Multi-Level CNN Feature Fusion from ResNet50 for Near-Duplicate Image Detection in Real Estate Imagery

Taras Panchenko[1*], Artem Bozhok[1,2] and Volodymyr Kubytskyi[1]
[1]Faculty of Computer Science and Cybernetics, Taras Shevchenko National University of Kyiv, 64/13, Volodymyrska Street, 01601 Kyiv, Ukraine
[2]LUN.ua
E-mail: taras.panchenko@knu.ua
[*]Corresponding author

Keywords: image vector representation, image descriptor, near-duplicate images, image near similarity, convolutional neural network, intermediate layers, ResNet50, image embedding

Received: September 25, 2025

The volume of images uploaded to the internet is increasing at an unprecedented rate, making image deduplication, through accurate near-duplicate detection, a critical task in computer vision. However, comparing images for similarity remains challenging due to complex visual structures and subtle appearance variations.

We propose a novel embedding method for image similarity detection. It constructs an enriched representation by concatenating outputs from multiple intermediate layers of a pre-trained ResNet50 convolutional neural network and trains a lightweight decision network on top to classify image pairs. Unlike aggregation approaches that average or sum intermediate features, our method preserves both low-level and high-level information in a single descriptor and maintains feature diversity. The multi-level embedding is further normalized to balance feature contributions and is evaluated against classical keypoint descriptors, a DCT-based perceptual hash, and a standard single-layer ResNet50 embedding.

We evaluate this method on three real-world image deduplication tasks derived from real estate listings, covering (a) near-identical property photos with graphical overlays, (b) interior room photographs captured from different angles, and (c) schematic floor plan images. The proposed embedding achieves F1-scores of 0.96, 0.87, and 0.77, representing a 10-15% absolute improvement over baseline methods, including classical feature descriptors and standard ResNet50 final-layer embeddings.

This approach has been successfully deployed in production on a large-scale real estate platform, reducing duplicate images and improving search quality. The results demonstrate that multi-layer CNN embeddings with explicit feature preservation offer a robust and scalable solution for near-duplicate image detection in structured domains such as real estate photography and schematic floor plans.

Povzetek: Predlagan je nov pristop za primerjanje slik, ki izboljša zaznavanje podvojenih slik in se izkaže kot učinkovit v praksi.

1 Introduction

Recent advancements in artificial intelligence have driven significant progress in image processing technologies. For example, generative models like OpenAI's DALL-E [1] and DALL-E 2 [2] can create realistic images and art from text prompts. At the same time, AI-based vision models have achieved or even surpassed human-level accuracy on certain image recognition benchmarks [3]. Despite these breakthroughs, there remain many practical image-focused tasks beyond generation and classification that are crucial for industry. One such problem is image deduplication, which relies on near-duplicate detection methods and image similarity modeling [4-7]. This task involves determining whether two images depict the same content or scene, even when they differ in resolution, perspective, lighting, or undergo minor edits. Efficient solutions to image deduplication are increasingly important as internet platforms deal with billions of images, where removing duplicates can save storage and improve user experience.

On real-estate platforms, near-duplicate photos are common; reliable detection improves storage efficiency and user experience.

Several public benchmarks define the problem space for near-duplicate detection, including the DISC21 dataset for large-scale copy detection tasks. These datasets capture common real-world transformations such as cropping, overlays, and perspective shifts, and provide standardized baselines for evaluating and comparing methods.

However, reliably detecting near-duplicate images is challenging. Images may undergo various transformations (scaling, rotation, color changes, etc.),

and defining a robust measure of visual similarity is non-trivial. Early approaches relied on hand-crafted feature descriptors such as SIFT [8, 9], SURF [10], and ORB [11]. These methods are effective for local structure matching but exhibit reduced robustness when confronted with complex, high-level patterns and real-world variations. These limitations motivated the shift toward learned representations. Numerous research efforts have extended these ideas, including geometric invariant features in transform domains [12] and perceptual hashing techniques [13,14]. However, existing approaches still show reduced accuracy and robustness under real-world transformations such as rotations, overlays, and scale variations.

In the last decade, deep learning approaches have revolutionized image representation learning [15-42]. Convolutional Neural Networks (CNNs) in particular can automatically learn rich feature hierarchies from images, and their use for image representation has become widespread. Typically, a pre-trained CNN (e.g., ResNet [37]) is used as a feature extractor by taking the output of its final layer (or an intermediate pooling layer) as a global image descriptor. CNN-based embeddings have consistently outperformed hand-crafted features in image retrieval and recognition tasks. For instance, the output of the penultimate layer of a pre-trained ResNet50 [37] (2048-dimensional feature) can serve as a descriptor for an image pair similarity comparison. Relying solely on a network's classifier head to represent images is suboptimal for similarity. That stage is optimized to separate categories, not to preserve nuanced cues that distinguish look-alikes.

In this paper, we address these shortcomings by proposing a novel image embedding technique that leverages intermediate CNN layer outputs. By combining features from multiple depths of the network, we obtain a more descriptive vector representation that encapsulates both low-level and high-level image content.

$$f_{inter} = \phi(Flatten(A^{(l_1)}), \ldots, Flatten(A^{(l_k)})) \quad (1)$$

where: f_{inter} - feature map from layer l_i (tensor of size $C_i \times H_i \times W_i$), $Flatten(\cdot)$ - operation of tensor alignment into a 1D vector, $\phi(\cdot)$ - aggregating function (Concatenate, AveragePool, MaxPool), k - number of intermediate layers used, f_{inter} - image feature vector built from convolutions.

We apply this representation to the image deduplication problem and demonstrate significant improvements in detection accuracy (as measured by F1-score) over the evaluated baselines: classical keypoint descriptors (SIFT/SURF/ORB), a DCT-based perceptual hash, and a single-layer ResNet50 embedding [4, 5, 31-37]. We validate our approach on three distinct use-cases of image similarity in the real estate domain, showing that our method can robustly handle different types of near-duplicate image scenarios. Moreover, our solution has been deployed in a production environment (at a large real estate listing company) to automatically eliminate duplicate images, underlining its practical relevance.

In this work, we use the term image deduplication to refer to the practical task of detecting and removing redundant or duplicate content from large-scale image collections. Near-duplicate detection refers more specifically to identifying images of the same scene or object that differ slightly due to edits, overlays, or viewpoint changes. Image similarity is the broader technical problem underlying both, which involves quantifying how visually alike two images are.

2 Related work

Existing research on near-duplicate image detection spans classical descriptors, perceptual hashing, and deep learning-based embeddings. Below, we summarize key approaches and highlight the gaps our work addresses.

Hand-Crafted Feature Methods: Early approaches to near-duplicate image detection relied on hand-crafted visual features and descriptors. One of the most influential classical methods is the Scale-Invariant Feature Transform (SIFT) by Lowe [8], which detects salient local keypoints and encodes them as invariant descriptors. SIFT features, and related descriptors like SURF [10] and ORB [11], have been successfully used to match images that contain identical objects or scenes, even under moderate transformations. For example, Chum et al. [9] demonstrated a system for finding near-identical images by efficiently matching SIFT keypoints in a large database. Other works improved robustness by incorporating geometric verification or by focusing on specific invariant features. Ke et al. [15, 16] proposed methods for fast near-duplicate detection and sub-image retrieval that break images into parts and use local feature matching with optimizations for speed. While traditional feature-based methods remain effective for simple transformations, their precision and recall degrade substantially when handling appearance variations or tasks requiring high-level semantic matching. A recent survey by Thyagharajan and Kalaiarasi [4] provides a comprehensive review of image near-duplicate detection techniques, concluding that while keypoint-based methods are useful, they may not suffice for more complex similarity tasks involving, for instance, different perspectives or non-identical but related content.

Global and Perceptual Hashing Techniques: Another line of work for image deduplication involves computing compact global signatures (or "hashes") for images that preserve visual similarity. Perceptual hashing methods, such as those based on Discrete Cosine Transform (DCT) or wavelet transforms, compress an image into a short binary code that changes only slightly for minor image modifications. One popular approach is the average hash or DCT hash, which was used as a baseline in our experiments. A DCT-based hash is obtained by resizing an image, applying the DCT, and quantizing the low-frequency coefficients into bits [13, 14]. The resulting binary hash can be compared via Hamming distance - a small Hamming distance indicates two images are likely near-duplicates. Perceptual hashes are computationally efficient, enabling rapid filtering of clearly dissimilar images in large-scale datasets. For

instance, if an image's hash differs significantly from a query hash, that image can be immediately excluded from further consideration, as it is unlikely to be a duplicate. Systems like Facebook's image DNA and pHash leverage such techniques to handle web-scale image matching. However, these hashes offer limited discriminative power; they are prone to collisions and insensitive to fine-grained differences, resulting in reduced recall for near-duplicate detection. Thus, perceptual hashing is often used in combination with more detailed comparisons.

Classical and Learning-based Similarity Measures: Beyond local features and hashing, researchers have explored various other strategies. Some methods use statistical measures and entropy of pixel patterns [18, 19] or exploit geometric invariants, such as features in the Radon transform domain [12], to identify duplicates from transformed images. Others applied bag-of-visual-words models and indexing structures for scalable retrieval of duplicate images [21]. Liu et al. [22], for example, introduced a variable-length signature for image matching, adapting the representation length to image content. Clustering techniques have also been employed; Xie et al. [23] used affinity propagation clustering on deep features to rapidly narrow down candidates for near-duplicates in a web-scale ImageWeb database. These methods contributed useful ideas (e.g., using dimensionality reduction, indexing, clustering) to tackle the efficiency challenges in large-scale duplicate search. Nonetheless, the advent of deep learning brought a substantial improvement in representation quality, as described next.

CNN-Based Embeddings for Near-Duplicates: With deep CNNs achieving high results in image recognition, they naturally became attractive for image similarity tasks. A straightforward approach is to use features from a CNN pre-trained on ImageNet [3] as image descriptors. For instance, a common baseline is to take the 2048-dimensional output of the global average pooling layer of ResNet50 (which precedes the classification layer) as the image's feature vector. In our experiments, we use this as one baseline, referred to as the "ResNet50 embedding" method. Such deep features generally outperform hand-crafted features on retrieval and matching tasks because they encode more semantic information. However, they may sometimes be too coarse for distinguishing near-duplicates, as they emphasize high-level content over low-level differences. Recent studies have proposed multiple enhancements to CNN-based embeddings for near-duplicate image detection. Zhou et al. [31] proposed a coarse-to-fine scheme that uses both global CNN features and local region features: an initial comparison on global features finds candidates, and then a more detailed local feature matching (using CNN-based keypoints) confirms the duplicates. This approach enables real-time near-duplicate detection by eliminating costly comparisons between dissimilar pairs. Kordopatis-Zilos et al. [32] introduced the idea of aggregating multiple intermediate CNN layer outputs for near-duplicate video frame retrieval. By summing or concatenating feature maps from different layers of a CNN, their system captured multi-level characteristics of frames and improved retrieval accuracy. Our approach is inspired by a similar intuition, though we apply it to static images and use concatenation of intermediate layers in ResNet50 to form a single descriptor. Zhang et al. [33] tackled a related problem of cross-modal (infrared vs. visible light) near-duplicate image detection by comparing CNN embeddings after aligning images with a spatial transformer network, illustrating the breadth of scenarios where learned embeddings are useful.

Transformers and Other Modern Approaches: In 2023 and 2024, new state-of-the-art techniques have emerged for image copy detection and similarity, taking advantage of advances in deep learning architectures. Notably, some works incorporate Vision Transformers (ViT) and Siamese networks to improve robustness. Fawzy et al. [43] combined a ViT-based feature extractor with dynamic data augmentation and efficient sampling to excel in an image copy-detection challenge. Their method augmented images and learned transformer-based embeddings that proved more invariant to edits and noise, achieving top accuracy on the DISC21 copy-detection dataset. Other researchers have integrated traditional perceptual hashing with deep learning: for example, by using a ViT to generate perceptual hash codes [44] or by using a Siamese CNN that learns to predict if two images are duplicates, then refining that with a hash filter [45]. Another frontier is the exploration of quantum algorithms for image similarity: Yang et al. proposed a quantum inner-product method to compute image similarities more efficiently, encoding classical CNN-derived feature vectors into quantum states. While such approaches are still experimental, they indicate the growing interest in accelerating and improving near-duplicate detection through unconventional means.

In summary, the field has evolved from keypoint matching and hashing toward learned embeddings and hybrid systems that integrate efficiency with accuracy. Yet, there remains a gap in achieving both high recall and precision across diverse image transformations in deployment-constrained settings. Our work contributes to this landscape by introducing an intermediate-layer CNN embedding approach evaluated against baselines that are commonly used or straightforward to integrate in industrial deduplication pipelines (classical keypoints, DCT-based perceptual hashing, and a single-layer ResNet50 embedding). Rather than competing directly with the latest transformer-based or task-specific copy-detection systems, we focus on quantifying and explaining the gains obtained by multi-layer CNN feature fusion over these widely adopted baselines in the context of real estate imagery.

To contextualize our approach, Table 1 summarizes representative families of methods used in near-duplicate image detection. For clarity, the table also includes an illustrative F1-score measured in our own evaluation on Task A (near-identical images with graphical overlays), using a consistent train/validation/test protocol. These values reflect evaluation on our dataset and are included to show how the method families behave under the same

conditions. The full quantitative comparison across Tasks A-C, with confidence intervals, is reported in Table 3.

Table 1: Representative method families for near-duplicate image detection, with Task A performance from our evaluation

Method	Feature Type	Robust to Transformations	Application Domain	F1-score (Task A)	Score source	Key Limitation
Classical SIFT/SURF/ORB	Local keypoints (hand-crafted)	Moderate (scale/rotation only)	General image matching/retrieval	0.81	This work (reimplementation)	Limited semantic understanding
Perceptual Hash	Global binary hash	Low (sensitive to overlays, edits)	Web-scale deduplication	0.70	This work (reimplementation)	Poor fine-grained discrimination
Single-layer ResNet50	Deep features (penultimate layer)	High for natural images	Image retrieval, deduplication	0.85	This work (reimplementation)	Misses low/mid-level details
Proposed method	Multi-layer ResNet50 + decision network	High (low-mid-high features)	Real estate (photos + floor plans)	0.96	This work (proposed)	Higher compute cost but scalable

3 Methodology

Our proposed solution for image deduplication is centered on a multi-layer CNN embedding that serves as a robust image descriptor, coupled with an efficient workflow for comparing image pairs. The methodology is organized into three stages: feature extraction from a pre-trained CNN, embedding construction (concatenation and normalization), and similarity decision modeling. Efficient storage and indexing are described separately in the efficiency subsection. Figure 1 summarizes the full pipeline and shows how these stages connect.

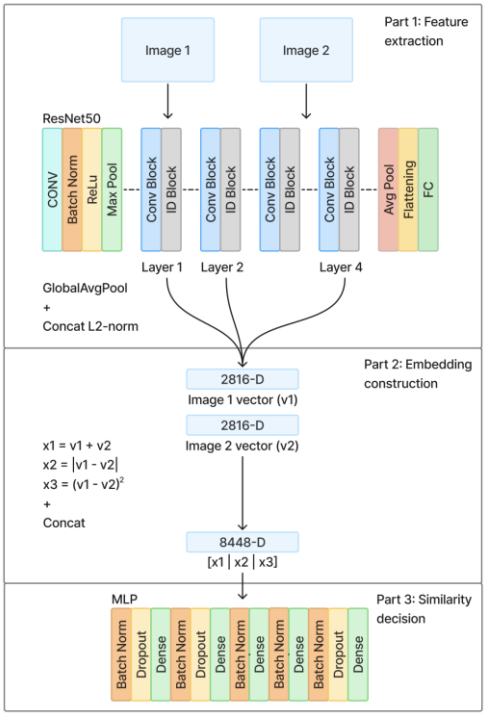

Figure 1: Pipeline overview

Algorithm 1 summarizes the complete pipeline in a reproducible, implementation-agnostic form.

Algorithm 1: End-to-end pipeline pseudocode

Input: labeled image pairs (I_a, I_b, y).

Output: trained decision model and inference procedure.

(1) Feature extraction (frozen backbone)

For each image I:

1. Decode to RGB (3 channels). If an input is single-channel (e.g., some floor plans), replicate to 3 channels.
2. Resize to 224×224 (bilinear interpolation).
3. Apply ImageNet normalization per channel: mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225].
4. Run frozen ResNet-50 (IMAGENET1K_V2) and tap stages C2=layer1, C3=layer2, C5=layer4.
5. Apply global average pooling to each tapped feature map to obtain vectors v_{C2}, v_{C3}, v_{C5}.
6. Concatenate $e = [v_{C2}; v_{C3}; v_{C5}]$ (2816-D).
7. L2-normalize $e \leftarrow \frac{e}{\|e\|_2}$. Store e.

(2) Pair fusion

Given embeddings e_1, e_2:

$z = [e_1 + e_2; |e_1 - e_2|; (e_1 - e_2)^2]$ (8448-D).

(3) Decision model training

1. Train the MLP on z with BCE loss and Adam (lr =10^{-5}), batch size 64, up to 30 epochs, using the specified class weights.
2. Select the best checkpoint by validation accuracy.
3. Choose decision threshold τ on validation to maximize F1; evaluate once on the held-out test split.

(4) Inference

For a query image I_a: compute embedding e_a. For each candidate image I_i (optionally shortlisted by a hash pre-filter), compute $p = MLP(z(e_a, e_i))$. Predict duplicate if $p \geq \tau$.

The remaining subsections detail each stage of Algorithm 1.

3.1 Image embedding construction using intermediate CNN layers

To build the proposed vector image representations, a pre-trained ResNet50 [3] CNN - originally meant for image classification - is employed. The model's final output layer is removed, and activations from selected intermediate layers are extracted and concatenated into a unified feature vector, which serves as the image embedding. This vector is then normalized so that the magnitude of any one feature (numerical component) does not dominate the similarity measure. The resulting representation is robust to geometric transformations, illumination changes, noise, and watermarks, enabling reliable similarity-based comparisons.

$$f = \frac{f}{\sqrt{\sum_{i=1}^{d} f_i^2 + \epsilon}} \quad (2)$$

where: f - is the initial feature vector, f_i - i-th component of the vector, d - vector dimension, ϵ - small addition to avoid division by zero (e.g., 10-8), f - normalized vector (unit length).

The detailed steps for creating vector representations of images using a combination of intermediate layers from a pre-trained ResNet50 are outlined below.

1. Load the pre-trained ResNet50 model.

The first step is to load a ResNet50 model that has been pre-trained on a large-scale image-classification dataset such as ImageNet [3]. This allows us to reuse the learned weights to extract high-, mid-, and low-level features from our input images.

2. Remove the final fully-connected layer.

Because the pre-trained ResNet50 is meant for image classification, its last layer is a fully-connected head that outputs class probabilities. We discard this layer, as it is not needed for building an image embedding.

3. Obtain activations from intermediate layers.

We extract the activations of ResNet50's intermediate layers, since each layer captures features at a different level of abstraction. These activations can be viewed as feature maps that reveal how strongly each learned filter responds to the input. To gather them, we pass the image through the network and record the activations at every chosen intermediate layer.

Selecting which layers to extract:

The choice of intermediate layers depends on the required level of abstraction and feature complexity for the specific task. In this work, we tap three stages of the torchvision ResNet-50: layer1 (C2), layer2 (C3), and layer4 (C5). For each stage's activation tensor (with channel counts 256, 512, and 2048, respectively), we apply global average pooling over spatial dimensions to obtain vectors of sizes 256, 512, and 2048. We then concatenate these into a 2816-D descriptor and apply L2-normalization. This choice provides a balanced mix of low-level (edges/textures), mid-level (geometric parts/layout), and high-level (object/scene) cues, while keeping the descriptor compact enough for large-scale indexing.

$$f^{(l)} = GlobalAvgPool(A^{(l)}), A^{(l)} \in R^{C_l \times H_l \times W_l} \quad (3)$$

where: $A^{(l)}$ - feature map on the layer l with C_l channels, height H_l and width W_l, $GlobalAvgPool$ - global average convolution along spatial axes, $f^{(l)}$ - vector R^{C_l}, that aggregates the layer.

For comparison, the single-layer baseline that uses only the penultimate stage of ResNet-50 produces a 2048-D embedding. Because memory usage scales with embedding size, the proposed 2816-D descriptor increases storage by approximately 1.38×. Distance computations in ANN indexes also scale roughly linearly with dimensionality, so the per-comparison arithmetic cost increases by a similar factor. Overall latency is mitigated by candidate selection via a hash pre-filter, after which only a small subset undergoes full vector comparison (see "Efficient Embedding Storage and Indexing").

Empirical layer selection and ablation (overview). We evaluated several candidates tap sets (e.g., {C2,C3,C4}, {C2,C3,C5}, {C2,C4,C5}) on a held-out validation split and selected {C2,C3,C5} as the default configuration. A full ablation study quantifying the contribution of each stage and their combinations - single-stage, two-stage, and three-stage descriptors - is reported in "Experiments and Results" (Tables 6-7), including per-task F1 and threshold-independent ROC-AUC/PR-AUC for Task A.

4. Merging the intermediate-layer activations.

The selected intermediate activations are aggregated into a single vector that serves as the image representation. We evaluated several aggregation strategies, including averaging/max pooling, Generalized Mean (GeM) pooling, and NetVLAD. We adopt concatenation of per-layer GAP descriptors because it is training-free and stable across heterogeneous data (photos and schematic floor plans), preserves complementary cues from different depths, and has predictable latency and memory costs for large-scale indexing. In pilot experiments on our validation split, GeM and NetVLAD did not yield consistent improvements over concatenation and added compute and implementation overhead; therefore, we use concatenation in the main method.

In summary, our vector representation leverages a pre-trained CNN, removes its classification head, and concatenates the chosen intermediate features (after per-layer global average pooling) into one normalized feature vector used for similarity.

Such embeddings support many comparison scenarios, including:

- Image retrieval. Given a query image, its vector can be used to retrieve similar images from a database.
- Similarity measurement. Two embeddings can be compared directly to quantify how alike the underlying images are.
- Classification. The vector can serve as input to a classifier that assigns the image to predefined categories.

3.2 Efficient embedding storage and indexing

For efficient similarity search, it is important to determine how to index image representations in a database. Indexing enables efficient retrieval and can be implemented using algorithms such as KD-trees, Ball trees, Annoy hash comparison, or DCT-based methods [13].

A DCT hash comparison can serve as a preliminary filter when working with multi-million-image databases, removing images that differ substantially from the query. A DCT hash is a compact, reliable image descriptor obtained by applying the discrete cosine transform (DCT) to a small, down-sampled version of the image. The resulting DCT coefficients are quantized and combined into a single DCT hash [14].

DCT hashes are compared by computing the bitwise distance between the query image's DCT hash and the DCT hashes stored in the database. This distance is typically measured with the Hamming metric - the number of bit positions at which the two binary strings differ.

$$b_i = sign(DCT(x_i) - \mu_{DCT}), d_H(b_1, b_2) = \\ = \sum_{j=1}^{n} I[b_1^{(j)} \neq b_2^{(j)}] \quad (4)$$

where: x_i - pixel or block of an image, $DCT(x_i)$ - coefficient after cosine transformation, μ_{DCT} - average value of DCT coefficients, b_i - the corresponding hash bit (0 or 1), d_H - Hamming distance between hashes, $I[\cdot]$ - indicator function (1, if true).

Images with large Hamming distances are treated as dissimilar and excluded from further embedding comparisons. Doing so greatly reduces the search space and therefore improves efficiency.

3.3 Similarity decision model

To compare two images, we use their embeddings $e_1, e_2 \in R^d$ and construct a fused pair representation using three element-wise channels: $e_1 + e_2$ (sum), $|e_1 - e_2|$ (absolute difference), and $(e_1 - e_2)^2$ (squared difference). These three channels are concatenated into a single vector $z \in R^{3d}$. With the proposed multi-layer descriptor $d = 2816$, the decision network input dimension is therefore $3d = 8448$.

Backbone usage. The ResNet-50 backbone is used as a fixed feature extractor: it is frozen and not fine-tuned during decision-model training. We use torchvision ResNet-50 pretrained weights IMAGENET1K_V2. The MLP is trained on pairs of precomputed embeddings.

Decision network architecture. The fused vector z is fed to a fully connected MLP with Batch Normalization (BN) after each Dense layer, ELU activations in hidden layers, and a final sigmoid producing a match probability $p \in [0, 1]$. The architecture used in all experiments is specified in Table 2.

Table 2: Decision MLP architecture

Block	Operation	Output dim	Activation	Dropout
Input	Pair-fusion concat $[e_1 + e_2, \lvert e_1 - e_2 \rvert, (e_1 - e_2)^2]$	8448	-	-
Preprocess	BN	8448	-	-
FC1	Dropout → Dense → BN	2048	ELU	0.50
FC2	Dropout → Dense → BN	1024	ELU	0.50
FC3	Dense → BN	512	ELU	-
FC4	Dense → BN → Dropout	256	ELU	0.25
Output	Dense	1	Sigmoid	-

Training objective and optimization. The model is trained with binary cross-entropy (BCE). We use Adam with a learning rate 1×10^{-5}, batch size 64, and train for 30 epochs. We select the best checkpoint based on validation accuracy (ModelCheckpoint with $monitor = val\_accuracy$, $mode = max$), and use that checkpoint for evaluation.

Class imbalance handling. We apply class weighting during training. The weights used are: class 0 weight $= 0.9947$, class 1 weight $= 1.0053$, indicating a near-balanced training split (approximately negative:positive $\approx 1.01:1$ under standard "balanced" class-weight computation).

Inference and thresholding. At inference time, the network outputs a probability p. The decision threshold τ is selected on the validation split (maximizing F1 for the corresponding task) and then fixed for evaluation on the held-out test split.

4 Experiments and results

We evaluated the proposed method on three real-world image deduplication scenarios, drawn from an industry dataset of real estate images. These scenarios correspond to distinct sub-problems frequently encountered in practice:

Task A: Near-identical images with different contexts. These are image pairs that are essentially the same photograph or graphic, but one might have additional graphical elements (such as watermarks, borders, or slight cropping differences). Examples of such image pairs are shown in Figure 2. We assembled a dataset of 10,000 such image pairs [47], with labels indicating duplicates vs. non-duplicates. This dataset is publicly available and is the benchmark used for Task A; the corresponding results are reported in Table 3. Threshold-independent ROC-AUC and PR-AUC for Task A are also reported in Table 4.

Figure 2: An example of near-identical images with different context

Task B: Multi-angle images of rooms. Here, the goal is to detect if two photos are of the same room or interior space, even if taken from different viewpoints or at different times. This task is challenging because, although the images are not pixel-identical, they depict the same physical space and can be recognized as such by humans. An example of such image pairs are shown in Figure 3. Our test dataset contains 12,500 image pairs of interiors, covering both matching pairs (same room) and non-matching pairs (different rooms), based on an apartment listing database.

Figure 3: An example of multi-angle images of rooms

Task C: Schematic floor plan matching. In this task, we deal with binary or line-drawing floor plan images. Two floor plans should be detected as duplicates if they represent the same layout, even if their orientations, scale, or slight drawing details differ. An example of such image pairs are shown in Figure 4. The test dataset includes 8,800 pairs of floor plan images, with ground truth on which pairs depict the same property layout.

Figure 4: An example of the same schematic floor plans

Dataset characterization, labeling, and reproducibility. We evaluate on three task-specific datasets constructed from real-estate imagery. Task A is public; Tasks B and C are derived from proprietary listing content. Although Tasks B/C cannot be redistributed due to licensing and privacy considerations (interior photographs may contain sensitive information), we provide sufficient characterization and a fully specified split strategy so that the methodology can be reproduced on alternative datasets.

Labeling. All three tasks use manually created labels for duplicate/non-duplicate image pairs. During annotation, human annotators assigned each pair to the positive (match/near-duplicate) or negative class using task-specific definitions. For Task A, labels correspond to near-identical images that may differ by graphical overlays, borders, minor crops, resizing, or compression. For Task B, labels indicate whether two interior photos depict the same physical room across viewpoint and illumination changes. For Task C, labels indicate whether two floor plans depict the same layout, allowing for rotation, scale changes, redraw differences, and added text or graphics.

Class balance. For reproducibility and stable threshold selection, the datasets were sampled to be approximately balanced: the positive/negative pair ratio is near 1:1 in train and validation split for all tasks.

Image resolution and preprocessing. Original images vary in size (shortest side ~320 px to longest side ~1920 px). For embedding extraction, each image is decoded to RGB, resized to 224×224, and normalized with the standard ImageNet channel statistics (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]). We do not apply synthetic augmentations (e.g., random crops/flips) during decision-network training because the classifier is trained on precomputed embeddings; instead, each task's labeled pairs already contain natural variations relevant to deployment (overlays, viewpoint changes, rotation/scale differences in floor plans, etc.).

Typical variation per task:

- Task A (near-identical with graphical context changes): overlays (watermarks, borders, text), mild cropping, resizing, compression artifacts, and small color/contrast changes.
- Task B (multi-angle interiors): viewpoint changes, partial overlaps, occlusions by

furniture, clutter changes, and lighting/exposure shifts.
- Task C (floor plans): rotation and scale variation, redraw differences, compression artifacts, and added symbols/text labels.

We evaluated four approaches: (1) the proposed intermediate-layer ResNet50 embedding with a learned decision model, (2) a single-layer ResNet50 embedding using the penultimate layer (Global CNN Embedding), (3) classical feature descriptors based on SIFT, SURF, and ORB (Keypoint Descriptors), and (4) a perceptual hashing baseline using a DCT-based hash (DCT Hash).

Baseline implementation and tuning protocol. To ensure fair comparison across baselines, we used a single train/validation/test split per task (70/10/20) with group-wise separation to prevent leakage, and we performed all model selection and threshold tuning on the validation split only, reporting final metrics on the held-out test split.

Global CNN embedding (single-layer ResNet50). We extract the 2048-D feature vector from the penultimate stage of a pre-trained ResNet50 and L2-normalize it. For a pair of images, we compute similarity using cosine similarity, then select a decision threshold on the validation set that maximizes F1 for the given task.

Perceptual-hash (DCT hash). For each image we compute a DCT-based perceptual hash and compare pairs using Hamming distance. A validation-tuned Hamming threshold determines whether a pair is classified as duplicate/non-duplicate.

Keypoint-descriptor (SIFT/SURF/ORB). We implement a standard local-feature matching pipeline using classical descriptors (SIFT, SURF, ORB). For each image pair we compute a match score based on descriptor correspondences (with standard match filtering and optional geometric consistency checks where applicable). The final duplicate/non-duplicate decision is made by thresholding this match score; the threshold (and, where relevant, the specific descriptor variant) is selected on the validation split to maximize F1, and then fixed for test evaluation.

Across all baselines, the validation-only tuning protocol ensures that reported test-set numbers reflect generalization rather than threshold overfitting.

To assess the effectiveness and statistical robustness of the proposed method, we conducted evaluations across three mutually exclusive test sets, corresponding to:
- Task A: Near-identical property photos with graphical overlays.
- Task B: Multi-angle interior photographs.
- Task C: Schematic floor plan images.

For each task (A-C), we use a held-out test set and split pairs into 70% train / 10% validation / 20% test. To prevent leakage, we apply group-wise splitting using a grouping key based on (listing id, image category), ensuring that all pairs belonging to the same listing and category are assigned to the same split. This prevents near-duplicate images originating from the same listing from appearing across different splits, and also respects the structural differences between categories such as interior photographs versus floor plans. Model selection and threshold tuning are performed on the validation split only, and final performance is reported on the held-out test split.

Across tasks, datasets are mutually exclusive. Mutual exclusivity between Task A (near-identical property photos) and Task B (multi-angle interiors) was ensured through manual verification during annotation, where these image types were reviewed side-by-side to avoid overlap. For Task C (floor plans), we first separated images using an image category classification algorithm, followed by additional manual verification during annotation to ensure that no floor-plan images or pairs overlap with Tasks A/B.

For Task A (near-duplicate images), we additionally report threshold-independent ROC-AUC and PR-AUC on the held-out test split (Table 4).

Table 3: The F1-scores for each approach on the three tasks

Task \ Method	**(1)****	**(2)**	**(3)**	**(4)**
Near-duplicate images with different graphic contexts (dataset size: 10 000 image pairs)	**0.96 ± 0.01***	**0.85 ± 0.02**	**0.81 ± 0.03**	**0.70 ± 0.03**
Multi-angle room photographs (dataset size: 12 500 image pairs)	**0.87 ± 0.02***	0.78 ± 0.03	0.76 ± 0.04	0.65 ± 0.04
Schematic layouts / floor plans (dataset size: 8 800 image pairs)	**0.77 ± 0.02***	0.68 ± 0.03	0.65 ± 0.03	0.63 ± 0.03

* Bold indicates the best result for each task (higher F1 is better).
** Column meanings: (1) F1-score of the proposed multi-layer descriptor with the learned MLP decision model, (2) F1-score of a single-layer ResNet50 embedding evaluated via cosine similarity + validation-tuned threshold (no MLP), (3) F1-score of SIFT/SURF/ORB descriptors, (4) F1-score of the DCT perceptual hash.

Table 4: ROC-AUC, PR-AUC metrics on the held-out test split

Method	ROC-AUC	PR-AUC
Proposed (multi-layer + MLP)	0.982 [0.981, 0.986]	0.975 [0.974, 0.981]
Global CNN embedding	0.923 [0.919, 0.934]	0.885 [0.880, 0.907]
Keypoint descriptors	0.894 [0.887, 0.911]	0.848 [0.843, 0.874]
DCT perceptual hash	0.807 [0.801, 0.830]	0.727 [0.723, 0.761]

* Values shown as median [95% CI] over 100 bootstrap resamples of the Task A test set.

Table 3 reports the mean F1-score ± 95% CI across the three tasks. The proposed intermediate-layer embedding method consistently achieves the highest performance, confirming its robustness.

Since the central contribution of this work is multi-level aggregation of intermediate CNN stages, we conduct a structured ablation across the ResNet-50 tap points used in the proposed descriptor: C2 (layer1), C3 (layer2), and C5 (layer4). For each task (A-C), we evaluate: (i) each single stage alone, (ii) each two-stage combination, and (iii) the full three-stage descriptor. Protocol. For every ablation setting, we follow the same train/validation/test split and leakage prevention described earlier. To isolate the effect of representation depth, all ablation variants are evaluated with the same pair-fusion and the same MLP decision head (sum, absolute difference, squared difference channels, followed by the MLP), with the decision threshold tuned on validation only and metrics reported on the held-out test set.

Metrics. We report F1 for all tasks (A-C). For Task A, we additionally report threshold-independent ROC-AUC and PR-AUC for each ablation setting.

Table 5: Ablation of ResNet-50 tap points (C2/C3/C5). F1 on held-out test sets (95% CI)

Descriptor variant	Task A F1	Task B F1	Task C F1
C2 only	0.77 ± 0.03	0.68 ± 0.03	0.63 ± 0.04
C3 only	0.80 ± 0.03	0.70 ± 0.02	0.67 ± 0.03
C5 only	0.87 ± 0.02	0.80 ± 0.02	0.69 ± 0.02
C2 + C3	0.82 ± 0.02	0.73 ± 0.03	0.69 ± 0.03
C2 + C5	0.89 ± 0.02	0.83 ± 0.02	0.71 ± 0.02
C3 + C5	0.92 ± 0.02	0.85 ± 0.02	0.74 ± 0.03
C2 + C3 + C5	**0.96 ± 0.01**	**0.87 ± 0.02**	**0.77 ± 0.02**

Table 6: Threshold-independent ablation results on task A (median [95% CI] over 100 bootstrap resamples)

Descriptor variant	ROC-AUC	PR-AUC
C2 only	0.902 [0.895, 0.908]	0.841 [0.829, 0.854]
C3 only	0.925 [0.919, 0.929]	0.877 [0.866, 0.888]
C5 only	0.965 [0.962, 0.968]	0.943 [0.937, 0.949]
C2 + C3	0.937 [0.932, 0.941]	0.896 [0.886, 0.906]
C2 + C5	0.975 [0.973, 0.977]	0.959 [0.954, 0.963]
C3 + C5	0.980 [0.978, 0.981]	0.973 [0.970, 0.976]
C2 + C3 + C5	**0.982 [0.981, 0.986]**	**0.975 [0.974, 0.981]**

The ablation results show that C5 is the strongest single stage across all tasks, indicating that high-level semantic structure provides a strong baseline for duplicate detection. However, adding intermediate stages yields consistent gains: the best two-stage variant is C3+C5, and the full C2+C3+C5 descriptor achieves the best overall performance. The improvements are not purely dominated by one stage: C3 provides a large complementary boost over C5, while C2 adds a smaller but consistent additional gain

when combined with C3+C5. The threshold-independent ROC-AUC/PR-AUC trends in Table 6 mirror the F1 ordering, confirming that the triplet improves the underlying ranking quality rather than only benefiting from threshold selection.

The "Global CNN embedding" baseline in Table 3 is evaluated using direct cosine similarity between single-layer embeddings with a tuned threshold, whereas the ablation settings here use the same MLP decision head for all variants (including C5-only). Consequently, the C5-only ablation results should not be interpreted as identical to the cosine-similarity single-layer baseline; the ablation is designed to isolate representation depth under a fixed decision function.

We report runtimes on an Intel Core i7 12-gen class server, batch size = 32, images are resized to 224×224. The pipeline follows our design with a hash pre-filter to shortlist candidates and a learned MLP decision network operating on fused embedding channels. The results are shown in Table 7.

Table 7: Runtime and throughput on CPU (single-threaded). Mean ± std; 95% CI for the mean in brackets

Stage	Mean time	95% CI	Throughput	Notes
Embedding generation (ResNet-50, multi-layer), batch = 32	2.40 ± 0.06 s/batch (≈ 75 ms/img)	[2.383, 2.417] s/batch	~13.3 img/s	Three tapped stages, GAP + concat, L2-norm
Pairwise decision (fused channels + MLP)	2.7 ± 0.3 ms/pair	[2.64, 2.76] ms/pair	~370 pairs/s	Using precomputed embeddings
End-to-end query (hash pre-filter + shortlist + MLP)	141 ± 12 ms/query	[138, 143] ms/query	~7.1qps	Typical production shortlist from the hash stage

The multi-layer descriptor increases extraction cost and descriptor dimensionality; therefore, operational efficiency depends on reducing the number of expensive comparisons. We use a DCT-hash pre-filter as a first-stage candidate selector. In our production setting, the DCT stage filters out approximately 99% of candidates and produces an average shortlist size in the hundreds. This converts the matching problem from comparing a query against an extremely large inventory to evaluating only $O(10^2)$ candidates per query in the MLP stage, which is the main reason the overall pipeline is feasible at scale.

A direct "with vs. without pre-filter" end-to-end speedup measured on a small offline dataset is not representative: without pre-filtering, latency would be dominated by the number of database candidates, and a realistic comparison requires a multi-million-scale inventory. For this reason, we report shortlist size and filtered fraction as the most informative and transferable indicators of the pre-filter's operational benefit.

Overall, across all tasks, the proposed intermediate-layer embedding yields a relative improvement of 10-15% in F1-score over using the standard single-layer CNN embedding alone, and even larger gains (up to ~20 percentage points absolute) over the classical keypoint and DCT hash baselines. Within the scope of these evaluated methods, these results validate that combining features from multiple CNN layers provides a more powerful representation for detecting image similarity. Moreover, the consistency of our method's superiority across three very different scenarios (photographic near-duplicates, cross-view photos, and schematic drawings) suggests that the approach generalizes well within the considered application domain. The pipeline relies only on generic visual features and no domain-specific metadata or heuristics, indicating that it can be applied to other image collections with minimal adaptation - typically threshold retuning and, if needed, light re-training of the decision head on a small labeled set.

Scope and limitations of baseline comparison. We emphasize that our baseline set is restricted to methods that are (i) mature and widely used in practice, and (ii) compatible with the compute and engineering constraints of our production environment. In particular, we do not include more recent transformer-based copy-detection systems, learned pooling schemes such as GeM/R-MAC, NetVLAD-style retrieval architectures, or self-supervised ViT/CLIP-style embeddings. Incorporating and fairly tuning such models would require dedicated GPU services, significant engineering integration, and access to larger labeled or pseudo-labeled datasets than were available within this deployment-focused study. Consequently, we frame our empirical claims as improvements over the evaluated baselines rather than over the full current state of the art in copy or near-duplicate detection.

Finally, as a real-world validation, the pipeline has been integrated into production on a large real-estate platform operating over a large image inventory. While exact traffic and inventory figures cannot be disclosed due to confidentiality constraints, the system is used continuously to suppress redundant content during listing ingestion and moderation. In production, the DCT pre-filter is an integral component, filtering

out approximately 99% of candidates and yielding shortlist sizes in the hundreds, after which the MLP decision stage is applied to the shortlisted pairs. Operationally, the deployment reduced moderator actions by ~40% and decreased the number of duplicate listings by ~25%, supporting the real-world impact of the proposed approach.

5 Discussion

The experimental results confirm that our approach outperforms the evaluated baselines - classical keypoint descriptors, a DCT-based perceptual hash, and a single-layer ResNet50 embedding - on all three tasks. In this section, we discuss the reasons for this improved performance relative to these methods, the novelty of our method in the context of related work, and some insights on its applicability and limitations. We focus on a carefully defined comparison against baselines that are representative, interpretable, and deployable under real-world constraints.

Why intermediate-layer embedding excels: Traditional CNN-based embeddings typically use the deepest layer of the network (just before classification) to represent an image. While this captures high-level semantic information, it can omit lower-level cues needed to distinguish near-duplicates. By incorporating intermediate CNN layers, our embedding spans multiple abstraction levels: early layers capture edges, textures, and simple patterns, whereas later layers encode object parts and global scene semantics. Concatenating these representations allows the method to exploit similarity in low-level structure and high-level arrangement within a single descriptor. Unlike approaches that average intermediate responses, our method preserves layer-specific contributions through concatenation and maintains feature diversity with a modest increase in dimensionality. In the real-estate setting, this design improved performance on floor-plan graphics and cross-angle room photos. Prior work [46] outlined combining intermediate CNN features for near-duplicate detection; we extend it by (i) specifying and justifying the ResNet-50 tap points - layer1 (C2), layer2 (C3), and layer4 (C5) - supported by ablations, and (ii) formalizing a three-channel pair-fusion (sum, absolute difference, squared difference) with a fully specified MLP decision model trained with BCE. In contrast to Kordopatis-Zilos et al. [25], who aggregate intermediate features for video keyframes, we concatenate per-layer GAP descriptors for static images to preserve layer-specific cues. To the best of our knowledge, this is the first application of this design to image deduplication in real estate imagery with quantified gains over a single-layer embedding.

Comparative analysis with recent methods: Our approach fits within the broader trend of enhancing image representations for similarity tasks. The coarse-to-fine CNN matching method of Zhou et al. [24] augments a global CNN feature with local feature matching. In contrast, we represent each image with a single multi-level embedding, so matching reduces to a vector comparison and integrates naturally with indexing and scalable retrieval frameworks. Transformer-based approaches (e.g., the ViT method of Fawzy et al., 2024) learn task-specific features for copy detection and can perform strongly on benchmark challenges, but they typically require large training sets and substantial compute. Our method reuses a pre-trained CNN and trains only a small decision network, which is comparatively lightweight and data-efficient. In our experiments, training on tens of thousands of labeled pairs was sufficient because ResNet50 provides strong base features.

When compared with classical keypoint and hashing approaches, the learned embedding is more discriminative and robust. Classical methods remain useful, particularly as an efficiency layer; therefore, we integrate a hash-based pre-filter to reduce the candidate set. Similar hybrid pipelines are common in practical systems, where hashing or clustering is used for candidate shortlisting and deep models make the final decision. In our setting, the hash stage eliminated a large fraction of irrelevant images, allowing the embedding comparison stage to focus on a smaller set of candidates.

Novelty and contribution: The core novelty of our work lies in the feature representation - the enriched image embedding derived from multiple CNN layers - and in demonstrating its benefits over strong but practically oriented baselines. This is a departure from many duplicate-detection studies that either use a single global CNN feature [24, 27] or a set of local features [9, 18], but not a single unified vector encoding multi-level information. By demonstrating that such an embedding can outperform these particular baselines, we provide evidence that multi-layer feature fusion is a promising design choice for image similarity under deployment constraints. In addition, our use of a learned decision model on top of the embeddings adds a layer of adaptability; rather than fixing a similarity metric, we let the data inform how to best combine similarity measures. While learned metrics for similarity (such as Siamese networks) have been explored, our approach is lightweight in that it does not require modifying the backbone CNN or conducting metric learning during the feature extraction phase - we only learn the final decision function. A comprehensive empirical comparison against the latest transformer- and ViT-based copy-detection and retrieval models is beyond the scope of this work and remains an important direction for future research.

We also emphasize that our method proved effective in a production environment. Integration with LUN's platform showed that the approach can handle the variability of user-uploaded data, including varying resolutions and occasional label noise. The system's ability to reduce duplicate images in this setting supports its robustness.

Discussion of failure cases: No method is without limitations. Despite these improvements, several limitations were identified. First, heavy texture or style transformations (e.g., painterly filters) produced embeddings with lower cosine similarity, leading to false negatives. Second, isolated common objects across otherwise different scenes (e.g., the same chair in different rooms) sometimes caused false positives, particularly when high-level features dominated similarity. Third, floor plan rotation and non-uniform scaling, while largely handled by intermediate features, occasionally caused mismatches in borderline cases. These errors reveal specific transformation types and object-level similarities to which the embedding remains sensitive. While the decision network partially mitigates such errors by learning contextual patterns, they remain the primary source of misclassifications and represent the most promising direction for targeted improvements. These findings suggest directions for future work, such as incorporating attention mechanisms to capture global image context or refining the feature extractor through additional training on hard negatives.

Per-task error levels. Using the operating threshold τ selected on validation, the proposed system yields the following absolute error rates on the held-out test sets: Task A: FP = 2.65%, FN = 3.24%; Task B: FP = 4.39%, FN = 4.89%; Task C: FP = 9.25%, FN = 7.65%. These values contextualize the qualitative breakdown below by showing the overall frequency of false matches and missed matches per task.

Among false positives (FPs), approximately 62% arose from different rooms with nearly identical layout/furniture but only small local differences (e.g., tile color, chandelier shape, presence/absence of small items, socket/switch placement). A further 18% were due to an isolated common object dominating the match signal (e.g., the same chair or lamp across different rooms). 9% resulted from lighting/white-balance artifacts that made surfaces appear more similar than they are, and 11% from label ambiguity at the boundary of "same vs. different" rooms. Overall, ~80% of FPs are attributable to small-object/local-detail similarity, consistent with our qualitative observations.

For false negatives (FNs), the primary causes were large viewpoint changes or occlusions (~41%), substantial clutter/furniture rearrangements (~25%), strong lighting/exposure shifts (~19%), and low resolution/blur (~15%). Notably, rooms without renovation - with repetitive textures and minimal distinctive cues - were over-represented in both FP and FN groups.

Other difficult cases we observed include mirror reflections (duplicating objects and confusing spatial cues), heavy watermark overlays (obscuring small discriminative details), and repeated developer-standard layouts across different apartments that produce near-identical geometry despite being distinct spaces.

Ablation findings. The ablation study in "Experiments and Results" (Tables 5-6) quantifies the contribution of each tapped stage and their combinations. Overall, the results show that multi-stage aggregation consistently improves performance over single-stage descriptors, and that the selected {C2,C3,C5} triplet offers the best average trade-off across the three tasks. The threshold-independent ROC-AUC/PR-AUC results on Task A further confirm that the improvement is not driven solely by threshold selection but reflects a stronger underlying ranking of matching vs. non-matching pairs.

In conclusion, the proposed intermediate-layer embedding method successfully advances the performance in image deduplication tasks, thanks to its comprehensive feature representation and learned comparison strategy. The method stands out for its combination of high accuracy, generality across tasks, and practical deployability.

6 Conclusion

This study addressed the problem of near-duplicate image detection and introduced an image descriptor generation method based on intermediate-layer CNN features that improves image similarity modeling over widely used baselines. We evaluated the method on three real estate image deduplication tasks and demonstrated that existing techniques in our comparison - ResNet50 global embeddings, classical keypoint descriptors, and DCT perceptual hashing - often exhibit limited accuracy in challenging settings. To address this limitation, we proposed an extended vector representation constructed by combining intermediate-layer outputs from a pre-trained ResNet50. We detailed a data preparation method and a step-by-step algorithm for building the model, which includes extracting and normalizing the multi-layer embeddings, storing them efficiently, and applying a learned decision neural network for comparing image pairs.

The use of intermediate CNN layers to construct the image embedding represents the key innovation and primary contribution of this work. This enriched embedding captures a more comprehensive set of image features than traditional single-layer embeddings, enabling superior discrimination between similar and dissimilar image pairs. The proposed method outperformed the evaluated baselines (ResNet50 global embedding, keypoint descriptors, and DCT hash) across all tasks on our datasets, with particularly strong gains in challenging scenarios such as cross-angle photo matching. The proposed model improved upon existing solutions, yielding more accurate and confident detection of near-duplicate images. In quantitative terms, it achieved F1-scores of 0.96, 0.87, and 0.77 on the respective tasks, outperforming baseline methods by a wide margin in each case. These results were validated on real-world data, and the model has been successfully deployed in

a production environment, demonstrating its practical applicability.

These findings suggest several promising directions for future research. One direction is to explore alternative backbone architectures or hybrid layer combinations (e.g., integrating ResNet50 and Vision Transformer features) to enhance descriptor performance. Another area is optimizing the embedding size - using dimensionality reduction techniques like PCA or autoencoders on the concatenated vector could retain the performance while reducing storage and computation needs.

Additionally, while our method was evaluated on distinct tasks separately, an interesting extension would be a unified deduplication system that can handle multiple types of near-duplicate criteria simultaneously (context duplicates, cross-angle, etc.) possibly by multi-task learning.

Future work could explore adaptive weighting of intermediate layers, allowing the contribution of each layer to be dynamically adjusted based on the image domain or transformation type.

In summary, this work presents an effective approach to image similarity modeling and deduplication by leveraging intermediate CNN features to construct a multi-level embedding. The approach aligns with industrial practice by combining deep representations with efficient candidate filtering. The method is applicable to domains that require detecting duplicate or near-duplicate images and provides a clear baseline for future work on improved representations and retrieval pipelines.

References

[1] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... & Sutskever, I. Zero-shot text-to-image generation. In International conference on machine learning 2021, pp. 8821-8831.

[2] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., & Chen, M. Hierarchical text-conditional image generation with clip latents 2022, 1(2), 3. https://doi.org/10.48550/arXiv.2204.06125

[3] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition 2009, pp. 248-255. https://doi.org/10.1109/CVPR.2009.5206848

[4] Thyagharajan, K. K., & Kalaiarasi, G. A review on near-duplicate detection of images using computer vision techniques. Archives of Computational Methods in Engineering 2021, 28(3), pp. 897-916. https://doi.org/10.1007/s11831-020-09422-6

[5] Kaur, G., & Devgan, M. S. Data deduplication methods: a review. International Journal of Information Technology and Computer Science 2017, 10, pp. 29-36. https://doi.org/10.5815/ijitcs.2017.10.03

[6] Islam, S. M., & Debnath, R. A comparative evaluation of feature extraction and similarity measurement methods for content-based image retrieval. International Journal of Image, Graphics and Signal Processing 2020, 10(6), 19. https://doi.org/10.5815/ijigsp.2020.06.03

[7] Bajaj, E. N., Gill, E. J. S., & Kumar, R. An approach for similarity matching and comparison in content-based image retrieval system. IJ Inf. Eng. Electron. Bus. 2015, pp. 48-54. https://doi.org/10.5815/ijieeb.2015.05.07

[8] Lowe, D. G. Distinctive image features from scale-invariant keypoints. International journal of computer vision 2004, 60, pp. 91-110. https://doi.org/10.1023/B:VISI.0000029664.99615.94

[9] Chum, O., Philbin, J., Isard, M., & Zisserman, A. Scalable near identical image and shot detection. In Proceedings of the 6th ACM international conference on Image and video retrieval 2007, pp. 549-556. https://doi.org/10.1145/1282280.1282359

[10] Bay, H., Tuytelaars, T., & Van Gool, L. Surf: Speeded up robust features. In Computer Vision-ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9 2006, pp. 404-417. https://doi.org/10.1007/11744023_32

[11] Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. ORB: An efficient alternative to SIFT or SURF. In 2011 International conference on computer vision 2011, pp. 2564-2571. https://doi.org/10.1109/ICCV.2011.6126544

[12] Lei, Y., Zheng, L., & Huang, J. Geometric invariant features in the Radon transform domain for near-duplicate image detection. Pattern recognition 2014, 47(11), 3630-3640. https://doi.org/10.1016/j.patcog.2014.05.009

[13] Tang, Z., Yang, F., Huang, L., & Zhang, X. Robust image hashing with dominant DCT coefficients. Optik 2014, 125(18), pp. 5102-5107. https://doi.org/10.1016/j.ijleo.2014.04.079

[14] Jie, Z. A novel block-DCT and PCA based image perceptual hashing algorithm. arXiv preprint arXiv:1306.4079 2013. https://doi.org/10.48550/arXiv.1306.4079

[15] Zeng, J. A Novel Block-DCT and PCA Based Image Perceptual Hashing Algorithm. arXiv preprint arXiv:1306.4079 2013. https://doi.org/10.48550/arXiv.1306.4079

[16] Ke, Y., Sukthankar, R., & Huston, L. An efficient parts-based near-duplicate and sub-image retrieval system. In Proceedings of the 12th annual ACM international conference on Multimedia 2004, pp. 869-876. https://doi.org/10.1145/1027527.1027729

[17] Nian, F., Li, T., Wu, X., Gao, Q., & Li, F. Efficient near-duplicate image detection with a

local-based binary representation. Multimedia Tools and Applications 2016, 75, 2435-2452. https://doi.org/10.1007/s11042-015-2472-1

[18] Zhao, W. L., Ngo, C. W., Tan, H. K., & Wu, X. Near-duplicate keyframe identification with interest point matching and pattern learning. IEEE Transactions on Multimedia 2007, 9(5), 1037-1048. https://doi.org/10.1109/TMM.2007.898928

[19] Zhao, W. L., & Ngo, C. W. Scale-rotation invariant pattern entropy for keypoint-based near-duplicate detection. IEEE Transactions on Image Processing 2009, 18(2), 412-423. https://doi.org/10.1109/TIP.2008.2008900

[20] Li, Y. A fast algorithm for near-duplicate image detection. In 2021 IEEE International Conference on Artificial Intelligence and Industrial Design (AIID) 2021, pp. 360-363).

[21] Chum, O., Philbin, J., & Zisserman, A. Near duplicate image detection: Min-hash and TF-IDF weighting. In Bmvc 2008, 810, pp. 812-815. https://doi.org/10.5244/C.22.50

[22] Liu, L., Lu, Y., & Suen, C. Y. Variable-length signature for near-duplicate image matching. IEEE Transactions on Image Processing 2015, 24(4), pp. 1282-1296. https://doi.org/10.1109/TIP.2015.2396208

[23] Xie, L., Tian, Q., Zhou, W., & Zhang, B. Fast and accurate near-duplicate image search with affinity propagation on the ImageWeb. Computer Vision and Image Understanding 2014, 124, 31-41. https://doi.org/10.1016/j.cviu.2013.12.011

[24] Zhou, Z., Lin, K., Cao, Y., Yang, C. N., & Liu, Y. Near-duplicate image detection system using coarse-to-fine matching scheme based on global and local CNN features. Mathematics 2020, 8(4), 644. https://doi.org/10.3390/math8040644

[25] Kordopatis-Zilos, G., Papadopoulos, S., Patras, I., & Kompatsiaris, Y. Near-duplicate video retrieval by aggregating intermediate cnn layers. In MultiMedia Modeling: 23rd International Conference, MMM 2017, Reykjavik, Iceland, January 4-6, 2017, Proceedings, Part I 23 2017, pp. 251-263. https://doi.org/10.1007/978-3-319-51811-4_21

[26] Zhang, Y., Zhang, S., Li, Y., & Zhang, Y. Single-and cross-modality near duplicate image pairs detection via spatial transformer comparing CNN. Sensors 2021, 21(1), 255. https://doi.org/10.3390/s21010255

[27] Barz, B., & Denzler, J. Do we train on test data? purging cifar of near-duplicates. Journal of Imaging 2020, 6(6), 41. https://doi.org/10.3390/jimaging6060041

[28] Matatov, H., Naaman, M., & Amir, O. Dataset and case studies for visual near-duplicates detection in the context of social media 2022. https://doi.org/10.48550/arXiv.2203.07167

[29] Tralic, D., Zupancic, I., Grgic, S., & Grgic, M. CoMoFoD - New database for copy-move forgery detection. In Proceedings ELMAR-2013 2013, pp. 49-54.

[30] Morra, L., & Lamberti, F. Benchmarking unsupervised near-duplicate image detection. Expert Systems with Applications 2019, 135, pp. 313-326. https://doi.org/10.1016/j.eswa.2019.05.002

[31] Barz, B., & Denzler, J. Hierarchy-based image embeddings for semantic image retrieval. In 2019 IEEE winter conference on applications of computer vision (WACV) 2019, pp. 638-647. https://doi.org/10.1109/WACV.2019.00073

[32] Berman, M., Jégou, H., Vedaldi, A., Kokkinos, I., & Douze, M. Multigrain: a unified image embedding for classes and instances. arXiv preprint arXiv:1902.05509 2019. https://doi.org/10.48550/arXiv.1902.05509

[33] Yu, Z., Zheng, J., Lian, D., Zhou, Z., & Gao, S. Single-image piece-wise planar 3d reconstruction via associative embedding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, pp. 1029-1037. https://doi.org/10.1109/CVPR.2019.00112

[34] Rau, A., Garcia-Hernando, G., Stoyanov, D., Brostow, G. J., & Turmukhambetov, D. Predicting visual overlap of images through interpretable non-metric box embeddings. In Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part V 16 2020, pp. 629-646. https://doi.org/10.1007/978-3-030-58558-7_37

[35] Feng, G., Hu, Z., Zhang, L., & Lu, H. Encoder fusion network with co-attention embedding for referring image segmentation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition 2021, pp. 15506-15515. https://doi.org/10.1109/CVPR46437.2021.01525

[36] Asadi-Aghbolaghi, M., Azad, R., Fathy, M., & Escalera, S. Multi-level context gating of embedded collective knowledge for medical image segmentation 2020. https://doi.org/10.48550/arXiv.2003.05056

[37] He, K., Zhang, X., Ren, S., & Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition 2016, pp. 770-778. https://doi.org/10.1109/CVPR.2016.90

[38] Lytvynenko, T. I., Panchenko, T. V., & Redko, V. D. Sales forecasting using data mining methods. Вісник Київського національного університету імені Тараса Шевченка. Серія: Фізико-математичні науки 2015, 4, 148-155.

[39] Bieda, I., & Panchenko, T. A systematic mapping study on artificial intelligence tools used in video editing. International Journal of Computer Science & Network Security, 2022. 22(3), 312-318.

https://doi.org/10.22937/IJCSNS.2022.22.3.40
[40] Bieda, I., Kisil, A., & Panchenko, T. An approach to scene change detection. In 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) 2021, 1, pp. 489-493. https://doi.org/10.1109/IDAACS53288.2021.9660887
[41] Panchenko, T., & Bieda, I. A Comparison of scene change localization methods over the open video scene detection dataset. International Journal of Computer Science & Network Security 2022, 22(6), 1-6.
[42] Kubytskyi, V., & Panchenko, T. An Effective Approach to Image Embeddings for E-Commerce. In IT&I 2022, pp. 341-349.
[43] Fawzy, M., Tawfik, N. S., & Saleh, S. N. Enhancing Image Copy Detection through Dynamic Augmentation and Efficient Sampling with Minimal Data. Electronics 2024, 13(16), 3125. https://doi.org/10.3390/electronics13163125
[44] Chandrasiri, M. D. N., & Talagala, P. D. Cross-ViT: Cross-attention Vision Transformer for Image Duplicate Detection. In 2023 8th International Conference on Information Technology Research (ICITR) 2023, pp. 1-6. https://doi.org/10.1109/ICITR61062.2023.10382916
[45] Qin, Y., Ye, O., & Fu, Y. An automatic near-duplicate video data cleaning method based on a consistent feature hash ring. Electronics 2024, 13(8), 1522. https://doi.org/10.3390/electronics13081522
[46] Kubytskyi, V.; Panchenko, T. Enriched Image Embeddings as a Combined Outputs from Different Layers of CNN for Various Image Similarity Problems More Precise Solution. In Advances in Artificial Systems for Logistics Engineering III; Hu, Z., et al. (Eds.); Lecture Notes in Data Engineering and Communications Technologies; Springer: Cham, Switzerland, 2023; 180, pp. 1-13. https://doi.org/10.1007/978-3-031-36115-9_30
[47] Image Similarity Dataset: 10000 labelled image pairs. Kaggle 2025. Available online: https://www.kaggle.com/datasets/pantaras/near-duplicate-images/data