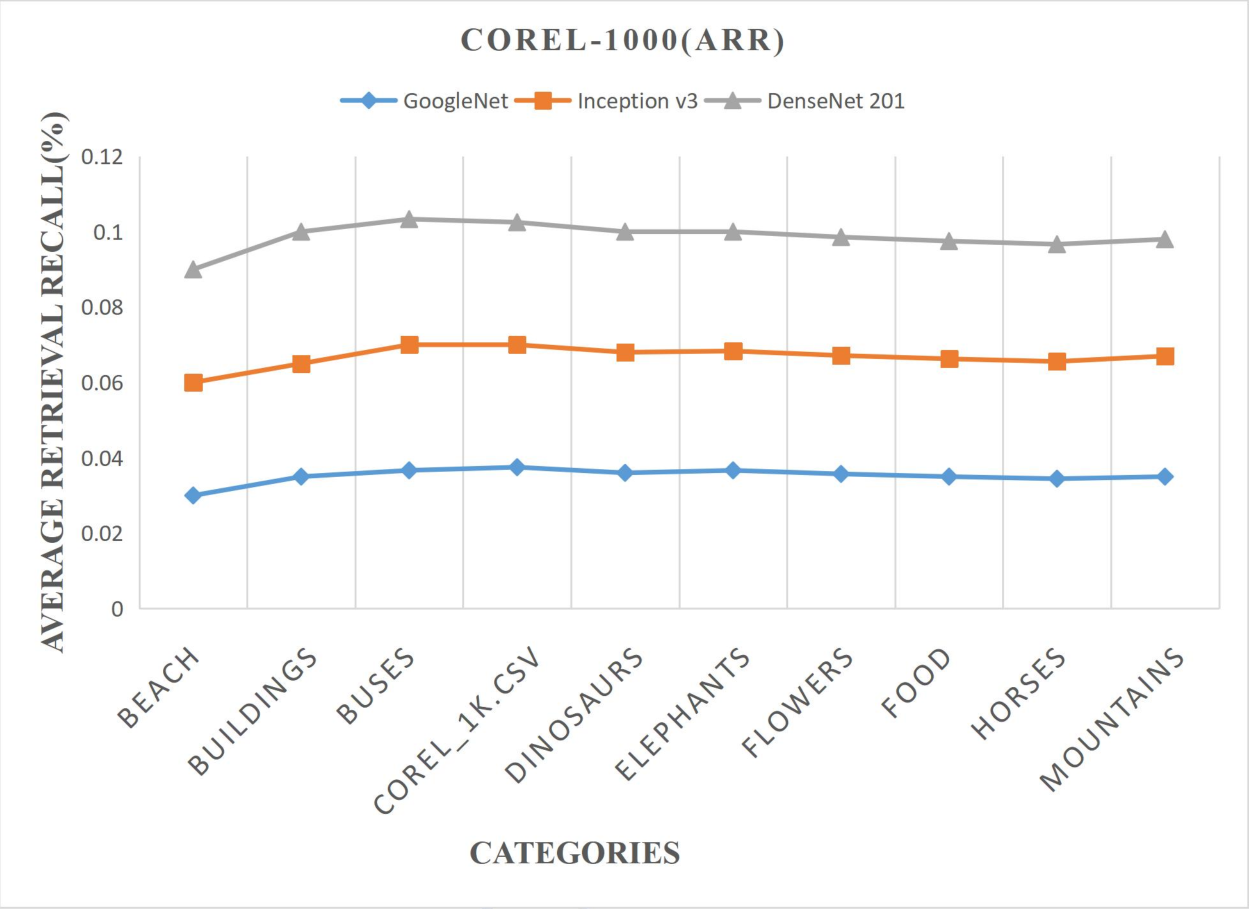

Figure 21: Average Retrieval Recall rate for 10 categories in Corel-1K dataset

Figure 21: Corel-1K dataset showing different sample images of categories

Figure 22 displays the mean average precision (mAP) results for the Corel-1000 dataset. DenseNet-201 achieved the highest mAP of 90%, followed by Inception v3 with 87%, while GoogleNet recorded the lowest at 80%.

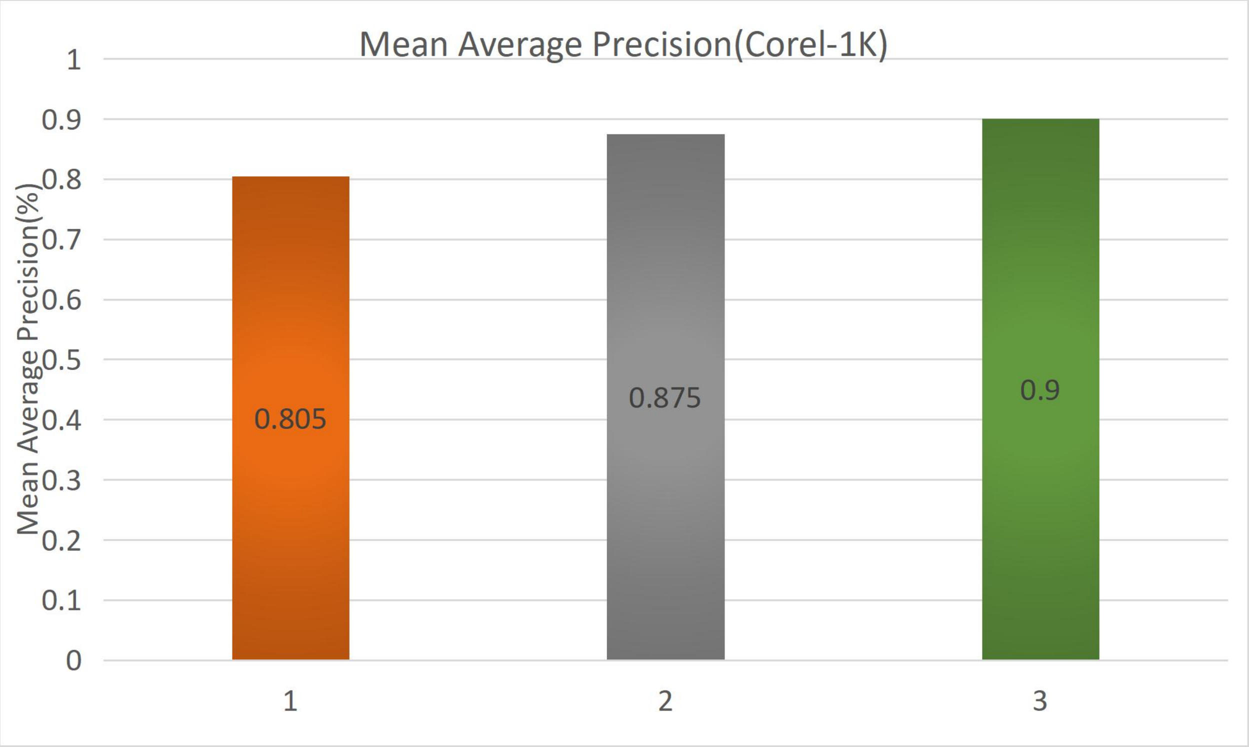

Figure 22: Corel-1K dataset Mean Average Precision rate

In figure 23 mean average recall is observed that is between 0.2 and 0,3 which reports 85% mean average precision using DenseNet-201, 95% using inception v3 and 75% using GoogleNet for cifar-10.

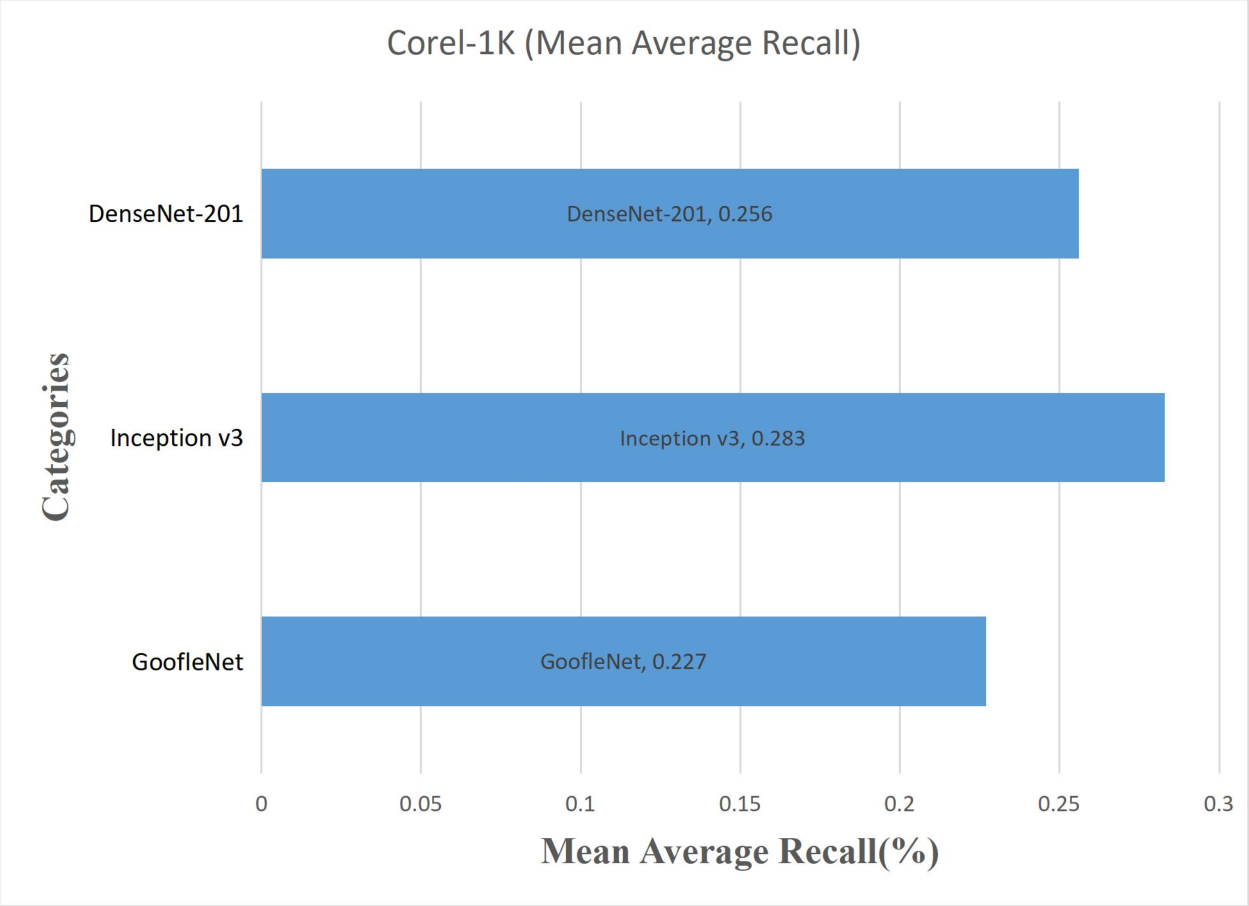

Figure 23: Mean Average Recall rate for 10 categories in Corel-1K dataset

To assess the effectiveness of the proposed method, the Zubud dataset was utilized. Random samples were selected to evaluate performance metrics such as precision, recall, average retrieval precision (ARP), average retrieval recall (ARR), mean average precision (MAP), mean average recall (MAR), and F-measure. The dataset contains various image groups labeled from *object001* to *object200*, with each class containing five images—totaling 1000 images. From these, 20 categories were selected for analysis. The images vary in dimensions, and within each selected category, visual features such as shape, color, and object structure were examined. The proposed approach demonstrated strong average precision values across most Zubud categories. A figure illustrates sample images from different groups within the Zubud dataset.

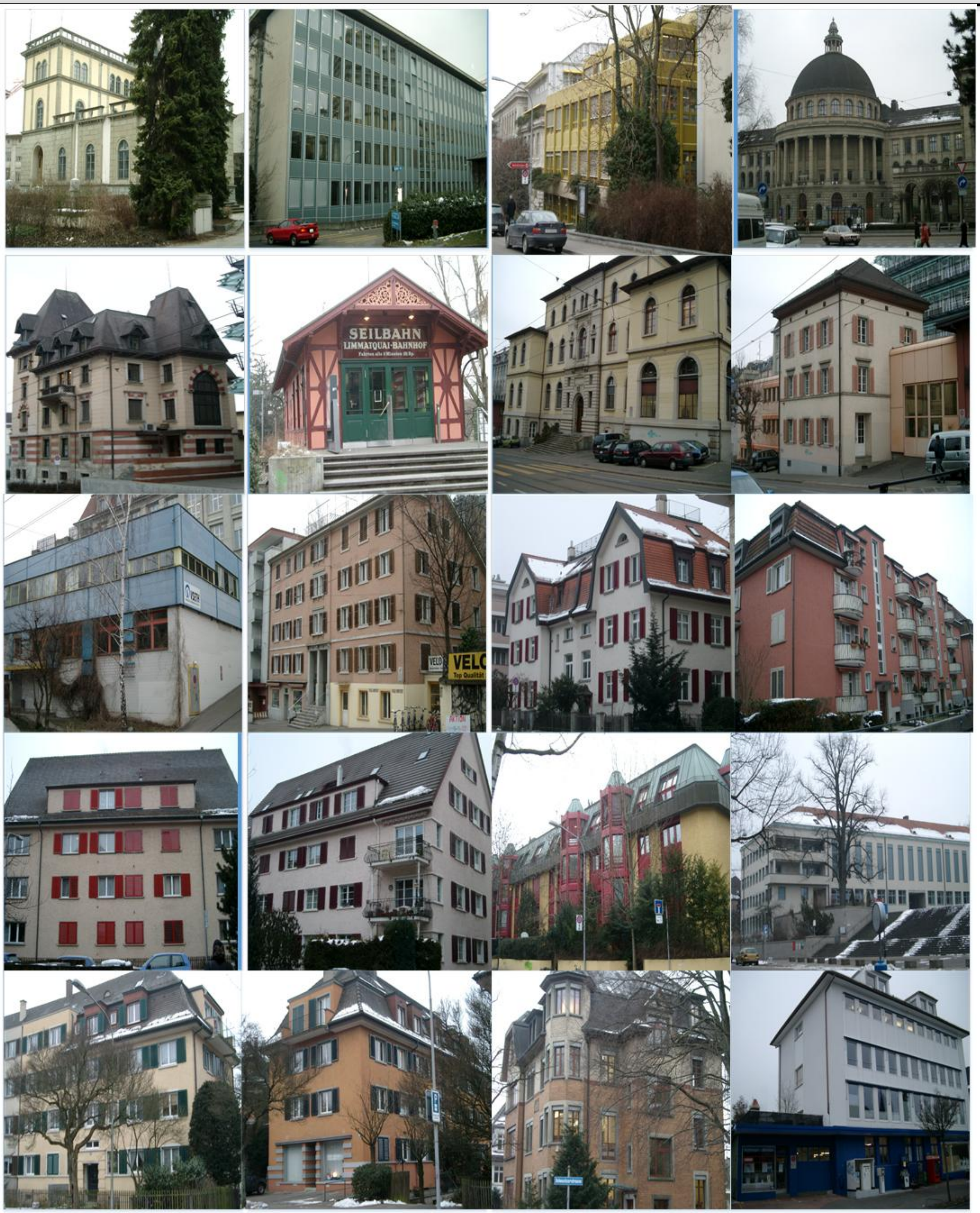

Figure 24: Zubud dataset showing different sample images of categories

Table 4: Value of precision and recall represented in tabular form for 200 categories of Zubud dataset.

Zubud									
GoogleNet			Inception v3				DenseNet 201		
Category	20	Precision	Recall	20	Precision	Recall	20	Precision	Recall
object0002	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0003	2	0.1	0.25	4	0.2	0.13	2	0.1	0.25
object0004	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0005	4	0.2	0.13	1	0.05	0.5	4	0.2	0.13
object0006	4	0.2	0.13	5	0.25	0.1	1	0.05	0.5
object0007	3	0.15	0.17	4	0.2	0.13	2	0.1	0.25
object0008	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0009	5	0.25	0.1	2	0.1	0.25	2	0.1	0.25
object0010	2	0.1	0.25	4	0.2	0.13	3	0.15	0.17
object0011	5	0.25	0.1	3	0.15	0.17	4	0.2	0.13
object0012	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0013	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0014	2	0.1	0.25	3	0.15	0.17	2	0.1	0.25
object0015	5	0.25	0.1	1	0.05	0.5	3	0.15	0.17
object0016	3	0.15	0.17	3	0.15	0.17	3	0.15	0.17
object0017	4	0.2	0.13	4	0.2	0.13	3	0.15	0.17
object0018	5	0.25	0.1	3	0.15	0.17	2	0.1	0.25
object0019	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0020	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1

object0021	2	0.1	0.25	3	0.15	0.17	2	0.1	0.25
object0022	5	0.25	0.1	5	0.25	0.1	2	0.1	0.25
object0023	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0024	2	0.1	0.25	5	0.25	0.1	2	0.1	0.25
object0025	4	0.2	0.13	5	0.25	0.1	2	0.1	0.25
object0026	4	0.2	0.13	5	0.25	0.1	2	0.1	0.25
object0027	3	0.15	0.17	4	0.2	0.13	1	0.05	0.5
object0028	1	0.05	0.5	4	0.2	0.13	1	0.05	0.5
object0029	4	0.2	0.13	4	0.2	0.13	4	0.2	0.13
object0030	1	0.05	0.5	3	0.15	0.17	3	0.15	0.17
object0031	2	0.1	0.25	5	0.25	0.1	4	0.2	0.13
object0032	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0033	3	0.15	0.17	5	0.25	0.1	3	0.15	0.17
object0034	4	0.2	0.13	2	0.1	0.25	4	0.2	0.13
object0035	1	0.05	0.5	4	0.2	0.13	4	0.2	0.13
object0036	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0037	1	0.05	0.5	4	0.2	0.13	4	0.2	0.13
object0038	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0039	3	0.15	0.17	2	0.1	0.25	3	0.15	0.17
object0040	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0041	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0042	2	0.1	0.25	3	0.15	0.17	2	0.1	0.25
object0043	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0044	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13

object0045	4	0.2	0.13	3	0.15	0.17	2	0.1	0.25
object0046	3	0.15	0.17	5	0.25	0.1	4	0.2	0.13
object0047	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0048	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0049	1	0.05	0.5	3	0.15	0.17	3	0.15	0.17
object0050	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0051	2	0.1	0.25	5	0.25	0.1	3	0.15	0.17
object0052	3	0.15	0.17	5	0.25	0.1	3	0.15	0.17
object0053	5	0.25	0.1	4	0.2	0.13	2	0.1	0.25
object0054	1	0.05	0.5	5	0.25	0.1	2	0.1	0.25
object0055	2	0.1	0.25	2	0.1	0.25	2	0.1	0.25
object0056	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0057	3	0.15	0.17	4	0.2	0.13	3	0.15	0.17
object0058	3	0.15	0.17	4	0.2	0.13	4	0.2	0.13
object0059	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0060	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0061	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0062	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0063	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0064	2	0.1	0.25	4	0.2	0.13	4	0.2	0.13
object0065	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0066	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0067	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0068	1	0.05	0.5	3	0.15	0.17	1	0.05	0.5

object0069	4	0.2	0.13	4	0.2	0.13	4	0.2	0.13
object0070	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0071	3	0.15	0.17	4	0.2	0.13	4	0.2	0.13
object0072	4	0.2	0.13	4	0.2	0.13	5	0.25	0.1
object0073	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0074	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0075	2	0.1	0.25	5	0.25	0.1	4	0.2	0.13
object0076	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0077	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0078	4	0.2	0.13	4	0.2	0.13	5	0.25	0.1
object0079	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0080	3	0.15	0.17	3	0.15	0.17	2	0.1	0.25
object0081	3	0.15	0.17	5	0.25	0.1	4	0.2	0.13
object0082	3	0.15	0.17	5	0.25	0.1	4	0.2	0.13
object0083	3	0.15	0.17	3	0.15	0.17	3	0.15	0.17
object0084	3	0.15	0.17	5	0.25	0.1	1	0.05	0.5
object0085	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0086	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0087	1	0.05	0.5	5	0.25	0.1	5	0.25	0.1
object0088	5	0.25	0.1	3	0.15	0.17	1	0.05	0.5
object0089	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0090	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0091	2	0.1	0.25	2	0.1	0.25	2	0.1	0.25
object0092	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17

object0093	5	0.25	0.1	5	0.25	0.1	1	0.05	0.5
object0094	3	0.15	0.17	2	0.1	0.25	2	0.1	0.25
object0095	3	0.15	0.17	4	0.2	0.13	1	0.05	0.5
object0096	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0097	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0098	2	0.1	0.25	5	0.25	0.1	5	0.25	0.1
object0099	5	0.25	0.1	3	0.15	0.17	3	0.15	0.17
object0100	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0101	4	0.2	0.13	5	0.25	0.1	4	0.2	0.13
object0102	2	0.1	0.25	3	0.15	0.17	3	0.15	0.17
object0103	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0104	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0105	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0106	3	0.15	0.17	5	0.25	0.1	3	0.15	0.17
object0107	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0108	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0109	3	0.15	0.17	3	0.15	0.17	3	0.15	0.17
object0110	5	0.25	0.1	5	0.25	0.1	2	0.1	0.25
object0111	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0112	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0113	2	0.1	0.25	2	0.1	0.25	2	0.1	0.25
object0114	5	0.25	0.1	4	0.2	0.13	2	0.1	0.25
object0115	2	0.1	0.25	3	0.15	0.17	1	0.05	0.5
object0116	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17

object0117	3	0.15	0.17	1	0.05	0.5	3	0.15	0.17
object0118	4	0.2	0.13	5	0.25	0.1	4	0.2	0.13
object0119	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0120	5	0.25	0.1	4	0.2	0.13	1	0.05	0.5
object0121	3	0.15	0.17	4	0.2	0.13	3	0.15	0.17
object0122	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0123	4	0.2	0.13	5	0.25	0.1	1	0.05	0.5
object0124	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0125	1	0.05	0.5	2	0.1	0.25	1	0.05	0.5
object0126	5	0.25	0.1	3	0.15	0.17	1	0.05	0.5
object0127	1	0.05	0.5	5	0.25	0.1	4	0.2	0.13
object0128	5	0.25	0.1	5	0.25	0.1	2	0.1	0.25
object0129	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0130	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0131	3	0.15	0.17	4	0.2	0.13	3	0.15	0.17
object0132	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0133	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0134	3	0.15	0.17	5	0.25	0.1	1	0.05	0.5
object0135	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0136	1	0.05	0.5	5	0.25	0.1	5	0.25	0.1
object0137	3	0.15	0.17	5	0.25	0.1	4	0.2	0.13
object0138	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0139	2	0.1	0.25	3	0.15	0.17	2	0.1	0.25
object0140	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1

object0141	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0142	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0143	2	0.1	0.25	5	0.25	0.1	5	0.25	0.1
object0144	3	0.15	0.17	3	0.15	0.17	3	0.15	0.17
object0145	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0146	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0147	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0148	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0149	5	0.25	0.1	5	0.25	0.1	1	0.05	0.5
object0150	5	0.25	0.1	4	0.2	0.13	1	0.05	0.5
object0151	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0152	5	0.25	0.1	5	0.25	0.1	1	0.05	0.5
object0153	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0154	3	0.15	0.17	1	0.05	0.5	4	0.2	0.13
object0155	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0156	1	0.05	0.5	3	0.15	0.17	5	0.25	0.1
object0157	2	0.1	0.25	3	0.15	0.17	1	0.05	0.5
object0158	5	0.25	0.1	4	0.2	0.13	4	0.2	0.13
object0159	2	0.1	0.25	5	0.25	0.1	4	0.2	0.13
object0160	4	0.2	0.13	4	0.2	0.13	2	0.1	0.25
object0161	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0162	3	0.15	0.17	1	0.05	0.5	3	0.15	0.17
object0163	4	0.2	0.13	4	0.2	0.13	4	0.2	0.13
object0164	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1

object0165	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0166	4	0.2	0.13	5	0.25	0.1	1	0.05	0.5
object0167	2	0.1	0.25	5	0.25	0.1	2	0.1	0.25
object0168	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0169	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0170	2	0.1	0.25	5	0.25	0.1	5	0.25	0.1
object0171	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0172	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0173	5	0.25	0.1	5	0.25	0.1	3	0.15	0.17
object0174	4	0.2	0.13	5	0.25	0.1	2	0.1	0.25
object0175	4	0.2	0.13	4	0.2	0.13	3	0.15	0.17
object0176	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0177	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0178	3	0.15	0.17	5	0.25	0.1	3	0.15	0.17
object0179	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0180	2	0.1	0.25	4	0.2	0.13	3	0.15	0.17
object0181	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0182	5	0.25	0.1	5	0.25	0.1	4	0.2	0.13
object0183	3	0.15	0.17	5	0.25	0.1	5	0.25	0.1
object0184	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0185	4	0.2	0.13	4	0.2	0.13	3	0.15	0.17
object0186	4	0.2	0.13	3	0.15	0.17	2	0.1	0.25
object0187	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1
object0188	3	0.15	0.17	2	0.1	0.25	4	0.2	0.13

object0189	4	0.2	0.13	5	0.25	0.1	3	0.15	0.17
object0190	5	0.25	0.1	5	0.25	0.1	2	0.1	0.25
object0191	1	0.05	0.5	3	0.15	0.17	3	0.15	0.17
object0192	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0193	2	0.1	0.25	4	0.2	0.13	5	0.25	0.1
object0194	4	0.2	0.13	5	0.25	0.1	5	0.25	0.1
object0195	2	0.1	0.25	2	0.1	0.25	3	0.15	0.17
object0196	2	0.1	0.25	5	0.25	0.1	4	0.2	0.13
object0197	2	0.1	0.25	5	0.25	0.1	5	0.25	0.1
object0198	2	0.1	0.25	5	0.25	0.1	4	0.2	0.13
object0199	3	0.15	0.17	5	0.25	0.1	2	0.1	0.25
object0200	5	0.25	0.1	5	0.25	0.1	2	0.1	0.25
object0201	3	0.15	0.17	4	0.2	0.13	4	0.2	0.13
zubud.csv	5	0.25	0.1	5	0.25	0.1	5	0.25	0.1

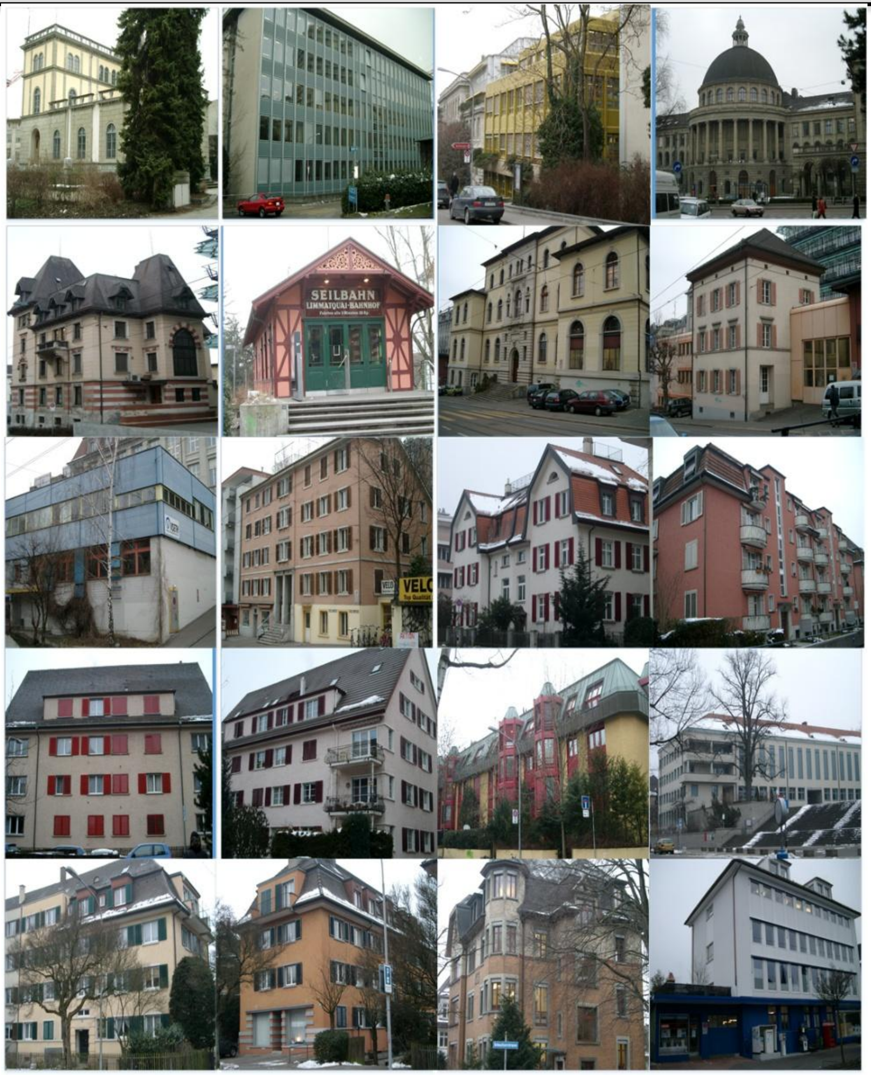

Figure 25: Zubud dataset showing different sample images of categories

Figure 25 illustrates the average precision results for various categories in the Zubud dataset using a bar chart. This visual approach facilitates effective classification and comparison across categories. Image classification was performed using convolutional neural networks (CNNs), incorporating feature extraction, image scaling, and integration techniques. The evaluation highlights that certain category—such as object006, object009, object013, and object014—achieved a perfect 100% average precision using DenseNet-201. Categories like object002, object004, object005, object010, object016, and object018 exhibited high precision rates between 90% and 95% when processed with GoogleNet. Some categories showed lower average precision, around 30%, using DenseNet-201. GoogleNet displayed comparatively lower precision in specific categories, though still exceeding 60% in most cases. Overall, more than half of the selected categories achieved an average precision above 70%, indicating strong performance on the Zubud dataset.

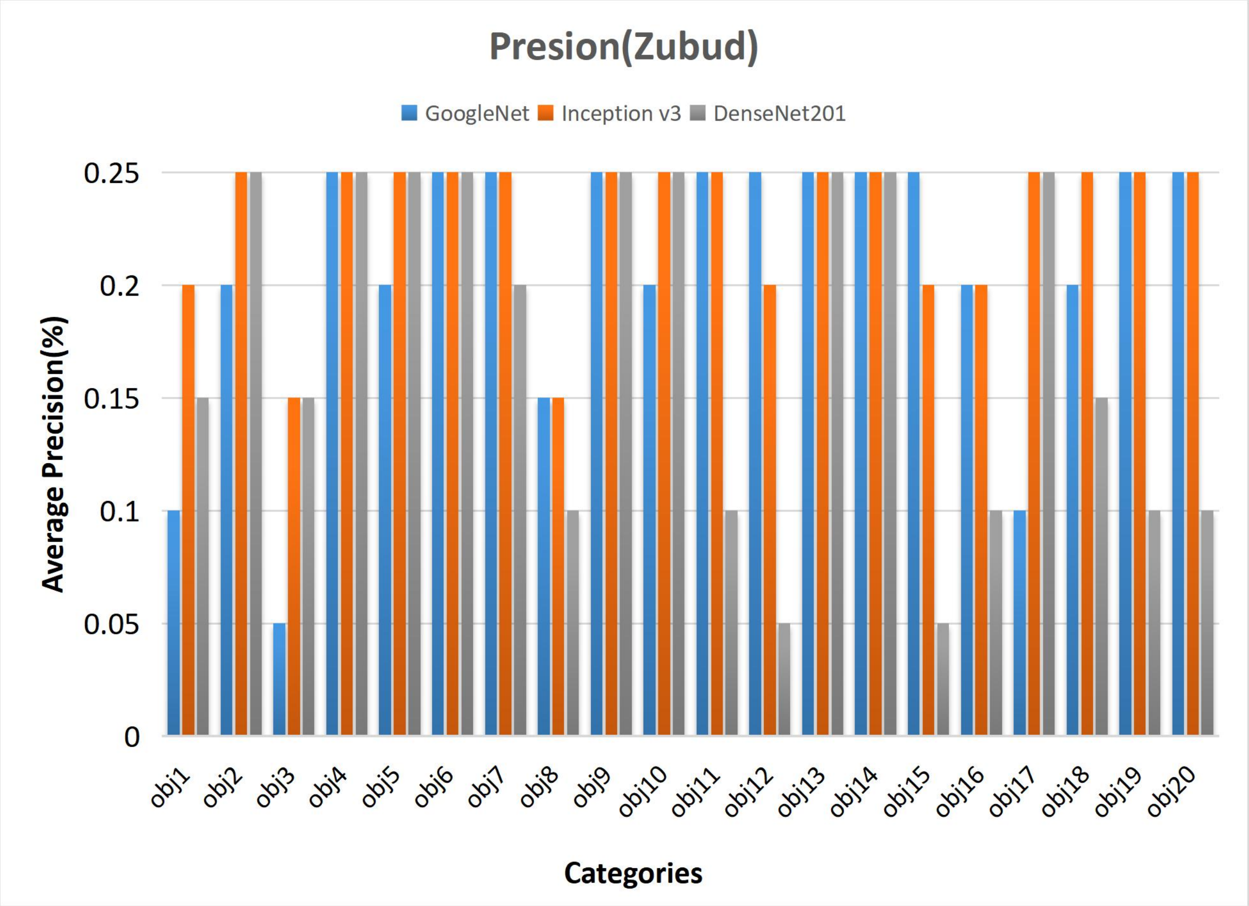

Figure 26: Average precision rate for 200 categories in zubud dataset.

Figure 27 presents the average recall values for 10 categories within the Zubud dataset, analyzed using three CNN-based feature extractors: GoogleNet, Inception v3, and DenseNet-201. A line graph is used to visualize category-wise performance, helping to assess the effectiveness of each network. DenseNet-201 demonstrated strong performance in categories such as obj30, obj80, obj120, and obj140, achieving recall rates close to 70%. However, the majority of categories showed recall values at or below 30%. Inception v3 delivered comparatively better results, with most categories achieving recall values of 25% or higher, and nearly half exceeding 60%. In several instances, Inception v3 recall values ranged from 80% to 90%. GoogleNet showed moderate performance, with categories like obj30, obj80, obj120, and obj140 reaching recall values below 50%. Overall, the dataset exhibited average recall rates around 50% for most categories, with Inception v3 providing the most consistent performance.

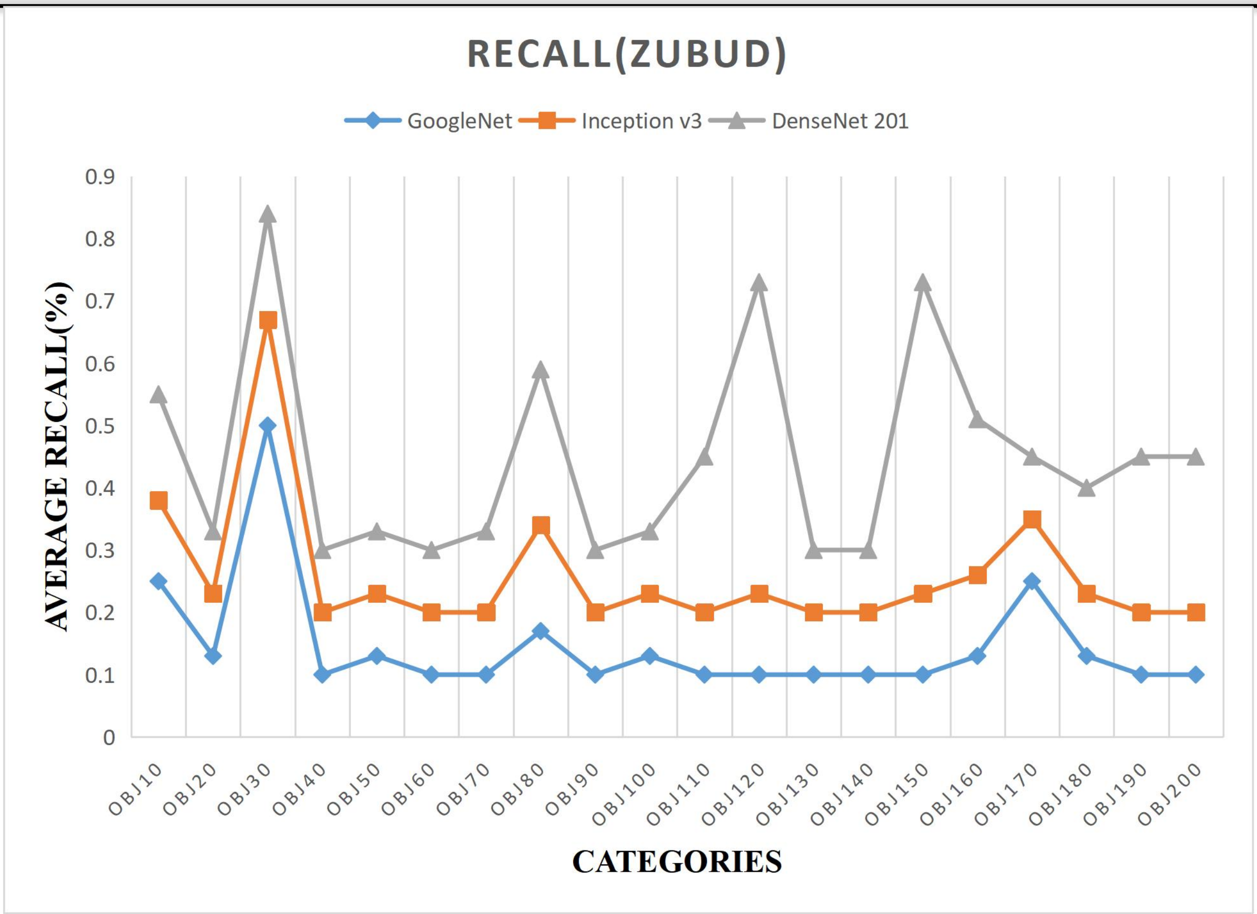

Figure 27: Average Recall rate for 200 categories in zubud dataset.

Figure 28 illustrates the average retrieval precision across various categories in the Zubud dataset using different CNN feature extractors. Inception v3 yielded the highest average retrieval precision, reaching approximately 96% for most object categories. DenseNet-201 also performed strongly, particularly for object001, object007, and object008, where it achieved precision levels exceeding 94%. In comparison, GoogleNet showed relatively lower precision, averaging around 60% across the evaluated categories.

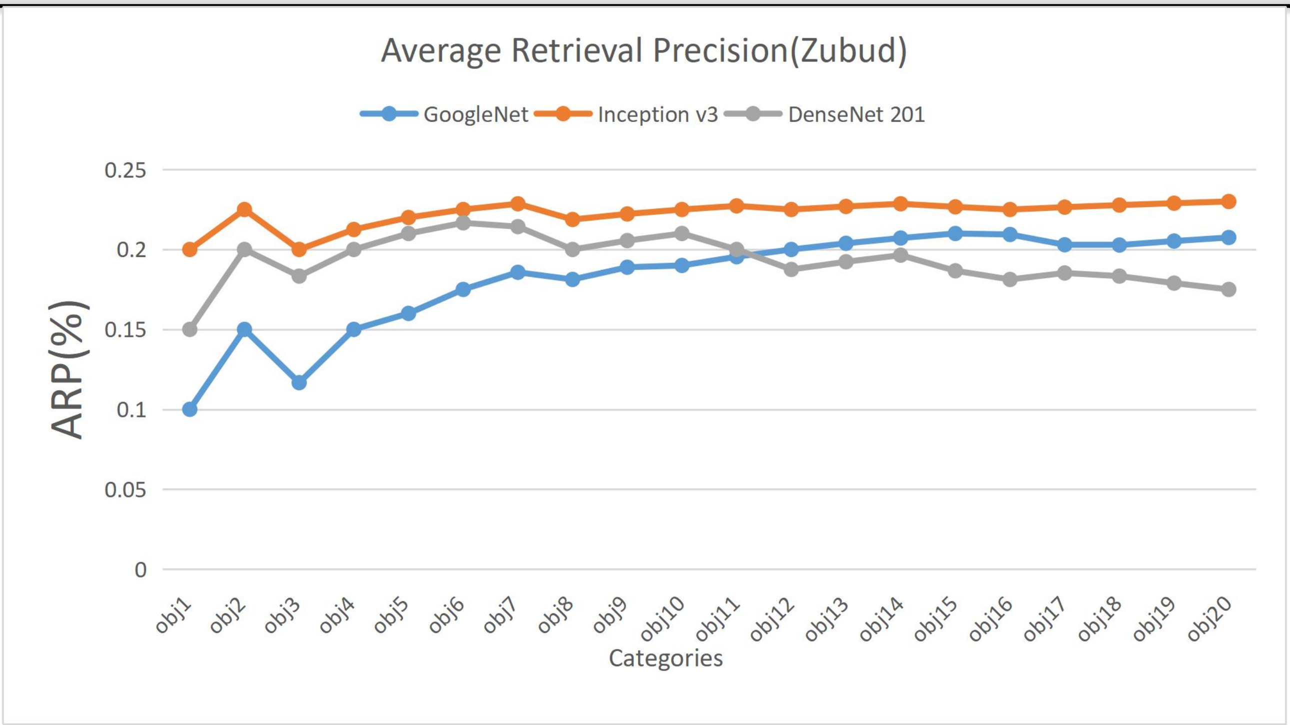

Figure 29: Average Retrieval precision rate for 200 categories in zubud dataset.

Figure 30 presents a line graph illustrating the effectiveness of average retrieval recall across different categories in the Zubud dataset. The recall values range from 30% to 85%, with the highest retrieval recall observed for object003. DenseNet-201 achieved the top recall rate for this category, exceeding 85%, while Inception v3 also performed strongly for the same object. Conversely, Inception v3 recorded the lowest recall for object200 (or object020). The graph further reveals that GoogleNet consistently produced lower recall values, indicating that it may offer higher precision in comparison.

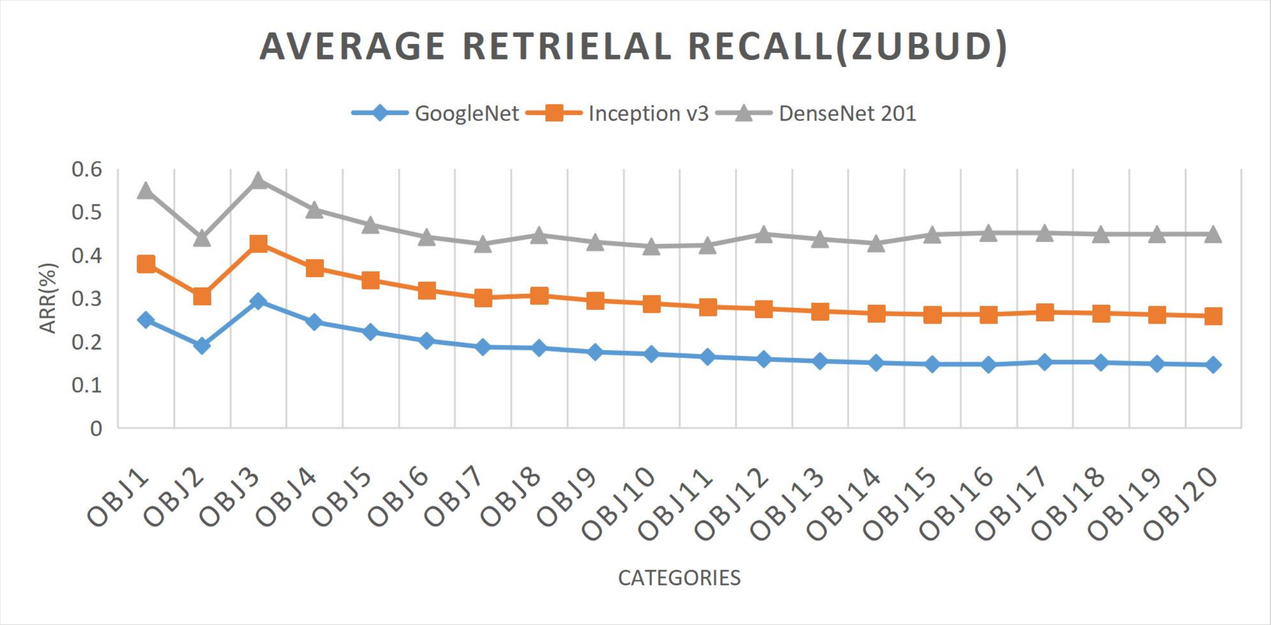

Figure 30: Average Retrieval Recall rate for 200 categories in zubud dataset.

Figure 31 illustrates the evaluation of mean average precision (MAP) for the Zubud dataset. Among the tested CNN models, Inception v3 achieved the highest MAP, reaching approximately 92%. GoogleNet followed with a MAP of around 83%. In contrast, DenseNet-201 yielded comparatively lower performance in terms of MAP for this dataset.

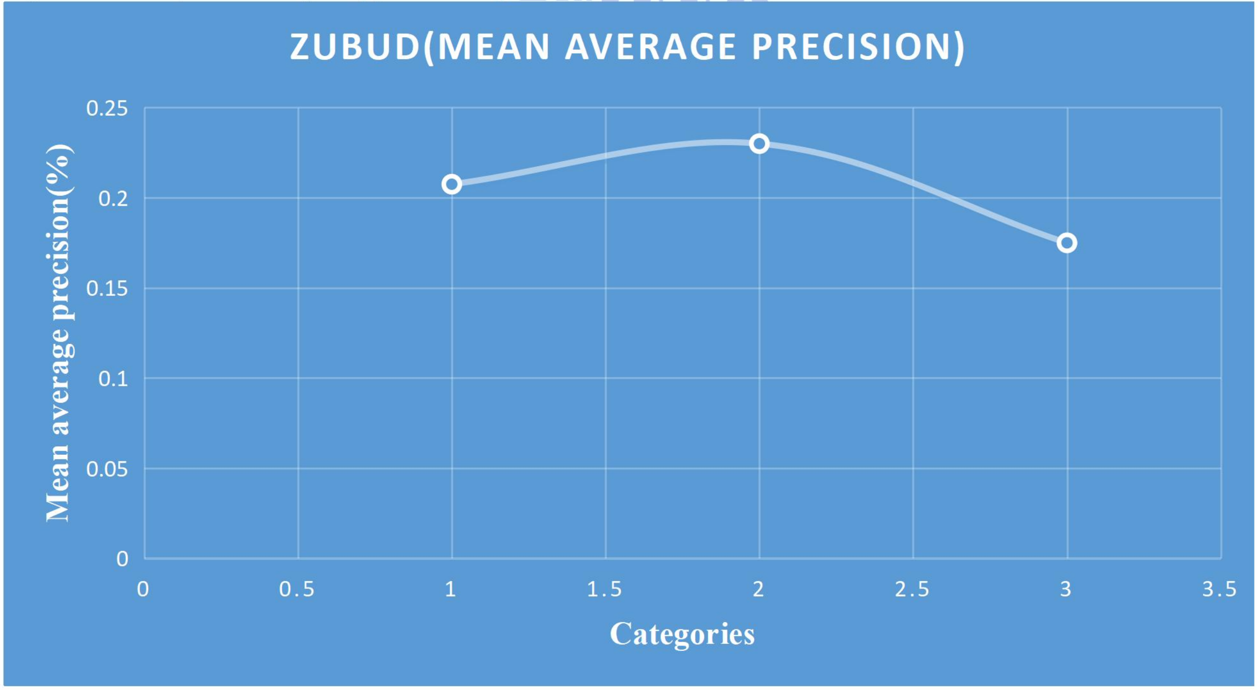

Figure 31: Mean Average precision rate for 200 categories in zubud dataset.

Figure 32 presents the evaluation of mean average recall (MAR) for the Zubud dataset. DenseNet-201 demonstrated the highest MAR, achieving approximately 90%. GoogleNet followed with a recall rate of around 70%, while Inception v3 showed a comparatively lower MAR of about 56%.

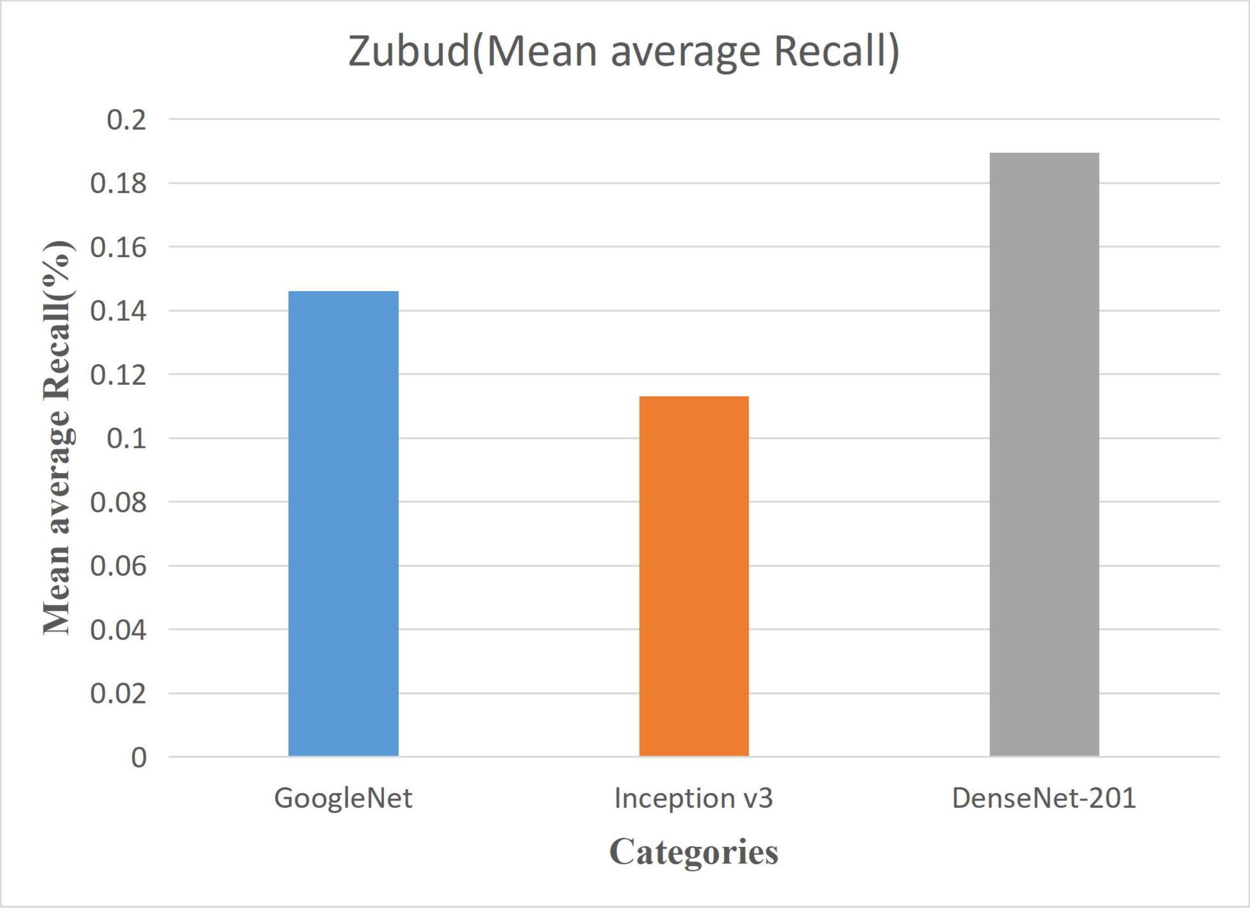

Figure 32: Mean Average Recall rate for 200 categories in zubud dataset.

4. Conclusion

This study focuses on the retrieval of images based on variations in shape, texture, and spatial color features. Using descriptive vectors, distinct shapes and color characteristics were extracted from various datasets through convolutional neural network (CNN) architectures such as GoogleNet, Inception v3, and DenseNet-201. Four benchmark evaluations—Average Precision (AP), Average Recall (AR), Average Average Precision (AAP), and Average Average Recall (AAR)—along with MAP and MAR, were conducted to assess retrieval similarity. The experimental outcomes demonstrated exceptional performance of the CBIR (Content-Based Image Retrieval) framework across all evaluated benchmarks. The method effectively identified and retrieved relevant features like color, shape, and texture from images across multiple datasets. To enhance the analysis, a combination of Non-Maximum Suppression (NMS) and neighborhood-based techniques was proposed in place of conventional filtering methods. The inclusion of spatial color as a feature in CBIR, along with global and neighborhood descriptors, was found to be beneficial, although in some cases it led to false positives. To address this, a circular sampling method was used to define neighborhood keypoints at equidistant positions, allowing for improved keypoint localization through pixel masking. Overall, the proposed approach yielded highly accurate image retrieval results from four large-scale datasets.

Reference

[1] Khan, S.U.R., Asif, S., Bilal, O. et al. Lead-cnn: lightweight enhanced dimension reduction convolutional neural network for brain tumor classification. Int. J. Mach. Learn. & Cyber. (2025). https://doi.org/10.1007/s13042-025-02637-6.

[2] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). Robust & Precise Knowledge Distillation-based Novel Context-Aware Predictor for Disease Detection in Brain and Gastrointestinal. arXiv preprint arXiv:2505.06381..

[3] Hekmat, A., et al., Brain tumor diagnosis redefined: Leveraging image fusion for MRI enhancement classification. Biomedical Signal Processing and Control, 2025. 109: p. 108040.

[4] Khan, Z., Hossain, M. Z., Mayumu, N., Yasmin, F., & Aziz, Y. (2024, November). Boosting the Prediction of Brain Tumor Using Two Stage BiGait Architecture. In 2024 International Conference on Digital Image Computing: Techniques and Applications (DICTA) (pp. 411-418). IEEE.

[5] Khan, S. U. R., Raza, A., Shahzad, I., & Ali, G. (2024). Enhancing concrete and pavement crack prediction through hierarchical feature integration with VGG16 and triple classifier ensemble. In 2024 Horizons of Information Technology and Engineering (HITE)(pp. 1-6). IEEE https://doi. org/10.1109/HITE63532.

[6] Khan, S.U.R., Zhao, M. & Li, Y. Detection of MRI brain tumor using residual skip block based modified MobileNet model. Cluster Comput 28, 248 (2025). https://doi.org/10.1007/s10586-024-04940-3

[7] Khan, U. S., & Khan, S. U. R. (2024). Boost diagnostic performance in retinal disease classification utilizing deep ensemble classifiers based on OCT. Multimedia Tools and Applications, 1-21.

[8] Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

[9] Al-Khasawneh, M. A., Raza, A., Khan, S. U. R., & Khan, Z. (2024). Stock Market Trend Prediction Using Deep Learning Approach. Computational Economics, 1-32.

[10] Khan, U. S., Ishfaque, M., Khan, S. U. R., Xu, F., Chen, L., & Lei, Y. (2024). Comparative analysis of twelve transfer learning models for the prediction and crack detection in concrete dams, based on borehole images. Frontiers of Structural and Civil Engineering, 1-17.

[11] Khan, S. U. R., & Asif, S. (2024). Oral cancer detection using feature-level fusion and novel self-attention mechanisms. Biomedical Signal Processing and Control, 95, 106437.

[12] Farooq, M. U., Khan, S. U. R., & Beg, M. O. (2019, November). Melta: A method level energy estimation technique for android development. In 2019 International Conference on Innovative Computing (ICIC) (pp. 1-10). IEEE.

[13] Asim, M. N., Ibrahim, M. A., Malik, M. I., Dengel, A., & Ahmed, S. (2020). Enhancer-dsnet: a supervisedly prepared enriched sequence representation for the identification of enhancers and their strength. In Neural Information Processing: 27th International Conference, ICONIP 2020, Bangkok, Thailand, November 23–27, 2020, Proceedings, Part III 27 (pp. 38-48). Springer International Publishing.

[14] Raza, A.; Meeran, M.T.; Bilhaj, U. Enhancing Breast Cancer Detection through Thermal Imaging and Customized 2D CNN Classifiers. VFAST Trans. Softw. Eng. 2023, 11, 80–92.

[15] Dai, Q., Ishfaque, M., Khan, S. U. R., Luo, Y. L., Lei, Y., Zhang, B., & Zhou, W. (2024). Image classification for sub-surface crack identification in concrete dam based on borehole CCTV images using deep dense hybrid model. Stochastic Environmental Research and Risk Assessment, 1-18.

[16] Muhammad, N. A., Rehman, A., & Shoaib, U. (2017). Accuracy based feature ranking metric for multi-label text classification. International Journal of Advanced Computer Science and Applications, 8(10).

[17] Mehmood, F., Ghafoor, H., Asim, M. N., Ghani, M. U., Mahmood, W., & Dengel, A. (2024). Passion-net: a robust precise and explainable predictor for hate speech detection in roman urdu text. Neural

Computing and Applications, 36(6), 3077-3100.

[18] Khan, S.U.R.; Asif, S.; Bilal, O.; Ali, S. Deep hybrid model for Mpox disease diagnosis from skin lesion images. Int. J. Imaging Syst. Technol. 2024, 34, e23044.

[19] Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X.; Zhu, Y. GLNET: Global–local CNN's-based informed model for detection of breast cancer categories from histopathological slides. J. Supercomput. 2023, 80, 7316–7348.

[20] Saleem, S., Asim, M. N., Van Elst, L., & Dengel, A. (2023). FNReq-Net: A hybrid computational framework for functional and non-functional requirements classification. Journal of King Saud University-Computer and Information Sciences, 35(8), 101665.

[21] Hekmat, Arash, Zuping Zhang, Saif Ur Rehman Khan, Ifza Shad, and Omair Bilal. "An attention-fused architecture for brain tumor diagnosis." Biomedical Signal Processing and Control 101 (2025): 107221.

[22] Khan, S.U.R.; Zhao, M.; Asif, S.; Chen, X. Hybrid-NET: A fusion of DenseNet169 and advanced machine learning classifiers for enhanced brain tumor diagnosis. Int. J. Imaging Syst. Technol. 2024, 34, e22975.

[23] Khan, S.U.R.; Raza, A.;Waqas, M.; Zia, M.A.R. Efficient and Accurate Image Classification Via Spatial Pyramid Matching and SURF Sparse Coding. Lahore Garrison Univ. Res. J. Comput. Sci. Inf. Technol. 2023, 7, 10–23.

[24] Farooq, M.U.; Beg, M.O. Bigdata analysis of stack overflow for energy consumption of android framework. In Proceedings of the 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 1–2 November 2019; pp. 1–9.

[25] Shahzad, I., Khan, S. U. R., Waseem, A., Abideen, Z. U., & Liu, J. (2024). Enhancing ASD classification through hybrid attention-based learning of facial features. Signal, Image and Video Processing, 1-14.

[26] Khan, S. R., Raza, A., Shahzad, I., & Ijaz, H. M. (2024). Deep transfer CNNs models performance evaluation using unbalanced histopathological breast cancer dataset. Lahore Garrison University Research Journal of Computer Science and Information Technology, 8(1).

[27] Bilal, Omair, Asif Raza, and Ghazanfar Ali. "A Contemporary Secure Microservices Discovery Architecture with Service Tags for Smart City Infrastructures." VFAST Transactions on Software Engineering 12, no. 1 (2024): 79-92.

[28] Khan, S. U. R., Asif, S., Zhao, M., Zou, W., Li, Y., & Li, X. (2025). Optimized deep learning model for comprehensive medical image analysis across multiple modalities. Neurocomputing, 619, 129182.

[29] Khan, S. U. R., Asif, S., Zhao, M., Zou, W., & Li, Y. (2025). Optimize brain tumor multiclass classification with manta ray foraging and improved residual block techniques. Multimedia Systems, 31(1), 1-27.

[30] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). AI-Driven Diabetic Retinopathy Diagnosis Enhancement through Image Processing and Salp Swarm Algorithm-Optimized Ensemble Network. arXiv preprint arXiv:2503.14209.

[31] Khan, Z., Khan, S. U. R., Bilal, O., Raza, A., & Ali, G. (2025, February). Optimizing Cervical Lesion Detection Using Deep Learning with Particle Swarm Optimization. In 2025 6th International Conference on Advancements in Computational Sciences (ICACS) (pp. 1-7). IEEE.

[32] Khan, S.U.R., Raza, A., Shahzad, I., Khan, S. (2025). Subcellular Structures Classification in Fluorescence Microscopic Images. In: Arif, M., Jaffar, A., Geman, O. (eds) Computing and Emerging Technologies. ICCET 2023. Communications in Computer and Information Science, vol 2056. Springer, Cham. https://doi.org/10.1007/978-3-031-77620-5_20

[33] Hekmat, A., Zuping, Z., Bilal, O., & Khan, S. U. R. (2025). Differential evolution-driven optimized ensemble network for brain tumor detection. International Journal of Machine Learning and Cybernetics, 1-26.

[34] Khan, S. U. R. (2025). Multi-level feature fusion network for kidney disease detection. Computers in Biology and Medicine, 191, 110214.

[35] Khan, S. U. R., Asif, S., & Bilal, O. (2025). Ensemble Architecture of Vision Transformer and CNNs for Breast Cancer Tumor Detection From

Mammograms. International Journal of Imaging Systems and Technology, 35(3), e70090.

[36] Meeran, M. T., Raza, A., & Din, M. (2018). Advancement in GSM Network to Access Cloud Services. Pakistan Journal of Engineering, Technology & Science [ISSN: 2224-2333], 7(1).

[37] Waqas, M., Ahmed, S. U., Tahir, M. A., Wu, J., & Qureshi, R. (2024). Exploring Multiple Instance Learning (MIL): A brief survey. Expert Systems with Applications, 123893.

[38] Mahmood, F., Abbas, K., Raza, A., Khan,M.A., & Khan, P.W. (2019). Three Dimensional Agricultural Land Modeling using Unmanned Aerial System (UAS). International Journal of Advanced Computer Science and Applications (IJACSA) [p-ISSN : 2158-107X, e-ISSN : 2156-5570], 10(1).

[39] M. Wajid, M. K. Abid, A. Asif Raza, M. Haroon, and A. Q. Mudasar, "Flood Prediction System Using IOT & Artificial Neural Network", VFAST trans. softw. eng., vol. 12, no. 1, pp. 210–224, Mar. 2024.

[40] M. Waqas, Z. Khan, S. U. Ahmed and A. Raza, "MIL-Mixer: A Robust Bag Encoding Strategy for Multiple Instance Learning (MIL) using MLP-Mixer," 2023 18th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 2023, pp. 22-26.

[41] Khan, S. U. R., & Khan, Z. (2025). Detection of Abnormal Cardiac Rhythms Using Feature Fusion Technique with Heart Sound Spectrograms. Journal of Bionic Engineering, 1-20.

[42] Khan, M.A., Khan, S.U.R. & Lin, D. Shortening surgical time in high myopia treatment: a randomized controlled trial comparing non-OVD and OVD techniques in ICL implantation. BMC Ophthalmol 25, 303 (2025). https://doi.org/10.1186/s12886-025-04135-3

[43] Shahzad, I., Raza, A., & Waqas, M. (2025). Medical Image Retrieval using Hybrid Features and Advanced Computational Intelligence Techniques. Spectrum of engineering sciences, 3(1), 22-65.

[44] Raza, A., Shahzad, I., Ali, G., & Soomro, M. H. (2025). Use Transfer Learning VGG16, Inception, and Reset50 to Classify IoT Challenge in Security Domain via Dataset Bench Mark. Journal

of Innovative Computing and Emerging Technologies, 5(1).

[45] Raza, A., & Shahzad, I. (2024). Residual Learning Model-Based Classification of COVID-19 Using Chest Radiographs. Spectrum of engineering sciences, 2(3), 367-396.

[46] Raza, A., Soomro, M. H., Shahzad, I., & Batool, S. (2024). Abstractive Text Summarization for Urdu Language. Journal of Computing & Biomedical Informatics, 7(02).

[47] HUSSAIN, S., RAZA, A., MEERAN, M. T., IJAZ, H. M., & JAMALI, S. (2020). Domain Ontology Based Similarity and Analysis in Higher Education. IEEEP New Horizons Journal, 102(1), 11-16.

[48] Raza, A., & Meeran, M. T. (2019). Routine of encryption in cognitive radio network. Mehran University Research Journal of Engineering & Technology, 38(3), 609-618.

[49] Khan, U. S., & Khan, S. U. R. (2025). Ethics by Design: A Lifecycle Framework for Trustworthy AI in Medical Imaging From Transparent Data Governance to Clinically Validated Deployment. arXiv preprint arXiv:2507.04249.

[50] Maqsood, H., & Khan, S. U. R. (2025). MeD-3D: A Multimodal Deep Learning Framework for Precise Recurrence Prediction in Clear Cell Renal Cell Carcinoma (ccRCC). arXiv preprint arXiv:2507.07839.

[51] Khan, S. U. R., Asim, M. N., Vollmer, S., & Dengel, A. (2025). FOLC-Net: A Federated-Optimized Lightweight Architecture for Enhanced MRI Disease Diagnosis across Axial, Coronal, and Sagittal Views. arXiv preprint arXiv:2507.06763.

[52] Bilal, O., Hekmat, A., & Khan, S. U. R. (2025). Automated cervical cancer cell diagnosis via grid search-optimized multi-CNN ensemble networks. Network Modeling Analysis in Health Informatics and Bioinformatics, 14(1), 67.