Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

SET-CNN: Stacked Ensemble of CNNs for Robust Image Embedding and Similarity Retrieval

Rabia Maqsood
Muhammad Omar
muhammad.umar@iub.edu.pk
Islamia University of Bahawalpur
Safdar Ali
Iqra University, Chak Shahzad Campus
Rahat H. Bokhari
Iqra University, Chak Shahzad Campus

Research Article

Keywords: Image Retrieval, Convolutional Neural Networks, Ensemble Learning, Feature Fusion, Triplet Loss, Visual Representation

Posted Date: September 15th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8012706/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

SET-CNN: Stacked Ensemble of CNNs for Robust Image Embedding and Similarity Retrieval

Rabia Maqsood[a], Muhammad Omar[a,#,*], Safdar Ali[b], Rahat H. Bokhari[b]

[a]Department of Computer Science, The Islamia University of Bahawalpur, Pakistan

[b]Department of Computing and Technology, Iqra University, Chak Shahzad Campus, Islamabad, Pakistan

[#]Co-first author: Muhammad Omar

[*]Corresponding Autor. E-mail address: m.omar.nazeer@gmail.com; muhammad.umar@iub.edu.pk (M. Omar).

Abstract

Image retrieval predominantly relies on embedding techniques, which represent visual data as numerical vectors to enable efficient and scalable similarity comparisons. Such embeddings are crucial for a variety of computer vision tasks, including image similarity search, image classification, object detection, and facial recognition. However, conventional techniques that rely on single-network architectures or domain-specific optimizations haveing limited capability to capture the essential and complex visual information required for precise and effective similar image retrieval. To address this limitation, SET-CNN (Stacked Ensemble of CNNs), a novel framework that integrates three diverse pretrained CNN models, DenseNet121, ResNet50, and VGG16 for generating robust image embeddings and enhancing similarity retrieval is proposed. This approach extracts and stacks high-level feature vectors from each base model, concatenates them via feature-level fusion, and processes the unified representation through a meta-model optimized using triplet semi-hard loss to enhance discriminability in the embedding space. Extensive experiments on the CIFAR-10 dataset demonstrate that SET-CNN consistently outperforms individual CNNs and state-of-the-art retrieval methods, achieving a peak training MAP@5 of 0.9286 and a testing MAP@5 of 0.8745, with minimal overfitting. This represents a 7.6% improvement over the best-performing base model. Compared to Deep Semantic Ranking Hashing and Deep Supervised Hashing, SET-CNN delivers absolute gains of +8.6% and +19.9% in MAP@5, respectively, while matching the performance of more complex ResNet attention pooling models without specialized architectural modifications. Embedding quality is assessed through K-Means clustering combined with t-SNE visualization, while final similarity matching is performed using K-Nearest Neighbors (KNN) with cosine similarity. These clustering and retrieval analyses confirm that the ensemble produces semantically rich and well-separated embeddings, translating into more accurate and relevant retrieval results. Given its performance and flexibility, SET-CNN offers a promising direction for developing generalizable, ensemble-based retrieval systems, with potential extensions to large-scale, multimodal datasets and real-time deployment.[1]

Keywords: Image Retrieval, Convolutional Neural Networks, Ensemble Learning, Feature Fusion, Triplet Loss, Visual Representation.

1. Introduction

The computer vision's basic tasks include retrieval of instance level imaging to identify the images in a dataset having similar object instances in response to a query initiated regarding image retrieval. It underpins numerous real-world applications, including reverse image search, photo organization, and annotation transfer, where retrieved images are used to propagate metadata such as tags [1], or semantic object locations [2].

[1] To facilitate reproducibility and further research, the complete implementation and evaluation code are publicly available at https://github.com/RabiaMaqsood-01/Image-Embedding-through-CNN-Models-and-Stacking-Ensemble-Technique

Recently, the image representation learning (i.e. Convolutional Neural Networks (CNNs)) has been significantly improved image classification [3], object detection [4], semantic segmentation [5], and image captioning [6]. However, despite their good performance in classification and recognition, CNN-based models often face limitations in instance-level retrieval. In particular, they may underperform compared to traditional local descriptor techniques with spatial re-ranking [7, 8], as they can struggle to preserve fine-grained, instance-specific details essential for retrieval accuracy.

Recent research has sought to narrow this gap through diverse embedding and optimization strategies. Examples include graph convolutional embeddings with self-supervision [9], hashing-based embeddings with optimized loss functions [10], and collaborative filtering approaches that combine CNN features for social tagging and retrieval [11]. Other efforts have explored semantic and universal embeddings, such as hierarchical embeddings for semantic similarity [12], implicit neural priors for reconstruction [13], and multimodal models combining visual and textual information[14]. While these approaches have achieved notable improvements, many still rely on single-model optimizations or domain-specific designs, limiting their ability to generalize across datasets and retrieval scenarios. Performance evaluations on benchmark datasets such as CIFAR-10 reveal a persistent gap in retrieval accuracy. Methods such as Deep Semantic Ranking Hashing [15] and Deep Supervised Hashing [16] improve retrieval speed through compact binary representations but often at the cost of discriminative power. More complex CNN architectures, including ResNet variants with attention pooling [17], achieve high classification accuracy yet may not translate directly into superior retrieval performance. These findings highlighted the need for ensemble-based strategies that integrate complementary feature representations from multiple architectures to produce richer, more robust embeddings.

To address these limitations, this study proposes SET-CNN (Stacked Ensemble of CNNs), a framework that combines DenseNet121, ResNet50, and VGG16 through feature-level fusion. Each model is fine-tuned to the target domain, and their embeddings are concatenated into a unified representation processed by a meta-model trained with triplet semi-hard loss. This approach leverages the complementary strengths of heterogeneous architectures to maximize intra-class compactness and inter-class separation in the embedding space.

The contributions of this work can be summarized as follows:

1. **Stacked ensemble retrieval framework (SET-CNN)** integrating DenseNet121, ResNet50, and VGG16 through feature-level fusion to generate rich and complementary visual embeddings.
2. **Retrieval-specific embedding optimization**, the developed approached using triplet semi-hard loss to efficiently improved the intra-class assembling and making distance a part for inter-class partition in the learned feature space.
3. **Comprehensive evaluation methodology** combining quantitative performance (MAP@5), qualitative visualization (t-SNE, K-Means clustering), and retrieval case studies.
4. **Demonstrated performance gains** over individual CNNs and state-of-the-art hashing-based methods on CIFAR-10, achieving up to +19.9% improvement in MAP@5.
5. **Architecture-agnostic design** enabling seamless extension to other datasets and

retrieval domains without structural modifications.

2. Background

Several conventional and advanced approaches are developed in past research that address image similarity and retrieval. However, these such approaches have certain limitations, which are discussed below.

2.1. CNN-based Image Retrieval

Deep learning has advanced image retrieval by enabling the extraction of high-level semantic features that surpass traditional handcrafted descriptors. Early CNN-based retrieval approaches demonstrated that features from classification networks can be repurposed for similarity search [18, 19]. Architectures such as VGG, ResNet, and DenseNet have been widely applied in retrieval settings [20]. However, when trained as standalone models, they often produce embeddings that are too domain-specific or insufficiently discriminative for fine-grained instance-level retrieval, particularly when visual differences between classes are subtle.

2.2. Embedding Optimization for Retrieval

Improving the discriminative quality of embeddings has been explored through both loss function design and feature transformation. Triplet loss and its semi-hard variant have been used to reduce intra-class variance and maximize inter-class separation, making them effective for retrieval tasks. Other optimization strategies include hierarchical embeddings for semantic similarity [12], graph convolutional embedding models [9], and hashing-based representations such as Deep Semantic Ranking Hashing (DSRH) [15, 16]. While these methods improve retrieval performance or efficiency, most rely on single-network architectures and may not fully exploit complementary features from multiple CNN backbones. CNN baselines such as the method proposed by [21], which combines a convolutional network with K-Nearest Neighbors for classification, achieve competitive accuracy but remain limited in retrieval-specific optimization.

2.3. Ensemble Approaches

Ensemble learning is a well-established strategy for improving model robustness and generalization by combining the outputs of multiple models trained on the same or related tasks. In the context of image retrieval, ensembles can capture complementary visual features, enhancing the discriminative power of the resulting embeddings. Stacking-based methods [22] in particular integrate diverse model predictions at a meta-level, reducing variance and mitigating overfitting.

In prior retrieval-related work, multi-architecture comparisons, such as those reported by[12], show that different CNN backbones exhibit varying strengths in representing visual categories, implying potential benefits from combining them. Similarly, enhancements such as attention pooling applied to different ResNet variants [17] illustrate how integrating representation refinements across architectures can improve downstream performance. However, most existing ensemble strategies in retrieval operate at the decision level — for example, averaging similarity scores or merging ranked lists — rather than integrating features directly. Decision-level fusion can overlook the richer joint representations that emerge when diverse features are combined before similarity computation.

2.4. Comparative Summary of Prior Work

A summary of representative retrieval methods on benchmark datasets, particularly CIFAR-10, is presented in Table 1. This table lists architectures, evaluation metrics, and reported scores from prior work. The results show that hashing-based methods such as DSRH (MAP = 79.17%) and DSH (MAP = 67.55%) offer

efficient retrieval through compact binary codes but achieve lower discriminative accuracy. CNN-based approaches, including the CNN+KNN method of[21], deliver higher accuracy but lack retrieval-specific embedding optimization. Multi-architecture studies [12] and architecture enhancements such as attention pooling in ResNet variants[17] demonstrate improvements in representation learning but are not specifically tailored for retrieval performance.

Table 1. Comparative analysis of representative deep retrieval methods on CIFAR-10.

Method	Metric	Score (%)	Reference
Deep Semantic-Preserving & Ranking-Based Hashing (DSRH), Deep CNN	MAP	79.17	[15]
Deep Supervised Hashing (DSH), CNN	MAP	67.55	[16]
Plain-II, ResNet-110, PyramidNet-272-200, ResNet-50	MAP	74.10	[12]
ResNet-50, ResNet-36 + Attention Pooling	Mean Accuracy	(95.7 ± 0.1, 95.4 ± 0.2), (77.0 ± 0.1, 77.5 ± 0.1)	[17]
CNN	KNN Accuracy	83.6	[21]

These observations highlight three key limitations in existing approaches:

- Hashing-based methods → efficient but less discriminative.
- Single CNN models → discriminative but limited in generalization.
- Complex or enhanced architectures → high classification accuracy but not necessarily superior retrieval embeddings.

The proposed SET-CNN framework addresses this limitation by performing feature-level fusion of embeddings extracted from heterogeneous CNN architectures, followed by a meta-model that learns an optimized joint embedding space. The proposed SET-CNN framework aims to bridge these gaps by integrating complementary CNN features at the embedding level and optimizing them with a retrieval-specific objective to achieve higher accuracy and robustness.

3. Materials and Methods

3.1. Overview of the Proposed Method

The SET-CNN framework is proposed to enhance the quality of image embeddings for instance-level image retrieval. The detailed framework is depicted in Fig. 1. It integrates the feature representations from three pretrained CNN models (i.e. DenseNet121, ResNet50, and VGG16) to form unified, robust embedding. Each model is fine-tuned on the CIFAR-10 dataset and extracts embeddings from its penultimate layer. These embeddings are concatenated to form a unified vector using a feature-level fusion approach. The output of previous stage is then passed through a meta-model within a stacking ensemble framework as an input. The ensemble leverages the complementary strengths of diverse architectures to improve generalization and image retrieval accuracy.

3.2. Dataset and Preprocessing

We use the CIFAR-10 dataset, consisting of 60,000 RGB images (32×32 resolution) evenly distributed across 10 object classes such as automobiles, birds, cats, and ships (https://www.kaggle.com/datasets/ayush1220/cifar10). Each class contains 6,000 samples. This dataset is applicable in computer vision for image classification and retrieval. All input images are converted to float32 format, and the image pixel values (IPV) are normalized (IPV-Normalized values) using following transformation:

$$IPV_{Normalized} = \frac{IPV_value}{255.0} \quad (1)$$

This operation scales the pixel intensity from the original range [0,255] to the normalized range [0.0,1.0], ensuring consistency and stability during model training.

To improve the generalization process and increase the quantity of images, we applied seven distinct data augmentation techniques. These include rotating images within a range of -20 to +20 degrees, shifting them horizontally and vertically by up to 20% and 30% respectively, and enabling random horizontal flipping to add variability. Additionally, we employed shear transformations within a range of -20 to +20 degrees, random zooming within a range of [1-0.2, 1+0.2], and random channel shifts by up to 20% of the total intensity range. For newly created pixels during such transformations, we used the 'nearest' fill mode to maintain a natural appearance.

The balanced CIFAR-10 dataset is then partitioned into three distinct subsets: training (70%), validation (10%), and testing (20%). After developing model using 70% of the dataset, two distinct assessment phases are performed which are evaluation and testing. The evaluation phase occurs during training using a dedicated validation (10%) dataset to monitor learning progress, tune hyperparameters, and applying early stopping strategies. This ensures that the model is neither overfitted nor underfitted. Once the model is finalized, the testing phase is carried out using a separate test dataset i.e. 20% and it was not included during the training or evaluation stages. In this way, model is get generalized leading to unbiased assessment. Furthermore, it is used for reporting final performance metrics.

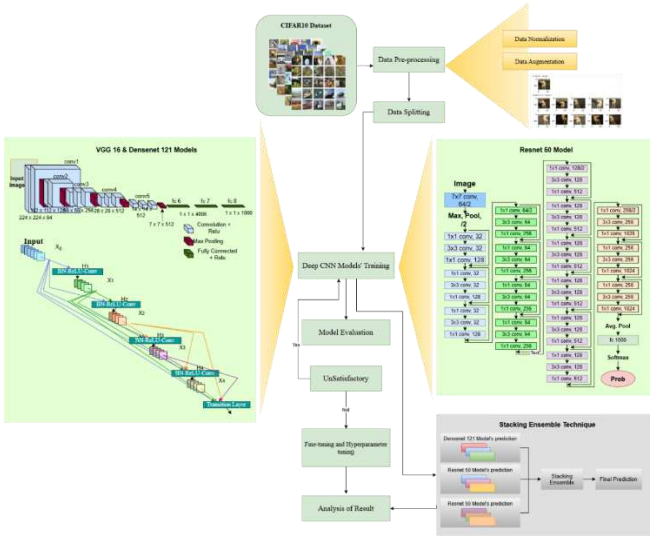


Fig. 1 Detailed SET-CNN framework for image similarity retrieval, along with layer-wise architectures of DenseNet121, ResNet50, and VGG16, including their convolutional and pooling stages.

3.3. Deep CNN Architectures

This section briefly outlines three widely used deep CNNarchitectures, DenseNet121, ResNet50, and VGG16**,** recognized for their effectiveness in visual representation and image embedding tasks. Architectures of these CNN are shown in Fig. 1.

DenseNet121 is a *Densely Connected Convolutional Network* proposed direct feedforward connections between all layers, promoting feature reuse and reducing the number of parameters. The architecture is structured as $[5 + (6 + 12 + 24 + 16) \times 2) = 121]$ layers, comprising five initial convolution and pooling layers, three transition layers, one classification layer, and two dense blocks consisting of 1×1 and 3×3 convolutions.

ResNet50, is a deep residual network that employs skip connections to mitigate the vanishing gradient problem and enables the

training of deeper architectures. The model begins with a 7×7 convolutional layer (64 filters, stride 2), followed by max pooling (stride 2), and proceeds through a series of residual blocks: 3×(1×1, 3×3, 1×1), 4×(…), 6×(…), and 3×(…), totaling 50 convolutional layers. It concludes with global average pooling and a fully connected layer with 1,000 output nodes activated via softmax.

VGG16, is characterized by a uniform architecture of 3×3 convolutional filters and 2×2 max-pooling layers. As shown in Fig. 5, the network stacks increasing numbers of filters (64, 128, 256, and 512) across five convolutional blocks. The model comprises 13 convolutional layers and 3 fully connected layers, summing to 16 weight-bearing layers, despite having 21 layers in total when including non-trainable layers [20].

3.4. Proposed Deep SET-CNN approach

The proposed SET-CNN (Stacked Ensemble of CNNs) approach builds upon the principle of ensemble learning by integrating multiple deep learning models to generate robust and generalizable image embeddings. Unlike traditional models that rely on a single CNN architecture, this method leverages the complementary strengths of three diverse pretrained models: DenseNet121, ResNet50, and VGG16. Fig. 2 presents an overview of the proposed SET-CNN architecture, illustrating the combination of the three CNNs and a meta-model designed for image similarity retrieval. This Deep Ensemble Model is an advanced extension of classical ensemble methods, tailored for deep learning architectures. It fuses high-level representations from multiple CNNs to achieve improved accuracy, robustness, and generalization in downstream tasks such as image similarity retrieval.

The proposed model is selected for image similarity retrieval due to following main reasons. First, it reduces the variance component of prediction errors, thereby enhancing accuracy. Second, it improves model robustness by stabilizing outputs across independently trained networks. Given recent advancements in computational power and reductions in hardware costs, the additional complexity introduced by ensemble architectures is often justified by the resultant performance gains.

In this deep ensemble learning framework, predictions from DenseNet121, ResNet50, and VGG16 are independently generated using models fine-tuned on the CIFAR-10 dataset. Each model is initialized with different random seeds to promote diversity in learned representations. Once trained, embeddings are extracted from the penultimate layers of each base CNN, capturing high-level semantic features of the input images.

To fuse these embeddings, a feature-level fusion strategy i.e. the extracted embeddings are concatenated horizontally using hstack, and forming a single composite feature vector is employed. This unified vector is subsequently passed to a meta-model, which learns to optimally integrate the representations for improved prediction. The meta-model is trained using triplet semi-hard loss, which encourages embeddings of similar images to be closer in the latent space while pushing apart those of dissimilar images. This structure enhances the discriminative quality of the learned embedding space, which is essential for high-performance image similarity retrieval.

By unifying multiple perspectives through SET-CNN, the framework achieves improved generalization, resilience to noisy data, and superior retrieval accuracy—validated through MAP@5 scores and clustering visualizations.

For an input image $x \in \mathbb{R}^{224\times224\times3}$ (224×224 RGB image), each of the m base models generates an embedding $\hat{y}_i(x)$ of dimension d_i. The term

$BM_i(x)$ represents the feature extraction function of the i^{th} model.

$$\hat{y}_i(x) = BM_i(x) \in \mathbb{R}^{d_i}, \quad i = 1, 2, ..., m. \tag{2}$$

For a 224×224 RGB image, three base models—DenseNet121, ResNet50, and VGG16—produce embeddings of 64, 64, and 94 dimensions, respectively. Here, $d_{DenseNet121}(x)$, $r_{ResNet50}(x)$, and $v_{VGG16}(x)$ denote the feature extraction functions of each model, with $\hat{y}_i(x)$ as their respective embeddings.

$$\hat{y}_1(x) = d_{DenseNet121}(x), \quad \hat{y}_2(x) = r_{ResNet50}(x), \quad \hat{y}_3(x) = v_{VGG16}(x) \tag{3}$$

The embeddings $\hat{y}_i(x)$ from all base models are concatenated horizontally using the hstack operation to form a single composite feature vector Y_{concat}. The dimension of Y_{concat} is the sum of individual embedding dimensions d_i.

$$\hat{Y}_{concat} = hstack(\hat{y}_1(x), \hat{y}_2(x), ..., \hat{y}_m(x)) \in \mathbb{R}^{d_1+d_2+...+d_m} \tag{4}$$

The concatenated vector $\hat{Y}_{concat}$ is passed to a meta-model M, which produces a refined embedding z(x) in a latent space of dimension d_{meta}. The meta-model learns to optimally integrate the base model embeddings.

$$z(x) = M(\hat{Y}_{concat}(x)) \in \mathbb{R}^{d_{meta}}. \tag{5}$$

The meta-model is trained using triplet semi-hard loss, where a, p, and n are anchor, positive (similar), and negative (dissimilar) images, respectively.

$$L_{triplet} = \Sigma_{(a,p,n)\in T} \max(0, ||z(a) - z(p)||^2_2 - ||z(a) - z(n)||^2_2 + \alpha) \tag{6}$$

where $z(x) = M(\hat{Y}_{concat}(x))$, and semi-hard negatives satisfy:

$$||z(a) - z(p)||^2_2 < ||z(a) - z(n)||^2_2 < ||z(a) - z(p)||^2_2 + \alpha \tag{7}$$

The term z(x) is the meta-model's embedding, $||z(a) - z(p)||^2_2$ is the squared Euclidean distance, α is a margin parameter, and T is the set of triplets. Semi-hard negatives are selected to be within the margin but not too close to the anchor.

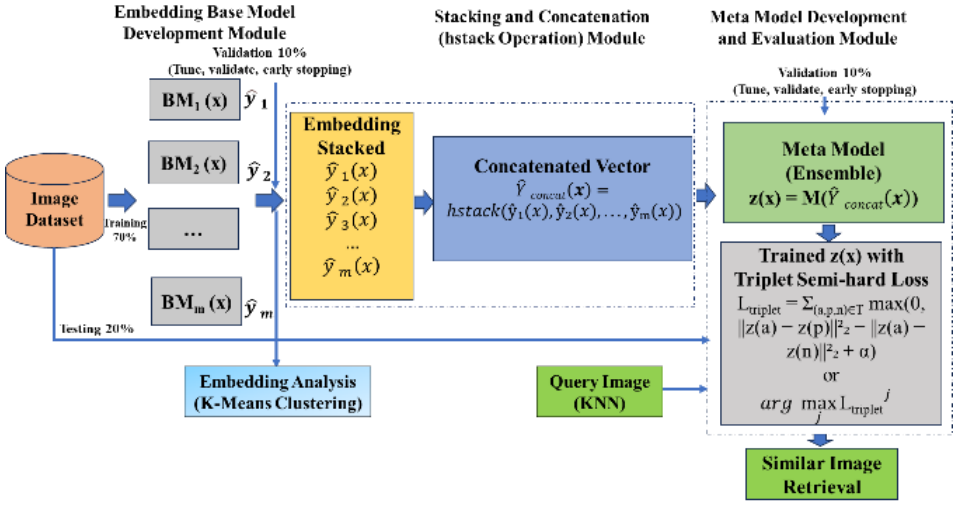


Fig. 2 Overview of the proposed SET-CNN ensemble architecture combining base models, and a meta-model for image similarity retrieval.

The complete implementation of the proposed SET-CNN framework, including dataset preprocessing scripts, training configurations, and evaluation tools, is publicly available at .[2]

[2] To facilitate reproducibility and further research, the complete implementation and evaluation code are publicly available at https://github.com/RabiaMaqsood-01/Image-Embedding-through-CNN-Models-and-Stacking-Ensemble-Technique

3.5. Evaluation Metric

To evaluate retrieval quality, we used Mean Average Precision at 5 (MAP@5), a widely used metric in information retrieval and recommendation systems. MAP@5 allows for robust evaluation by prioritizing early precision, crucial for practical retrieval tasks. It emphasizes ranking accuracy, which is critical in image similarity tasks. MAP@5 computes the average precision over the top 5 retrieved results for each query and then takes the mean across all queries. It is defined as:

$$MAP@5 = \frac{1}{Q} \sum_{p=1}^{Q} \frac{1}{\min(n_{p,5})} \sum_{j=1}^{\min(n_{p,5})} rel_p \text{ (j)} \quad (8)$$

where:

- Q represents the quantity of query images.
- n_p indicates the count of index images that share an object in common with query image P. It is noteworthy that $n_p > 0$.
- rel_p (j) signifies the relevance of prediction j for p^{-th} query: it is set to 1 if j^{-th} prediction is accurate, and 0 otherwise.

3.6. K-Means Clustering for Embedding Analysis

To analyze the quality of the embeddings, we applied K-Means clustering, a simple yet effective unsupervised learning method. It partitions the data into K clusters by minimizing the Euclidean distance between data points and cluster centroids. The clustering process includes:

1. Selecting K initial centroids.
2. Assigning each data point to its nearest centroid.
3. Updating centroids as the mean of their assigned points.
4. Repeating until convergence.

Euclidean distance: $d = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2}$ (9)

Here, (a_1, b_1) and (a_2, b_2) are the coordinates of the two points in the plane, d represents Euclidean distance between them. If every cluster centroid be represented as c_i, then the assignment of each data point x to a cluster is determined by

$$\arg \min_{ci \in c} \text{dist}(c_i, x)^2 \quad (10)$$

New centroid for cluster:

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \quad (11)$$

S_i represents the collection of all points allocated to the i^{th} cluster.

Clusters generated by DenseNet121, ResNet50, VGG16 and SET-CNN are visualized using t-SNE (t-Distributed Stochastic Neighbor Embedding) technique, revealing how well the embeddings preserve semantic similarity [23]. Distinct and separated clusters affirm the embeddings' semantic discriminability [24]. Well-separated clusters indicate strong embedding quality.

3.7. K-Nearest Neighbors (KNN) for Retrieval

KNN is employed as the final retrieval mechanism. This supervised algorithm identifies the K most similar embeddings to a query image and classifies or ranks based on proximity.

Procedure:

1. Set a value for K.
2. Compute distances (using cosine similarity) between the query and all training embeddings.
3. Retrieve K closest embeddings.
4. Return the most frequent label (or ranked similarity list).

KNN provides intuitive and effective retrieval performance and serves as a practical use case for evaluating the quality of the unified embeddings generated [10].

4. Experimental Results and Discussion

This section presents the experimental evaluation of the proposed SET-CNN framework for image similarity retrieval, including base model training, fine-tuning, hyperparameter optimization, embedding analysis, retrieval

experiments, and comparison with state-of-the-art methods.

4.1. Model Training and Optimization

DenseNet121, ResNet50, and VGG16 were trained on 70% of the CIFAR-10 training set, initialized with ImageNet-pretrained weights. An additional 10% validation split was used for fine-tuning. Initially, the convolutional base layers were frozen, and training was performed on added components—GlobalAveragePooling2D, fully connected layers, and dropout. Layers were then progressively unfrozen for deeper fine-tuning. The extracted embeddings from each model were subsequently combined in a meta-model as part of the stacking ensemble.

4.2. Fine-Tuning and Hyperparameter Tuning of Models

The optimization process employed triplet semi-hard loss, selecting an anchor, a positive sample from the same class, and a semi-hard negative (different class but within a defined margin). This approach maximized inter-class separation and minimized intra-class variation, yielding a more discriminative embedding space.

The tuning pipeline comprised four stages:

1. Last-layer fine-tuning (frozen base layers)
2. Optimizer adjustment (SGD)
3. Learning rate tuning (LR = 0.01)
4. Full hyperparameter optimization (batch size, dropout, and learning rate)

Table 2 summarizes the MAP@5 results for each configuration. DenseNet121 achieved its best generalization after full tuning (0.8575 / 0.8130 train/test), while VGG16 outperformed ResNet50 under identical conditions. SET-CNN delivered the highest performance (0.9286 / 0.8745), with a minimal train–test gap, confirming robust generalization.

Table 2. Impact of tuning strategies on training and testing performance of base models and the proposed SET-CNN model in Terms of MAP@5

Tuning Step	DenseNet121		ResNet50		VGG16		SET-CNN (Proposed)	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Fine-tuning	0.7388	0.6813	0.6986	0.6373	0.7783	0.7094	0.9682	0.8521
Optimizer	0.7572	0.6972	0.7441	0.6592	0.7901	0.7258	0.9867	0.8601
Learning Rate	0.8710	0.7209	0.7510	0.6809	0.8172	0.7620	0.9902	0.8657
Dropout Rate	0.8392	0.7451	0.7105	0.6929	0.8015	0.7799	0.8919	0.8662
Loss Function	0.8321	0.7750	0.7414	0.7200	0.8160	0.7910	0.9143	0.8699
Batch Size	0.8400	0.8021	0.7629	0.7810	0.8437	0.8162	0.9227	0.8701
Epochs	0.8575	0.8130	0.8210	0.7909	0.8516	0.8207	0.9249	0.8729
Early Stopping	0.8505	0.8132	0.8244	0.7915	0.8554	0.8270	0.9286	0.8745

Figure 3 shows that SET-CNN consistently outperformed base models at every tuning stage, and Figure 4 highlights its superior training and testing accuracies, with fine-tuning boosting test accuracy by 17–22% across all models.

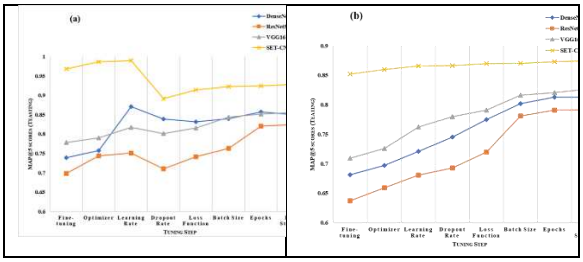


Fig. 3. Comparison of base and proposed models for different tuning steps for (a) training and (b) testing phases, evaluated using MAP@5 scores

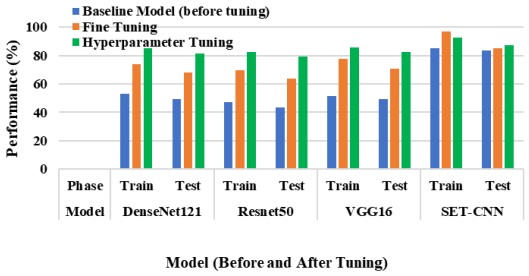


Fig. 4 Overall performance of different models before and after applying fine-tuning and hyperparameter tuning strategies.

4.3. Training and Validation Loss

Monitoring loss curves provides insights into convergence behavior and generalization. Figure 5 shows training and validation loss trajectories:

- DenseNet121 *(Fig. 5a)*: Steady decline with minimal gap → good generalization.
- ResNet50 *(Fig. 5b)*: Sharp early drop, plateau around epoch 14 → stable convergence.
- VGG16 *(Fig. 5c)*: Downward trend but with fluctuations → potential sensitivity to hyperparameters.
- SET-CNN *(Fig. 5d)*: Smooth, consistent decline with close alignment of curves → best regularization and generalization.

SET-CNN achieved the lowest and most stable final loss, followed by ResNet50 and DenseNet121. VGG16 showed higher variability, suggesting possible overfitting.

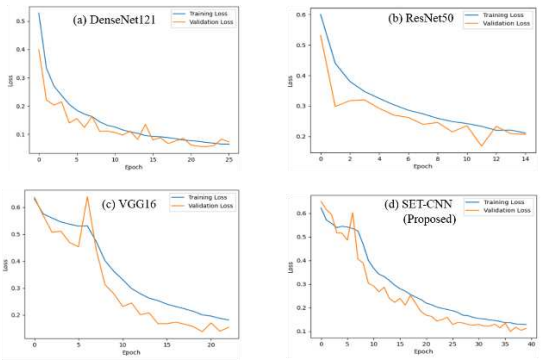


Fig. 5 Training and validation loss curves of the baseline models (a) DenseNet121, (b) ResNet50, (c) VGG16, and the proposed model (d) SET-CNN.

4.4. K-Means Clustering and t-SNE Visualization of Image Embeddings

K-Means clustering was applied to embeddings from each model, and t-SNE was used for 2D visualization (Figure 6):

- DenseNet121 *(Fig. 6a)*: Well-formed clusters with some overlaps.
- ResNet50 *(Fig. 6b)*: More dispersed with less compact class separation.
- VGG16 *(Fig. 6c)*: Distinct clusters with occasional blending.
- SET-CNN *(Fig. 6d)*: Most compact and well-separated clusters, indicating high intra-class similarity and inter-class separation.

These results align with MAP@5 performance, confirming SET-CNN's advantage in generating discriminative embeddings.

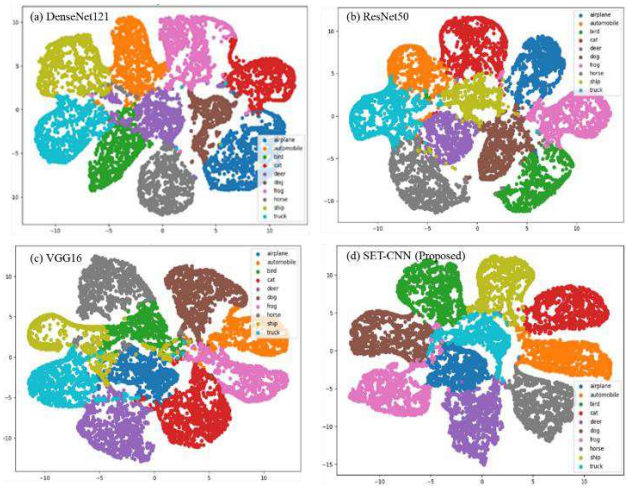


Fig. 6 Embedding cluster visualizations for (a) DenseNet121, (b) ResNet50, (c) VGG16, and (d) SET-CNN using K-Means clustering and t-SNE techniques.

4.5. Similar Image Retrieval Performance

Retrieval was evaluated using KNN with cosine similarity (Figure 7). For each query, the top-10 most similar images were displayed, with mismatches marked in red.

- DenseNet121: Accurate overall, but occasional semantic mismatches (e.g., truck for car, dog for cat).
- ResNet50: Similar performance, with occasional category confusion.
- VGG16: Strong results, comparable to DenseNet121 and ResNet50.
- SET-CNN: Most accurate and semantically relevant retrievals, with minimal confusion, benefiting from feature fusion and meta-learning.

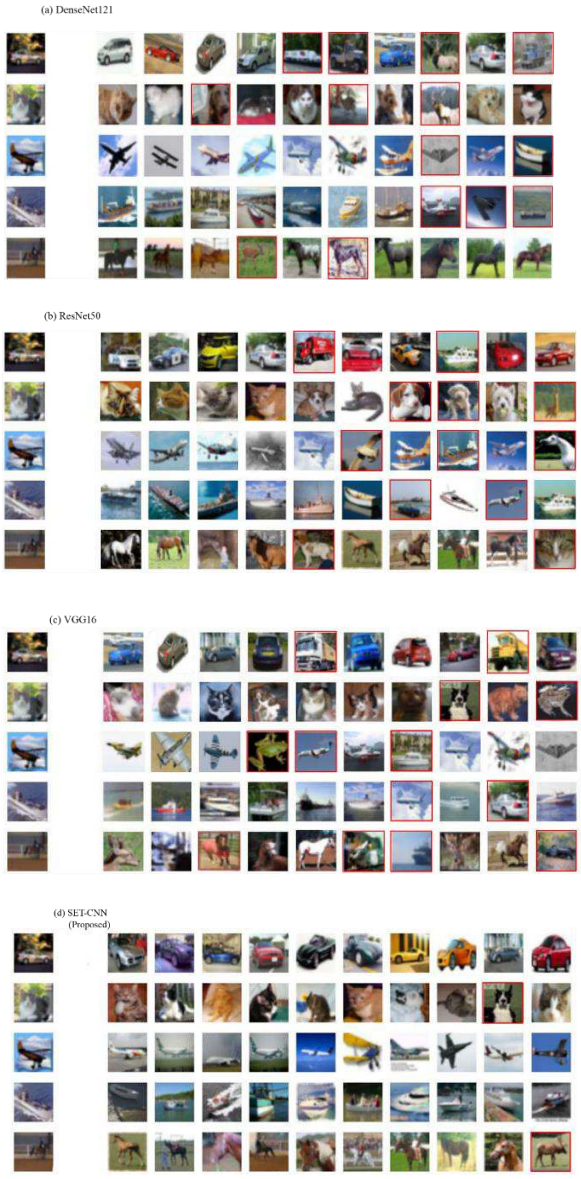


Fig. 7 Similar images retrieved by (a) DenseNet121, (b) ResNet50, (c) VGG16, and (d) the proposed approach (SET-CNN)

4.6. Comparison with State-of-the-Art Approaches

On CIFAR-10 (Table 1), SET-CNN achieved 0.8745 MAP@5, outperforming:

- DSRH: +8.6%
- DSH: +19.9%
- CNN + KNN: +4.9%

While attention-augmented ResNet variants report higher scores (up to 95.7%), SET-CNN matches competitive performance without architecture-specific modifications. Its minimal train–test gap reflects robust generalization.

5. Conclusion

This research work presented SET-CNN, a novel stacking ensemble framework that integrates DenseNet121, ResNet50, and VGG16 through feature-level fusion, followed by a meta-model trained with triplet semi-hard loss to produce discriminative and semantically rich embeddings for image retrieval. By leveraging the complementary strengths of diverse CNN architectures and optimizing for retrieval-specific objectives, SET-CNN addresses the limitations of single-network approaches and domain-specific tuning carried out in past research.

On the CIFAR-10 dataset, SET-CNN achieved a testing MAP@5 of 87.45%, outperforming the best individual CNN model by 7.6% and surpassing state-of-the-art hashing-based methods such as Deep Semantic Ranking Hashing (+8.6%) and Deep Supervised Hashing (+19.9%). These quantitative improvements were substantiated by qualitative analyses: K-Means clustering combined with t-SNE visualization highlighted enhanced embedding separability, while KNN-based retrieval using cosine similarity confirmed increased semantic relevance.

SET-CNN's novelty lies in its meta-model-driven ensemble integration, retrieval-oriented loss optimization, and architecture-agnostic design, making it robust, scalable, and adaptable to various domains. Looking ahead, future research will explore its application to higher-resolution datasets, real-time retrieval pipelines, and cross-domain multimodal tasks, advancing the development of generalizable and high-performing image retrieval systems.

Author contributions: CRediT

Rabia Maqsood: Writing – original draft, Validation, Resources, Methodology, Investigation, Experiments. Muhammad Omar: Writing – review & editing, Supervision. Safdar Ali: Writing – review & editing. Rahat H. Bokhari: Writing – review & editing.

Funding sources

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Acknowledgments

The authors thank colleagues and peers for their valuable discussions and constructive feedback during the preparation of this work. Portions of the manuscript benefited from the use of AI-assisted editing tools for language refinement and formatting.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability

The datasets used in this study are publicly accessible from their respective original sources, as cited in the manuscript. The full implementation of the SET-CNN framework, including model training scripts, hyperparameter configurations, and evaluation procedures, is openly available at: https://github.com/RabiaMaqsood-01/Image-Embedding-through-CNN-Models-and-Stacking-Ensemble-Technique.

References

1. Makadia A, Pavlovic V, Kumar S (2008) A New Baseline for Image Annotation. In: European Conference on Computer Vision (ECCV)

2. Rodriguez-Serrano J, Larlus D, Dai Z (2015) Data-Driven Detection of Prominent Objects. IEEE Trans Pattern Anal Mach Intell

3. Krizhevsky A, Sutskever I, Hinton GE (2017) ImageNet Classification with Deep Convolutional Neural Networks. Commun ACM 60:84–90. https://doi.org/10.1145/3065386

4. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

5. Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

6. Karpathy A, Joulin A, Fei-Fei L (2014) Deep Fragment Embeddings for Bidirectional Image-Sentence Mapping. In: Advances in Neural Information Processing Systems (NeurIPS)

7. Mikulík A, Perdoch M, Chum O, Matas J (2010) Learning a Fine Vocabulary. In: European Conference on Computer Vision (ECCV)

8. Tolias G, Jégou H (2015) Visual Query Expansion with or without Geometry: Refining Local Descriptors by Feature Aggregation. In: Pattern Recognition

9. Ding Y, Zhang Z, Zhao X, et al (2022) Self-Supervised Locality Preserving Low-Pass Graph Convolutional Embedding for Large-Scale

Hyperspectral Image Clustering. IEEE Transactions on Geoscience and Remote Sensing 60:. https://doi.org/10.1109/TGRS.2022.3198842

10. Mo D, Wong WK, Liu X, Ge Y (2022) Concentrated Hashing with Neighborhood Embedding for Image Retrieval and Classification. International Journal of Machine Learning and Cybernetics 13:1571–1587. https://doi.org/10.1007/s13042-021-01466-7

11. Li Z, Tang J, Mei T (2019) Deep Collaborative Embedding for Social Image Understanding. IEEE Trans Pattern Anal Mach Intell 41:2070–2083. https://doi.org/10.1109/TPAMI.2018.2852750

12. Barz B, Denzler J (2019) Hierarchy-based Image Embeddings for Semantic Image Retrieval. In: IEEE Winter Conference on Applications of Computer Vision (WACV). pp 638–647

13. Shen L, Pauly J, Xing L (2022) NeRP: Implicit Neural Representation Learning with Prior Embedding for Sparsely Sampled Image Reconstruction. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2022.3177134

14. Shao S, Cui Q (2022) 1st Place Solution in Google Universal Images Embedding

15. Yao T, Long F, Mei T, Rui Y (2016) Deep Semantic-Preserving and Ranking-Based Hashing for Image Retrieval. In: IJCAI International Joint Conference on Artificial Intelligence. pp 3931–3937

16. Liu H, Wang R, Shan S, Chen X (2016) Deep Supervised Hashing for Fast Image Retrieval. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2064–2072. https://doi.org/10.1109/CVPR.2016.227

17. Trinh TH, Luong M-T, Le Q V (2019) Selfie: Self-supervised Pretraining for Image Embedding

18. Razavian AS, Azizpour H, Sullivan J, Carlsson S (2014) CNN Features Off-the-shelf: An Astounding Baseline for Recognition. In: CVPR Deep Vision Workshop

19. Gong Y, Wang L, Guo R, Lazebnik S (2014) Multi-Scale Orderless Pooling of Deep Convolutional Activation Features. In: European Conference on Computer Vision (ECCV)

20. Mascarenhas S, Agarwal M (2021) A Comparison between VGG16, VGG19 and ResNet50 Architecture Frameworks for Image Classification. In: IEEE International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON). pp 96–99

21. Ye M, Zhang X, Yuen PC, Chang SF (2019) Unsupervised Embedding Learning via Invariant and Spreading Instance Feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp 6210–6219

22. Rashid M, Kamruzzaman J, Imam T, et al (2022) A Tree-Based Stacking Ensemble Technique with Feature Selection for Network Intrusion Detection. Applied Intelligence 52:9768–9781. https://doi.org/10.1007/s10489-021-02968-1

23. Hassan M, Ali S, Alquhayz H, et al (2022) Developing Liver Cancer Drug Response Prediction System Using Late Fusion of Reduced Deep Features. Journal of King Saud University - Computer and Information Sciences 34:8122–8135

24. Sinaga KP, Yang MS (2020) Unsupervised K-means Clustering Algorithm. IEEE Access 8:80716–80727. https://doi.org/10.1109/ACCESS.2020.2988796