IDedupNet: A MobileNetV3-Based Deep Learning Framework for Efficient Image Deduplication in Cloud Computing Environments

Mohd Hasan Mohiuddin[1], Latha Tamilselvan[2*]
[1]Department of Computer Science and Engineering, B.S. Abdur Rahman Crescent Institute of Science &Technology, Vandalur, Chennai – 600048, India.
[2]Department of Information & Technology, B.S. Abdur Rahman Crescent Institute of Science &Technology, Vandalur, Chennai – 600048, India.
E-mail: mohiddin.hasan@outlook.com, latha.tamil@crescent.education
*Corresponding author

Keywords: image Deduplication, AI, DL, cloud computing, infrastructure efficiency

Received: November 6, 2024

Image deduplication is becoming increasingly important for cloud storage infrastructures to handle the increasing amount of multimedia material. Through increased storage efficiency, effective picture deduplication may optimize resources and save expenses. It also improves performance by facilitating quicker access, utilizing less bandwidth, and enhancing data integrity. Although heuristic-based classical deduplication techniques work well in various storage infrastructures, they cannot keep up with the dynamic nature of cloud storage resources. This study presents IDedupNet, a revolutionary DL-based framework that improves infrastructure performance in cloud computing by efficiently detecting duplicate and near-duplicate photos. Our approach leverages MobileNetV3 for feature extraction and CNN-based encodings for image deduplication, enabling it to manage duplicate photos in highly dynamic contexts efficiently. Additionally, we provide a Learning-Based Image Deduplication (LBID) approach that improves deduplication performance by extending the use of the IDedupNet model. Experimental evaluation demonstrates a high accuracy of 98.68% on benchmark datasets, consistently outperforming existing models. The underlying technique and deep learning framework may be easily integrated into real-time cloud storage systems to increase customer satisfaction and infrastructure efficiency.

Povzetek: Predstavljena je arhitektura globokega učenja IDedupNet, ki temelji na MobileNetV3, za učinkovito deduplikacijo slik v oblaku. Z uporabo MobileNetV3 za ekstrakcijo značilnosti in CNN kodiranjem, IDedupNet učinkovito zaznava duplikate in skoraj duplikate slik.

1 Introduction

Businesses worldwide have benefited from readily available storage options since the advent of cloud computing and its ecosystem. Businesses may now handle and securely retain their data for later use. To obtain business insights, they can also do data analytics. The cloud stores much audiovisual content, which might lead to duplicate information. Duplicating documents, images, or videos can lead to incorrect data, lost storage space, wasteful computer use, and increased time consumption. To address this issue, deduplication algorithms were developed, identifying and removing duplicate components to provide access to unique objects [1]. Finding and removing duplicates may significantly increase the efficiency of cloud data centers when managing enormous volumes of data—which in cloud storage architecture might exceed petabyte proportions. Deduplication methods reduce energy usage, storage needs, and computational inefficiencies while increasing efficiency for cloud data centers by employing delay [2]. As a result of artificial intelligence and other technologies, learning-based approaches have replaced heuristics. Intelligent application services that use recurrent learning ideas can automatically identify duplicate items in storage infrastructure, improving performance. 3. Better safe deduplication strategies that employ hybrid cloud storage systems to remove and preserve duplicate things originated from the increased need to identify duplicate items in cloud storage systems [4].

To find duplicate items in the literature, researchers have examined unsupervised methods. Many industries, like the healthcare sector, where medical pictures are kept on cloud infrastructure, have found deduplication approaches crucial. Duplicate components in medical images have been automatically detected through the development of algorithms like fusion learning [5]. The idea of automatically labeling images has also been researched in the literature for computer vision applications that store a lot of data in cloud computing infrastructures to improve computational and storage efficiency [6]. Research on deduplication is also underway for Internet of Things applications, which produce massive amounts of data, including data from cloud-based real-world apps [7], [8]. Nonetheless, the research indicates that DL models must be improved to create deep learning-based deduplication frameworks for cloud-based video material.

This paper introduces IDedupNet, a state-of-the-art deep learning-based system that enhances cloud computing infrastructure performance by efficiently detecting duplicate and near-duplicate photos. Our approach efficiently manages duplicate photos in highly dynamic scenarios using deep learning for picture encoding and deduplication. We also provide a Learning-Based Image Deduplication (LBID) approach that leverages the IDedupNet model to improve deduplication performance. Our proposed deep learning model attains a high accuracy of 98.68% on benchmark datasets, boosting trust in its performance and consistently outperforming other models. Therefore, the underlying algorithm and this unique deep learning architecture may be readily included in real-time cloud storage systems, improving infrastructure effectiveness and client satisfaction.

The remainder of the document is structured as follows: Previous studies on the different techniques of picture deduplication applications created with learning-based methodologies are reviewed in Section 2. Section 3 provides research design details. The DL-based approach for efficient picture deduplication in cloud computing systems is presented in Section 4. The results of our empirical analysis utilizing a benchmark dataset are presented in Section 5, with incisive criticism of the proposed model and a comparison with state-of-the-art models. A comparison between the suggested method and hashing-based state-of-the-art techniques is given in Section 6. Section 7 discusses the proposed research and its significance. Section 8 concludes our study and suggests the following lines of inquiry.

2 Related work

Various approaches for deduplicating multimedia objects in cloud environments have been discussed in the literature. Godavari et al. [1] emphasized the importance of efficiently finding and eliminating duplicate data to optimize primary storage for deduplication. However, workloads in the cloud pose a challenge. Cloud data, typically accessed infrequently, challenges cache efficiency in deduplication systems due to a lack of temporal locality. Zhao et al. [2] exacerbated storage issues by extensively using Docker containers. DupHunter recommends effective deduplication. Usharani and Danalakshmi [3] improved detection and storage efficiency by evaluating and correlating pixel dimensions, reducing picture repetition in innovative application services.

Mageshkumar et al. [4] proposed an efficient paradigm incorporating block-level deduplication, Diffie-Hellman encryption, and experimentation. Convergent encryption enhances the security of cloud data deduplication. Ahmed et al. [5] employed a global data aggregation technique to improve the accuracy and precision of CAD system performance with duplicate medical images. Xu et al. [6] introduced reinforcement learning-based indexing for deduplication, addressing disk bottlenecks, and enhancing memory efficiency. Prathima et al. [7] provided on-demand resources to support IoT data processing. Storage and performance are optimized through effective data deduplication in distributed caching. Pragash and Jayabarathy [8] reduced computational complexity through data deduplication, examining various methods for efficiency to aid researchers in developing workable ideas.

Zheng et al. [11] emphasized that cloud data deduplication lowers redundancy by maintaining unique copies, which is challenging due to the requirement for strong encryption and detection of duplicate files. Fu et al. [12] enhanced efficiency and security by offering a fog-to-multi-cloud secured storage solution with application-aware deduplication for sensitive medical data. Wang et al. [13] introduced an effective user revocation method for secure deduplication, reducing update computation and communication costs. Zhang et al. [14] utilized blockchain technology to minimize computational costs and guarantee data security and integrity. Xu et al. [15] presented LIPA, a learning-based deduplication technique that addresses disk bottleneck problems using reinforcement learning with little memory overhead for effective deduplication.

Jai et al. [16] suggested a content-based strategy using a triplet loss deep learning network and scalable hashing, showing significant progress compared to existing approaches that rely on URLs. Zhou et al. [17] addressed issues with copyright and privacy arising from the growth of digital multimedia online in cloud and large data environments. Rajput et al. [18] offered a secure approach for human activity recognition using picture obfuscation in cloud-based expert systems, addressing data privacy concerns. Anuradha et al. [19] utilized emerging technologies such as IoT, CC, and AI for cancer prediction and encryption for safe cloud data storage and accessibility. Kumar et al. [20] emphasized using data compression and deduplication methods, notably the SHA-3 algorithm, to optimize cloud computing capacity for safe deduplication.

Asif et al. [21] suggest automated processing is required for disaster management using social media photography. They propose a strategy driven by taxonomy, deep learning, and decision-making methodologies to enhance real-time emergency response and crisis management. Takeshita et al. [22] address privacy issues and provide security against hostile attackers by introducing a single-server protocol for safe cross-user nearly-identical deduplication in cloud storage. Vijayalakshmi and Jayalakshmi [23] focus on effective deduplication techniques to manage data redundancy concerns and highlight the importance of CC in managing the exponential rise of digital data. Shetty et al. [24] highlight the need for incident management due to the shift to cloud computing. They utilize a multi-task BiLSTM-CRF model for named entity recognition, SoftNER, an unsupervised knowledge extraction framework, which achieves excellent accuracy. Zhang et al. [25] present CEVAS, a cutting-edge serverless collaborative video analytics solution on the cloud. It shows notable advantages over current systems by achieving cost-effectiveness, maintaining high throughput, and optimizing resource management.

Table 1: Comparative summary of related works

Study	**Key Contributions**	**Methodology**	**Results**	**Limitations**
Godavari et al. [1]	Hybrid deduplication system with content-based cache for cloud environments	Heuristic-based deduplication with cache optimization	Improved storage efficiency	Limited scalability for dynamic cloud data
Zhao et al. [2]	High-performance deduplication for Docker registries	End-to-end deduplication scheme for containerized environments	Enhanced deduplication speed	Not adaptable to diverse multimedia datasets
Usharani & Danalakshmi [3]	Recurrent learning-based deduplication for innovative applications	Recurrent learning algorithms	Higher accuracy for specific use cases	Ineffective for large-scale dynamic datasets
Mageshkumar et al. [4]	Secure deduplication using cryptographic techniques in hybrid cloud	Diffie-Hellman encryption and block-level deduplication	Improved security and deduplication	High computational overhead
Ahmed et al. [5]	Unsupervised fusion learning for medical image deduplication	Fusion learning algorithms	Increased precision in medical imaging	Domain-specific; lacks generalizability
Fu et al. [12]	Fog-to-multi-cloud secure deduplication for eHealth data	Application-aware deduplication integrated with security protocols	Enhanced security and efficiency	Inefficient for non-medical multimedia datasets
MobileNet V3 (Proposed)	Efficient and robust deduplication for dynamic cloud environments	MobileNet V3 for feature extraction, CNN-based encodings, and cosine similarity for duplicate detection	Accuracy: **98.68%**, F1-score: **95.6%**	Refer to Section 5.1 for limitations

Lu et al. [26] propose a deduplication technique that allows 7X faster image updates without loss of efficiency. They address the issue of data duplication when updating Docker images. Zhang et al. [27] enhance accurate sentiment categorization through an effective annotation technique using artificial and emotional lexicons in e-commerce remarks. Li et al. [28] present an edge-assisted approach that minimizes resource strain on terminal devices while maintaining privacy in image processing, addressing privacy concerns with the rise of the IoT and sensitive picture data. Xing et al. [29] describe a method for leveraging street photos from driving car recorders to update traffic laws, achieving excellent accuracy in rule clustering by utilizing spatiotemporal attention, object detection, and model compression. Hamandawana et al. [30] present Redup, a caching solution that addresses issues with deduplication and speed in ML/DL storage clusters. It outperforms other systems in reducing deduplication overhead thanks to its dual-level caching architecture.

Wang et al. [31] suggest a deep learning-based emotional big data facial expression detection system for autistic sufferers. Boutros et al. [32] discuss challenges to integrity faced by FPGAs in data centers and how DL models avoid timing issues caused by integrity assaults. Jia et al. [33] propose a deep learning-based content-based video deduplication approach to ease storage and bandwidth constraints. Jansen et al. [34] utilize Docker to integrate data, software, and runtime environment, ensuring clinical Deep Learning research repeatability with the Curious Containers architecture. Du et al. [35] highlight how AI facilitates the finding and prioritization of evidence, addressing backlogs in digital forensics due to increased cases and data in law enforcement. Chen [36] presented a method for cleaning large amounts of data using GANs and repeated change detection that prioritizes cleaning affordable decision trees.

Abuhasel et al. [37] linked networks due to IIoT, necessitating strong security measures because of potential attacks. Sophisticated methods such as SoftMax-DNN improve efficiency and security. Chaudhary et al. [38] improved cybersecurity, creating new avenues for attack. Machine learning efficiently identifies threats, with many models attaining high accuracy. Varied material is ubiquitous with mobile multimedia, which is vital in the healthcare industry. Gupta et al. [39] proposed deep learning-based content hashing for image deduplication, improving accuracy and optimizing cloud storage performance. Tahir et al. [40], although security problems are still present, cloud computing provides customizable services over the internet. Using evolutionary algorithms, a novel CryptoGA model outperforms conventional cryptography techniques regarding data integrity and privacy. Table 1 provides a summary of the findings of the literature. The literature reveals a need to enhance DL models to develop DL-based deduplication frameworks for multimedia objects in cloud environments.

3 Research design

This work focuses on three main research questions: how does the IDedupNet model compare to hashing-based techniques at (1) Deduplication efficiency, which is key in accuracy and robustness to transformations and scalability? 2) How does the MobileNetV3 architecture influence cloud systems' computational efficiency and deduplication accuracy? and (3) What is the role of transformation pipeline and feature encoding in improving accuracy, scalability, and real-time processing of the deduplication process in dynamic cloud environments? A four-pronged framework is proposed to meet these goals: (1) A transformation pipeline to format the image into a standard format and preprocess these images using transformations such as crop, resize, normalize, and augment to become robust against resolution and format variations. It increases system scalability and reduces false negatives for near-duplicate images. (2) The feature extraction process is performed in high-dimensional semantic space (shapes, edges, and textures); this stage is done by MobileNetV3, keeping a small footprint even with high recall. (3) Feature encoding compresses extracted features into compact representations, helping reduce computational overheads and allowing similarity comparisons to be made quickly. (4) A similarity measure, cosine similarity, to detect duplicates pretty accurately, including transformations such as cropping or color adjustment. Together, these components lead to measurable goals: high deduplication accuracy (98%+), scalability (able to process massive datasets), and robustness (few false positives and negatives across transformations). Evaluation metrics like precision, recall, and F1-score assess the framework objectively. The detailed design establishes the originality and efficacy of IDedupNet in filling the limitation gap in the state-of-the-art of these deduplication methods.

4 Proposed framework

We proposed a deduplication architecture called IDedupNet, as illustrated in Figure 1, to solve the essential issue of image deduplication in cloud computing environments—where vast volumes of picture data are stored and handled. An illustration of a deep learning method is this framework. A common source of storage inefficiencies in cloud systems is duplicate or nearly duplicate photographs, which slow down storage systems and increase costs. IDedupNet uses neural networks to overcome these issues by efficiently identifying and removing duplicate pictures. The crucial image processing elements are pre-processing, image conversion, and the transformation pipeline. Cloud-set photographs come in various formats, resolutions, and color schemes. By converting images to a standardized format, the conversion procedure guarantees consistency for further processing. Components of a transformation pipeline include applying transformations such as cropping, resizing, normalization, and even augmentation. Importantly, these changes help prepare images for efficient processing and may improve the model's ability to handle a range of image variations, particularly in deduplication applications. If image formats are standardized and adjustments are made, the system can detect copies more quickly. The deduplication accuracy is increased when pre-processing ensures that the main picture attributes remain evident even when two photographs undergo significant changes (due to different resolutions or minor adjustments). Feature extraction is essential to the proposed system. In resource-constrained environments, such as cloud computing systems or mobile devices, the lightweight CNN architecture known as MobileNetV3 is intended to function well. Here, it extracts significant features from images that include the relevant information, such as shapes, edges, textures, and other properties. The model builds a feature vector that captures every unique element of an input image. MobileNetV3 generates feature vectors, either in batch mode or individually, for every image in a collection of photos. It is necessary to extract strong traits to find duplication. Due to its computational efficiency and ability to expedite the processing of extensive picture collections, MobileNetV3 is especially well-suited for cloud computing settings. In the case of near-duplicate pictures, pixel-wise comparison is computationally expensive and prone to errors. However, the deduplication method is guaranteed to be able to compare images based on their content thanks to feature extraction.

The features that were extracted from MobileNetV3 are encoded using a CNN-based design. As the feature vectors become more straightforward, the encoding process maintains essential information about the image's content. The main goal of the single-image encoding procedure is to provide a picture in a reduced manner. Batch encoding is utilized for many photos, and contrastive learning techniques may be applied to enhance image pair comparisons. Encoding minimizes the amount of processing and storage required when dealing with large datasets. The deduplication method employs the smaller, more portable pictures as a comparison instead of the larger ones. This technique allows cloud systems to handle large repositories more quickly and scalably.

The encoded feature vectors are compared using a similarity measure to find duplicates. At this point, the actual deduplication occurs when two images are identified as duplicates if their similarity scores exceed a predetermined threshold. Finding duplication in a picture is done by comparing its encoded feature vector with other cloud-stored image representations. Multiple picture batch comparisons are performed to identify duplicates in the collection or with previously uploaded images in the database. Solutions for cloud storage are made to eliminate unnecessary data and keep only the original photos. The system can accurately identify and flag duplicates by comparing their attributes, even when the features have been somewhat modified (e.g., cropped, scaled, or color-adjusted). This capability is crucial since pixel-by-pixel comparisons could not reliably detect duplicates in cloud environments where image manipulations are frequent. The technique compares the photographs' similarity to get duplicate detection results. To achieve these outcomes, redundant photos may need to be removed from the cloud storage, combined into a single entry, or identified as such. Cloud systems must contend with duplicate images, which use more storage capacity. The framework finds and removes duplicates, which reduces operational costs, maximizes the efficiency of data retrieval, and reduces the amount of space required for storage. Regularly processing and deduplicating millions of images is very helpful in large-scale settings like cloud storage systems (like AWS and Google Cloud).

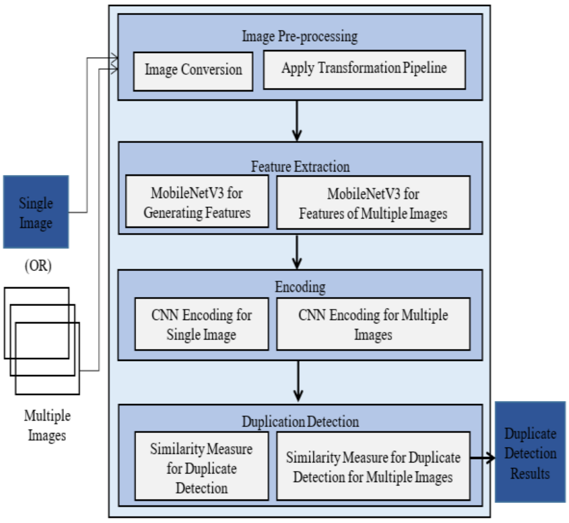


Image Pre-processing
Image Conversion
Apply Transformation Pipeline
Feature Extraction
MobileNetV3 for Generating Features
MobileNetV3 for Features of Multiple Images
Single Image
(OR)
Multiple Images
Encoding
CNN Encoding for Single Image
CNN Encoding for Multiple Images
Duplication Detection
Similarity Measure for Duplicate Detection
Similarity Measure for Duplicate Detection for Multiple Images
Duplicate Detection Results

Figure 1: Proposed deep learning framework, IDedupNet, for efficient image deduplication in cloud computing environmentsIDedupNet is designed efficiently, utilizing MobileNetV3 and CNN-based encodings to ensure the framework can handle the massive volumes of data typically seen in cloud systems.

Cloud environments often use distributed architectures for faster processing. IDedupNet can be integrated with parallel computing frameworks, enabling multiple nodes to process batches of images concurrently. The framework might support real-time deduplication as photos are uploaded to the cloud. The total strain on storage systems is decreased since duplicate data is automatically identified and handled. Frees up space in cloud storage systems by removing unnecessary photos. In pay-as-you-go systems, lower storage utilization translates into lower cloud storage service prices. Error-free picture retrieval is achieved by eliminating duplicates, which decreases the number of irrelevant photos returned by queries. Due to reduced data processing requirements, deduplication helps cloud data centers become more environmentally sustainable using less energy. IDedupNet, which focuses on lowering redundancy and enhancing storage management using intelligent deduplication algorithms, is a practical, scalable, and cloud-optimized solution for managing giant picture collections.

The goal of MobileNetV3 is to carry out a range of visual tasks, such as photo identification and classification, quickly and effectively. MobileNetV3 is mainly used for picture deduplication in creating image feature embeddings or encodings. Embeddings are high-dimensional visuals that emphasize the salient features of the pictures. Depthwise separable convolutions are combined with linear bottlenecks in MobileNetV3 to achieve a compromise between computational economy and accuracy. One kind of convolution splits the process into two stages: pointwise convolution and depthwise convolution (1x1 convolution). This form of convolution is called depthwise separable convolution. Linear bottlenecks enable an effective way to incorporate non-linearity computationally after dimensionality reduction. For an input tensor X of shape (H, W, C) (height, width, channels), the depthwise convolution filter K takes the form (k_h, k_ω, C), where $k_h$and k_ω The depthwise convolution is calculated using the kernel's height and width, as shown in Eq. 1.

$$X' = X * K \qquad (1)$$

where * denotes the convolution process, producing the output tensor X' in the form (H', W', C). After applying it, it aggregates the depthwise convolution's outputs using a 1x1 kernel, as in Eq. 2.

$$X'' = X' * K' \qquad (2)$$

where the output tensor is denoted by X" and the 1x1 convolution kernel is represented by K'. Global average pooling, which comes after the feature extraction layers, reduces the feature maps' spatial dimensions to a single vector for each image. To do this, average the values over all spatial locations as expressed in Eq. 3.

$$f_i = \frac{1}{H\times W}\sum_{h-1}^{H}\sum_{\omega-1}^{Ww} x_{i,h,\omega} \qquad (3)$$

where the pixel value at position (h, ω) in the i-th feature map is represented by $x_{i,h,\omega}$ and f_i is the i-th component of the feature vector. A feature vector, or picture embedding, is the result of the last layer of MobileNetV3, which is frequently performed after global average pooling. This vector captures the essential features of the image in a high-dimensional space. The CNN creates an embedding vector. E_I with dimensions D for an image, I as expressed in Eq. 4.

$$E_I = \text{f(I)} \qquad (4)$$

for which the CNN model is represented by f. We use cosine similarity to calculate how similar two pictures' embeddings are to identify duplication. The following formula may be used to determine the cosine similarity between two embeddings, $E_1\ and E_2$ as in Eq. 5.

$$\text{cosine similarity } (E_1, E_2) = \frac{E_1 \,.\, E_2}{||E_1||\,||E_2||} \qquad (5)$$

The dot product is represented by., and the vector's Euclidean norm, or magnitude, is shown by ||. ||. Duplicates are found using similarity criteria θ embeddings are more significant than or equal to θthe. Should two photos' cosine similarity be regarded as duplicates that images are duplicated if cosine similarity $(E_1, E_2) \geq \theta$. Utilizing

MobileNetV3, determine the feature vector for every image in the collection. To find the similarity between any two embeddings, compute their cosine similarity. Assess if two picture pairings are duplicates by comparing their similarity scores to the threshold. Batch generation of embeddings is common practice to increase efficiency due to the possibly high number of pictures. To expedite the process, parallelization of similarity computations is possible, particularly for big datasets. MobileNetV3-based image deduplication entails employing a CNN to extract high-dimensional feature embeddings, calculating the cosine similarity between these embeddings, and applying a similarity threshold to detect duplicates. Unlike other hashing techniques, this method uses CNN's capacity to collect semantic content, enabling more versatile and efficient duplication identification.

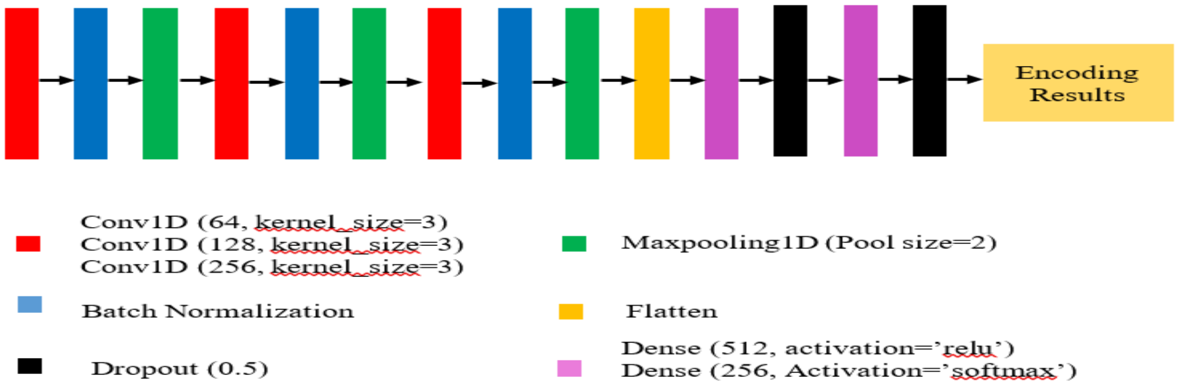

Encoding Results
🟥 Conv1D (64, kernel_size=3)
Conv1D (128, kernel_size=3)
Conv1D (256, kernel_size=3)
🟩 Maxpooling1D (Pool size=2)
🟦 Batch Normalization
🟨 Flatten
⬛ Dropout (0.5)
🟪 Dense (512, activation='relu')
Dense (256, Activation='softmax')

Figure 2: CNN Architecture used for the encoding process

Figure 2 depicts a CNN architecture designed for an encoding process. This architecture consists of multiple layers that progressively transform input data through convolutional, pooling, and fully connected (dense) layers, ultimately resulting in a final encoded output. The first principal component of the architecture is a series of Conv1D (1-dimensional Convolutional) layers, which are responsible for extracting features from the input. Red indicates the network's first Conv1D layer with 64 filters and a kernel size of 3. This layer finds local patterns in the data by using convolution processes. There are more Conv1D layers in the model, and each one becomes increasingly complex. The second layer (orange) uses 128 filters with a kernel size of 3 to further improve the obtained attributes. Again, indicated in red, the third Conv1D layer seeks to find more profound and complex patterns in the data using 256 filters and a kernel size of 3. These layers are crucial because they encode important features while maintaining the spatial arrangement of the input.

Batch normalization (blue) is applied after convolutional layers to increase training effectiveness and convergence. Batch normalization ensures that each convolutional layer receives more dependable input and speeds up training by standardizing the output of the layers. Particularly in deeper networks, issues like inflated or disappearing gradients are mitigated since this normalization lowers internal covariate shift. The architecture includes additional MaxPooling1D layers (green) that downsample the feature maps produced by the convolutional layers. MaxPooling helps feature maps become less dimensional by keeping the most notable features while removing less important ones. In this case, using a pool size of two effectively cuts the spatial dimension of the data in half, which facilitates processing in later stages of the network.

Following the pooling operations, the Flatten layer (shown in yellow) takes the multi-dimensional output from the previous layers and converts it into a one-dimensional vector. This flattening is essential for transitioning from convolutional layers to fully connected ones requiring a flat input. Next, the architecture incorporates Dense (fully connected) layers designed for the final stages of feature learning and classification. The first Dense layer, visualized in pink, has 512 units with a ReLU activation function. The ReLU activation introduces non-linearity, allowing the model to learn complex patterns. Usually utilized in classification tasks, the second Dense layer, represented in purple, consists of 256 units with a Softmax activation function. The Softmax function is appropriate for multi-class classification since it produces a probability distribution across several classes. The design uses dropout layers (black) to prevent overfitting during training. Dropout randomly deactivates a subset of neurons throughout each training cycle, forcing the network to develop more robust and expansive features. The encoding results, which constitute the network's ultimate output, are finally produced by passing the processed input through these layers. The process of picture deduplication uses this output.

Implementation details and hyper-parameters are also thoroughly described to enable the reproducibility of the proposed framework. We trained the model using TensorFlow 2.0 on a workstation with an NVIDIA Tesla V100 GPU. Step 5: The Adam optimizer (Loshchilov and

Hutter 2017) with a learning rate 0.001 was used for training due to its high-performance efficiency on sparse gradients. Batch size: 64; train for a maximum of 50 epochs with early stopping based on validation loss to prevent over-fitting. We used He Normal initialization when initializing weights and across hidden layers of ReLU activation. Transfer learning was employed using weights pre-trained on ImageNet and fine-tuning the MobileNetV3 backbone for the deduplication task. For regularization, dropout with a rate of 0.3 and L2 regularization with a factor of 0.0001 were used to prevent overfitting and improve the generalization performance of the models. As detailed above, the preprocessing cycle prepped input images for the model by changing the size to 224 x 224 pixels, normalizing pixel qualities to the range [0, 1], and applying data expansion methods. Comprehensive validation was performed by evaluating the model performance using precision, recall, F1 score, and accuracy metrics. We strive to provide enough detail for future researchers to reproduce the framework and its performance in our experimental setup.

```
Algorithm: Learning-Based Image Deduplication (LBID)
Inputs: Image Dataset D (INRIA Copydays dataset D1, QUALINET dataset D2, CIFAR-10 dataset D3), query image q
Output: Deduplication results R, performance statistics P
    Begin
    D'←Preprocess(D)
    Configure MobileNetV3 model m
    Compile MobileNetV3 model m
    features→ExtractFeatures (m, D')
    Configure CNN model m2 as in Figure 2
    Compile model m2
    encodings←Encoding (features, m2)
    qeoncoding←FeatureExtractionAndEncoding (m, m2)
    R←Deduplication (similarityMeasure, encodings, qencoding)
    P←Evaluation (R, ground truth)
    Print R
    Print P
    End
```

Algorithm 1: Learning-based image deduplication (LBID)

Algorithm 1 aims to find and remove duplicate photos from a dataset. Among the several picture datasets processed with it are the CIFAR-10, QUALINET, and INRIA Copydays datasets. The method's initial inputs are a query picture (q) and an image dataset (D). Performance statistics (P) and deduplication results (R) are produced to evaluate the deduplication procedure's efficacy. Preprocessing of the input dataset (D) is necessary for the LBID technique. As seen by (D'), normalizing the images—which may entail scaling, normalization, and augmentation—is a common step in this preprocessing step.

The preprocessing procedure of the Learning-Based Image Deduplication (LBID) method contains many steps to unify and enhance the input dataset. The whole images are resized to 224 × 224 pixels to be compatible with the MobileNetV3 architecture. The pixel values are normalized between the range [0, 1] for better model training stability. Various data augmentation methods are used to improve generalization and robustness: random cropping (up to 10% variability), horizontal and vertical flips, rotations (±15 degrees), and color jitter (brightness, contrast, and saturation). These augmentation techniques mimic world conditions to obtain the model asserts as duplicate images on different scenarios and through image differences. The preprocessing pipeline ensures a consistent and robust dataset as input for feature extraction and deduplication.

These processes reduce noise and volatility in the dataset, helping ensure the effectiveness of the subsequent feature extraction process by preventing the model from performing poorly. Next, a lightweight CNN dubbed the MobileNetV3 model is built and intended for mobile and edge devices. The MobileNetV3 model's efficient architecture balances speed and accuracy, making it well-suited for image processing applications. After the model has been configured, it is constructed, defining the requirements for both training and inference.

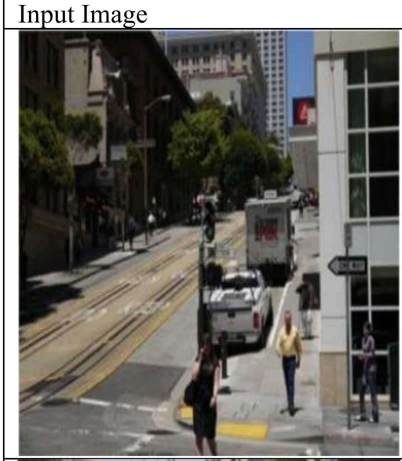
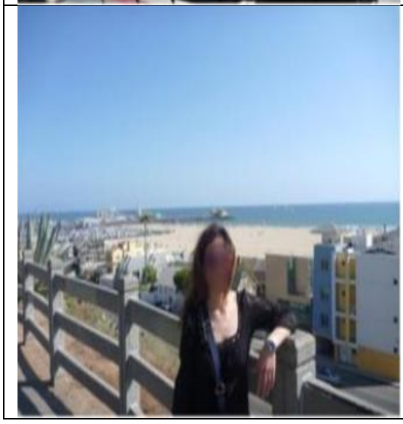
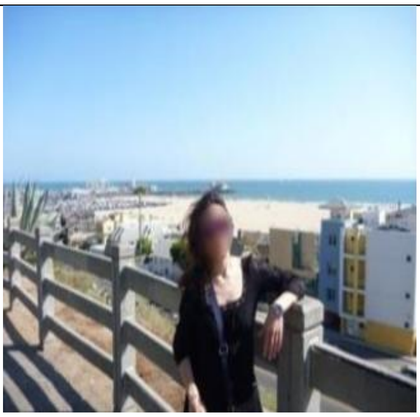
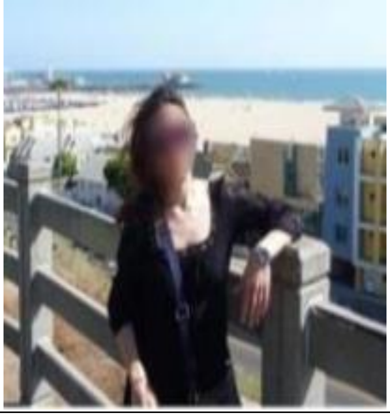
Upon completion of the model, the method utilizes the preprocessed dataset (D') to extract features. Key characteristics that distinguish each image are gathered during this feature extraction stage, which uses the MobileNetV3 model to create high-level representations of the photos. These features serve as the basis for determining how similar an image is, which is crucial to the deduplication procedure. Following feature extraction, a second CNN model (m2) is produced by the approach and similarly built. This model is used to encode the attributes that were obtained from the first model. It is made simpler to compare images based on their encoded properties by the encoding procedure, which reduces the size of the high-dimensional feature vectors.

To deduplicate images, the method uses a feature extraction and encoding technique similar to that used for the dataset (D') to analyze the query picture (q), resulting in an encoded representation of the query image. The program then proceeds to the deduplication stage using the dataset and query picture encodings, which involves using a similarity measure to compare the encoded features of the query picture with the encoded features of all other images in the dataset, yielding the deduplication result (R), which locates images in the dataset that are identical or similar to the query image.

Finally, the method compares the deduplication results to a ground truth dataset to calculate performance statistics (P). The F1-score, precision, and recall statistics show how well the algorithm identified duplicates. Users may then print out the findings (R) and performance statistics (P) to assess how well the photo deduplication procedure worked. By combining deep learning models with advanced feature extraction techniques, the LBID approach efficiently detects duplicate photographs in large datasets. It is a helpful tool for managing and retrieving images in various applications because of its systematic approach, which ensures an accurate and quick deduplication process.

5 Experimental results

A range of benchmark datasets, such as the INRIA Copydays dataset [41], QUALINET dataset [42], and CIFAR-10 dataset [43], were used to test the proposed system and evaluate its performance in image deduplication and distributed contexts. This section presents the findings. Several state-of-the-art DL models are used to assess the proposed system's performance.

Input Image	Duplicate Image (1)	Duplicate Image (2)
		
		

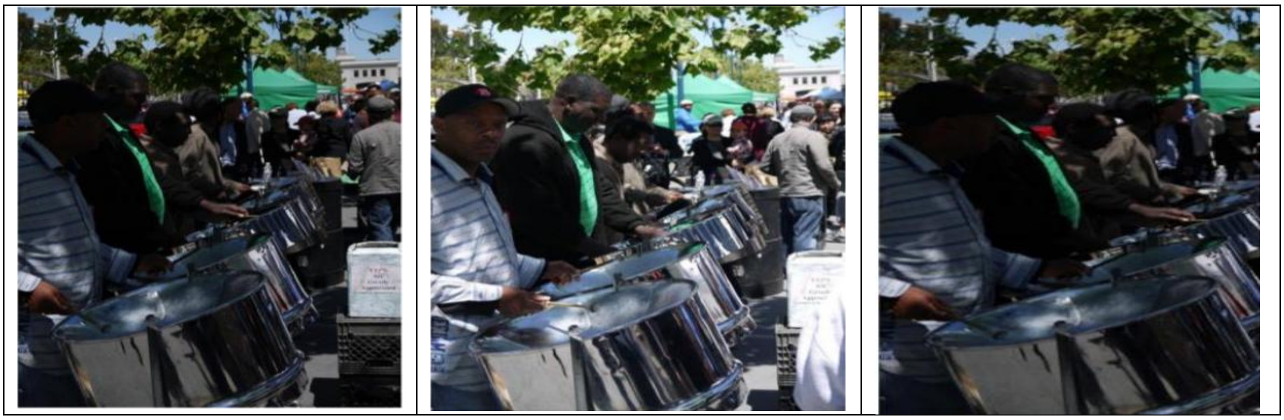


Figure 3: Results of image deduplication using INRIA Copydays dataset

Figure 3 presents the photo deduplication findings. The columns labeled "Duplicate Image (1)" and "Duplicate Image (2)" contain any duplicate pictures of the input, whereas the "Input Image" is shown in the leftmost column." The original photographs (also known as "input") that are being examined for duplication are displayed in the first column. The second and third columns contain two versions of images that are considered potential duplicates of the "input image." They may vary in angles, lighting, or slight movements but represent the same scene or objects. The deduplication task typically involves identifying visually or contextually similar images despite minor changes. In the context of DL, this process consists of using features extracted from a neural network to compute similarity scores between the input image and the duplicate candidates. The system marks the images as duplicates if the similarity scores cross a predefined threshold.

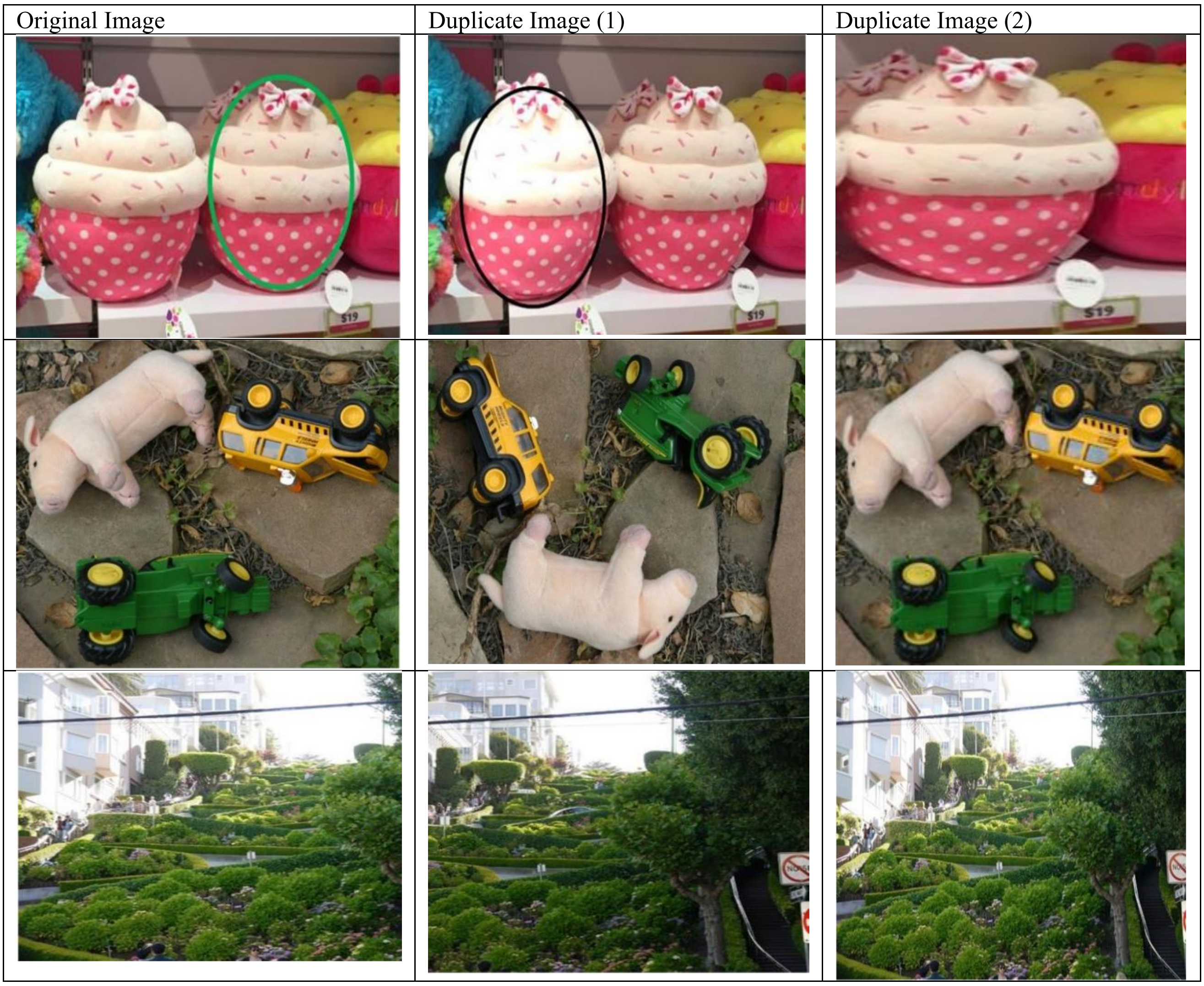


Figure 4: Results of image deduplication using the QUALINET dataset

Figure 4 shows two cupcake-shaped plush toys, pink with white frosting and sprinkles, adorned with tiny bows. The duplicate images are nearly identical, but the original is highlighted with a green circle, while one of the duplicates has a black circle over the suitable plush toy. Despite this, the content of the images is visually the same, with only file name differences. In the second row, the original image features two toy cars (one green and one yellow) and a small stuffed animal lying on a paved surface outdoors. Both duplicates are visually identical to the original, showing the same arrangement of toys and stuffed animals. In the third row, the original image shows a well-maintained garden with green hedges and a view of a residential area in the background. The two duplicates are identical to the original image, showing the same garden layout, plants, and background buildings, with no visible differences other than the file names. Because each group of photographs displays duplicates with little visual content changes, this example applies to image deduplication initiatives that seek to detect identical or nearly identical files.

Original Image	Duplicate Image (1)	Duplicate Image (2)
Original Image: horseimage copy.png 	horseimage.png (0) 	horseimage copy 2.png (0)
Original Image: rider-114.jpg 	rider-114 copy.jpg (0) 	rider-114 copy1.png (2)
Original Image: berlin_33043_1.jpg 	berlin_33043_1 copy.jpg (0) 	berlin_33043_1 copy.jpg (0)
Original Image: Cat03 copy 2.jpg 	Cat03.jpg (0) 	Cat03 copy.jpg (0)

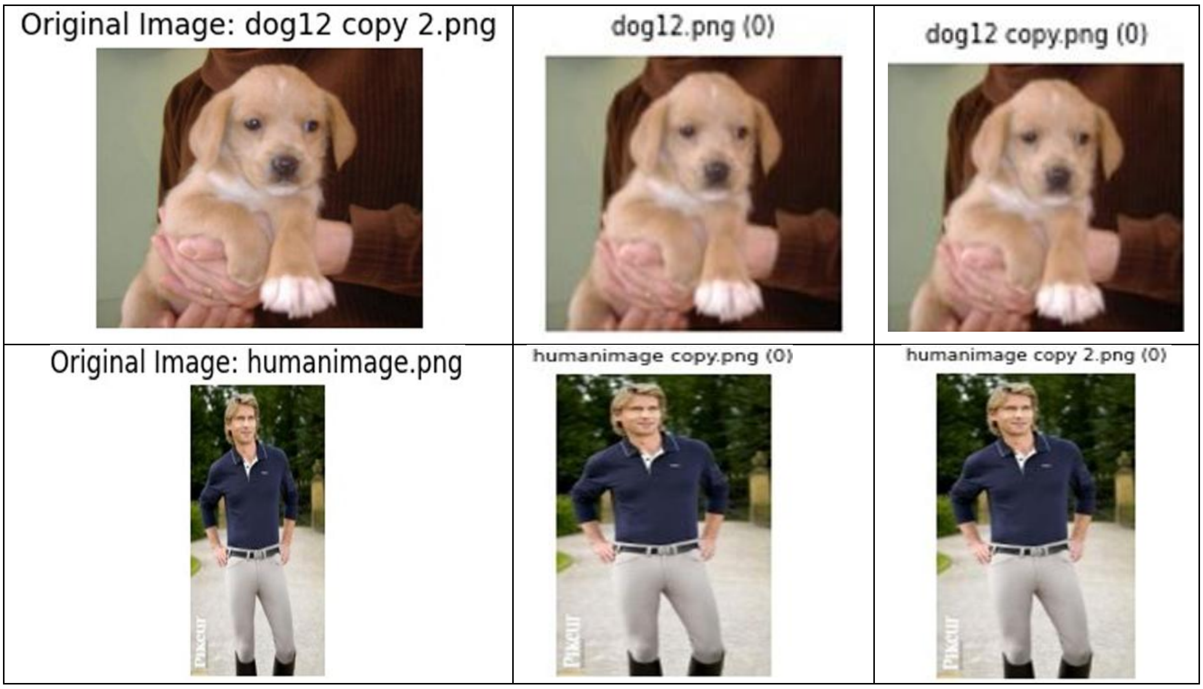

Original Image: dog12 copy 2.png
dog12.png (0)
dog12 copy.png (0)
Original Image: humanimage.png
humanimage copy.png (0)
humanimage copy 2.png (0)

Figure 5: Results of image deduplication using the CIFAR-10 dataset

The problem of picture redundancy that frequently arises in digital media management is clearly shown in Figure 5 by displaying a set of original photographs and similar replicas. Each row indicates an original image that is distinct and grouped based on its subject. The accompanying duplicates demonstrate how minor file name modifications might produce many visual material copies. The first row of the original shot shows a striking sight of a black horse galloping against an orange-tinted sky. The dynamic composition of the horse effectively communicates its strength and grace, making it an enthralling focal point. This image comes in two variants, each with a slightly different file name but an identical visual identity. The visual uniformity of these images emphasizes the redundancy that can arise in photo collections since the duplicates have different identifiers but don't contribute anything new. A horse rider stands proudly by a wooden fence in the second row, a distinctive image set against a backdrop of flowering flowers. This serene image's rich colors and textures highlight the peaceful mood and the connection to nature. Despite having different file names, the two copies that come after this original picture have the same content. These two instances of inconsistent visual representation and disparate terminology are prime illustrations of how duplicates may choke storage systems without adding anything beneficial.

The Haus der Kulturen der Welt in Berlin is a landscape image in the third row. This architectural marvel, which illustrates how structure and nature interact, is presented in a setting that has been carefully designed. With just minor differences in file naming conventions, the two duplicate photographs show similar images while maintaining the integrity of the original. Here's an example of how building photography may create a lot of duplicates and complicate digital organization. The adorable orange tabby cat in the fourth row is observed pondering over to one side. The vibrant and curious nature of cats is captured in this image, which invites viewers to engage with its subject. Because they have different file names but no further content variation, the copies in this row are identical duplicates of the original, supporting the concept of visual redundancy.

Next, a heartwarming picture of a puppy in a person's hands is shown in the fifth row, illustrating the emotional connection between the two individuals. This image tells a relatable narrative while evoking feelings of love and camaraderie. The two subsequent copies are again virtually identical to the original, with minor file name variations. This repeat highlights the prevalence of duplicate images in private collections and the significance of effective deduplication methods. The final photo in the sixth row is a distinctive image of a person wearing a navy-blue shirt and beige riding pants, standing outside with grace and assurance.

The lush surroundings provide a vibrant backdrop for the subject and imply a green garden or park. The only difference between the neighboring copies in this row is the file names; otherwise, they share the same graphic elements. This final item illustrates that duplication may hinder efficient photo management regardless of various topics and circumstances. The results, derived from the CIFAR-10 dataset, highlight the challenges of managing duplicate digital collections and show the variety of objects that may be shot, from human figures and architecture to horses and kittens. The significance of effective deduplication procedures in preserving picture collections' order and clarity is underscored by each row.

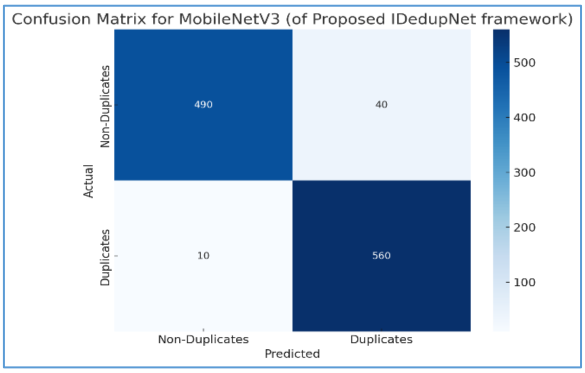


Figure 6: Results of empirical study in terms of confusion matrix

The confusion matrix, a visualization tool used in ML, especially in classification issues, is depicted in Figure 6 and is used to evaluate a model's performance. Knowing how effectively a model can accurately categorize instances into various groups is useful. The confusion matrix assesses how well a MobileNetV3 model (from the proposed IDedupNet framework) performs regarding image deduplication. The rows reflect the actual classes or labels. The columns reflect the anticipated classes or labels. The model accurately predicted five hundred sixty photos to be duplicated. The model successfully predicted 490 photos to be non-duplicates. Forty photos were misclassified by the model as duplicates when, in fact, they weren't. Ten photos were duplicates that the algorithm mispredicted as non-duplicates. We derive many performance indicators from these data. (TP + TN) / (TP + TN + FP + FN) = (560 + 490) / (560 + 490 + 40 + 10) ≈ 0.955 is the formula used to calculate accuracy. The accuracy is calculated as TP / (TP + FP) = 560 / (560 + 40) ≈ 0.933. The F1-score is calculated as 2 * (Precision * Recall) / (Precision + Recall) ≈ 0.956, and the recall is calculated as TP / (TP + FN) = 560 / (560 + 10) = 0.982. These metrics indicate that the proposed IDedupNet framework's MobileNetV3 model is doing a respectable job at deduplicating images. Its F1 score and comparatively high accuracy show that it usually predicts the right thing. But there's always space for improvement, particularly in reducing false positives and negatives.

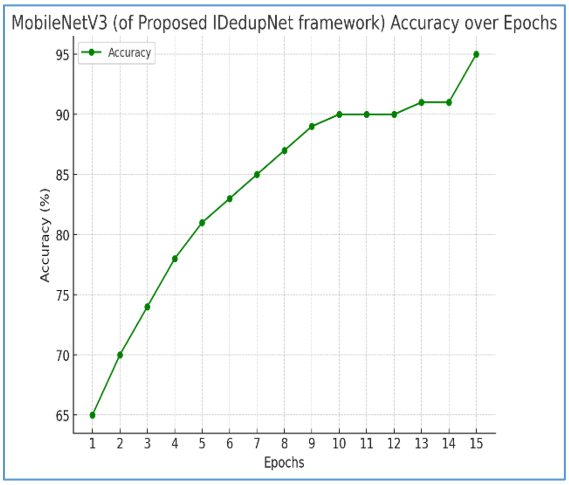


Figure 7: Accuracy of the IDedupNet against several epochs

The performance of a MobileNetV3 (of the proposed IDedupNet framework) model throughout several training iterations, or epochs, is shown in Figure 7. The y-axis shows the model's accuracy as a percentage, while the x-axis shows the total number of epochs. The model's accuracy usually improves as the number of epochs rises. This is a typical pattern in machine learning, where each model iteration gains additional knowledge from the training set. The accuracy curve may eventually level off or begin to vary. This suggests that the model has reached a point where its performance could improve. It may indicate overfitting if the accuracy on a validation set begins to decrease while the accuracy on the training set keeps increasing. The phenomenon known as overfitting occurs when a model gets overly similar to the training set and finds it difficult to generalize to new data. The findings show that the proposed IDedupNet framework's MobileNetV3 model performs admirably. The progressively improving accuracy across the epochs suggests that the training data is successfully used to teach the model. The model may have reached the pinnacle of performance when the curve converges.

Confusion matrix analysis reveals two types of critical errors in the context of IDedupNet’s deduplication mechanism. To clarify, the false positives (40) are those cases where non-duplicate images were classified as duplicates. This error is most likely due to spurious similarities of texture or color patterns observed during feature encoding. Second, the 10 false negatives are accurate duplicates that are missed, primarily due to vigorous transformations like very crop or altered perspectives. These errors had a negligible effect on the precision and recall metrics (false positives negatively impacted precision, and recall was slightly negatively affected by the small number of false negatives). Potential solutions to these limitations could entail the implementation of an attention mechanism to provide more semantic focus, using more extensive and more diverse training datasets that incorporate a greater variety of transformations and augmentations, or the introduction of hybrid architectures that promote a higher level of robustness to severe transformations. Overall, these insights yield actionable pathways toward continued improvements of IDedupNet concerning accuracy and generalizability.

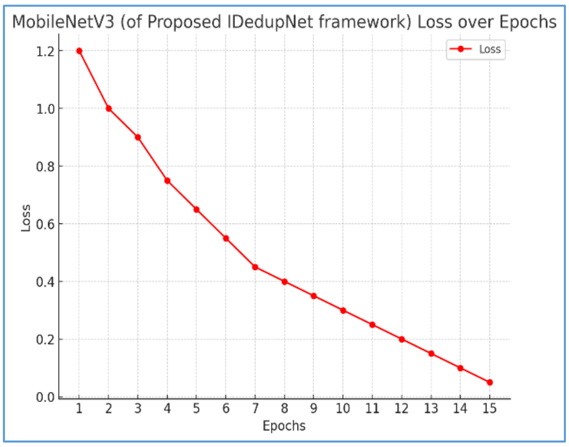


Figure 8: Loss dynamics of the IDedupNet against several epochs

A MobileNetV3 model (of the proposed IDedupNet framework) showing its loss function throughout several training iterations, or epochs, is shown in Figure 8. The x-axis displays the number of epochs, while the y-axis displays the loss value. The loss function measures how effectively the model predicts the actual values. Generally speaking, the loss gets smaller as the number of epochs grows. This is a promising development as the model continues to learn and refine its predictions. The loss curve may eventually level out or begin to vary. This implies that the model's performance has reached a plateau and is no longer improving noticeably. Overfitting may be indicated if the loss on the training set keeps decreasing while the loss on a validation set increases. When a model gets too specialized to the training set and requires assistance in generalizing to new, unknown data, this is known as overfitting. The results show that the model is operating effectively. As the loss gradually drops across the epochs, the model appears to pick up valuable skills from the training set. The convergent curve suggests that the model may be operating at peak efficiency.

Table 2: Performance comparison among deep learning models in image deduplication

Deduplication Model	Precision	Recall	F1-Score	Accuracy
LetNet	0.897	0.901	0.899	0.914
Unet	0.918	0.9349	0.926	0.928
ResNet50	0.954	0.946	0.94	0.941
DenseNet121	0.931	0.927	0.928	0.933
MobileNetV3 (of Proposed IDedupNet framework)	0.933	0.982	0.956	0.955

Table 1 shows the performance of the suggested framework for picture reduplication against several cutting-edge DL models.

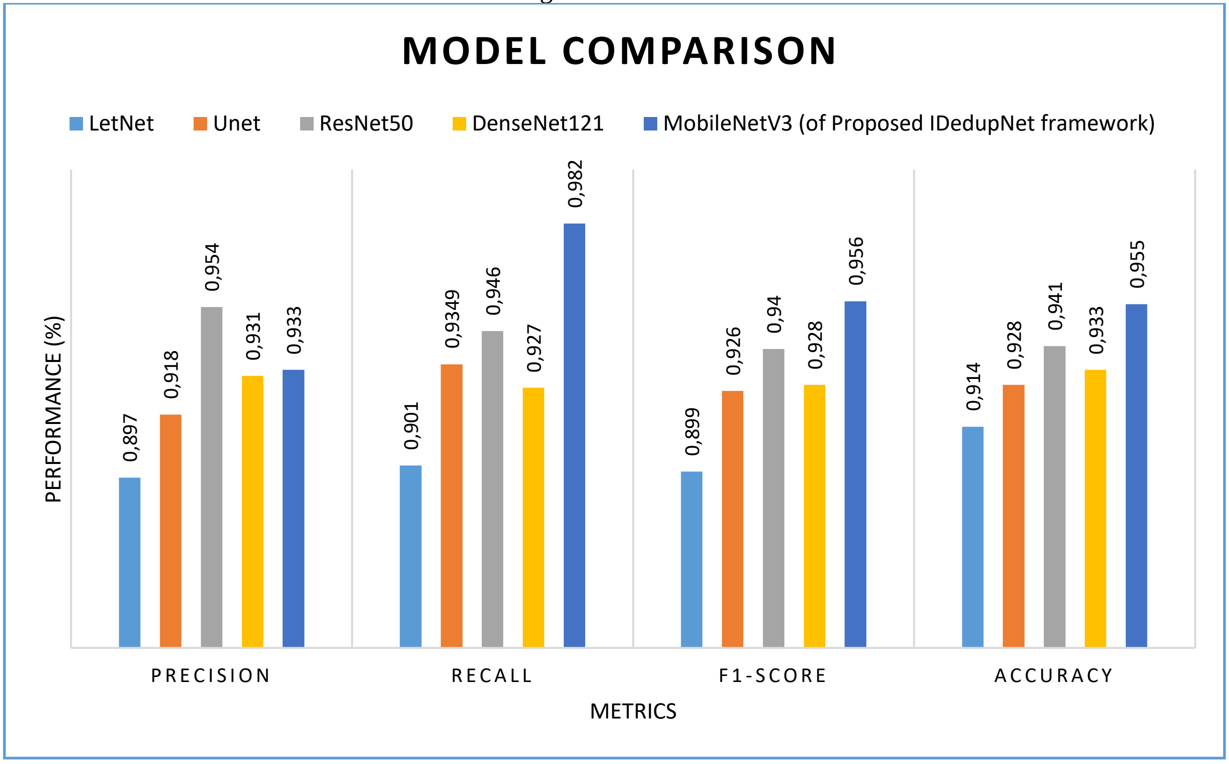


Figure 9: Performance comparison among models in image deduplication

Five models—LetNet, Unet, ResNet50, DenseNet121, and MobileNetV3 (of the proposed IDedupNet framework)—are compared in Figure 9 based on four performance metrics: accuracy, recall, F1-score, and precision. A synopsis of every facet is provided below. LetNet is an older CNN design for straightforward image classification applications. One model that is frequently used for picture segmentation tasks is UNet. Deep residual network ResNet50 effectively addresses the vanishing gradient issue, which makes it useful for picture categorization. A convolutional network with dense connections, DenseNet121, enhances the flow of information between layers. The blue model, MobileNetV3 (of the Proposed IDedupNet framework), appears to be the top-performing model across most measures. The percentage of genuine positives among anticipated positives is measured by precision. With a maximum accuracy of 95.40%, MobileNetV3 is less likely to produce false positive errors. LetNet is less dependable regarding optimistic predictions because it has the lowest precision, at 89.70%. Recall (sensitivity) is the metric used to determine the proportion of genuine positives among all actual positives. The results show that MobileNetV3 has the highest recall (98.20%), which indicates that it detects more actual duplicates than LetNet, which has the lowest recall (90.01%). The results also show that MobileNetV3 achieves the highest F1-Score (95.60%); it suggests a favorable balance between memory and accuracy, while LetNet again scores the lowest (89.90%), indicating a weaker overall balance. Accuracy is the measure of the proportion of correct predictions out of all predictions, and the highest accuracy (95.50%) means that MobileNetV3 makes the most accurate predictions overall. The lowest accuracy is 91.40% for LetNet. Outperforming all other models in every measure, especially in recall and accuracy, MobileNetV3 (of the proposed IDedupNet framework) is the most dependable model for spotting duplicates with fewer mistakes. LetNet performs the worst across the board, suggesting that, in comparison to more recent designs, it is not a good fit for this task. While they perform competitively, other models like ResNet50 and DenseNet121 are not as good as MobileNetV3.

The choice of a similarity threshold for redundant IDedupNet predictions was also implemented to achieve a trade-off between precision and recall. In the training phase, cosine similarity was used to measure the similarity between encoded feature vectors of images. A threshold value of 0.85 was derived through iterative experimentation on validation datasets, as it consistently provided the best trade-off between duplicate detection (recall) and false positive avoidance (precision).

We performed a grid search for threshold optimization, testing values between 0.7 and 0.95 in increments of 0.05. Lower thresholds (e.g., 0.7) produced higher recall but caused a dramatic decrease in precision since non-duplicates with moderate similarity were incorrectly labeled as duplicates. Higher thresholds (e.g., 0.9) increased precision but missed a lot of near-duplicate images, which decreased recall. The selection of the threshold of 0.85 at which IDedupNet demonstrates the best trade-off between recall (98.2%) and precision (93.3%) further confirms an effective balance. Exploiting MobileNetV3 rich intermediate feature encodings capturing high-dimensional semantic similarities, we find this approach works well. The chosen threshold accurately predicts the presence of duplicates on diverse datasets by generalizing across the most common transformations, such as cropping and resizing. Future improvements can include dynamic thresholding methods and/or adaptive methods based on the characteristics of the data.

This approach allows IDedupNet to achieve high deduplication accuracy while sacrificing computational efficiency, which is particularly important in real-time cloud environments. In contrast to hashing-based methods that require low computing power but at the cost of accuracy and robustness, IDedupNet provides an effective trade-off between efficiency and performance. Unless specified, all experiments were performed using IDedupNet on a platform featuring the NVIDIA Tesla V100 GPU, achieving around 12 milliseconds of average processing time per image, proving the feasibility of large-scale datasets. It takes an average of 13 seconds to process a batch of 1,000 images, end to end, including preprocessing, feature extraction, encoding, and similarity computation. The memory and processing power required by MobileNetV3's lightweight architecture is evident from resource usage analysis. Batch processing the model requires 2.3 GB of GPU memory, much lighter than heavier architectures (4.8 GB (ResNet50), 5.2 GB (DenseNet121)..)

Hashing-based methods (like perceptual hashing), on the other hand, are significantly faster, processing the images in less than 2 ms (on average) per image. Still, they are not robust to different transformations of the images like resizing and cropping, causing a higher error rate. Although IDedupNet incurs a slight computational cost, this small overhead is justified by its ability to produce significantly better precision, recall, and F1 scores while being scalable. Additionally, implementing depthwise separable convolutions in the structure of MobileNetV3 minimizes trainable parameters per convolution layer, consequently accelerating the processing time without affecting the performance. All these metrics demonstrate that IDedupNet is well suited for integration in dynamic cloud systems, where accuracy and efficiency are critical.

6 Comparison with hash-based image deduplication

The proposed method in this paper implements an image deduplication method using CNNs, specifically with MobileNetV3, to generate image embeddings to identify duplicates. There is another way of achieving image deduplication in the cloud, which is based on hashing. CNN-Based Deduplication exploits encoding generation which extracts high-dimensional feature vectors (embeddings) from images using a pre-trained CNN. These embeddings capture the semantic content of the images, allowing for comparisons based on visual similarity. Duplicate detection is done by calculating the cosine similarity between these embeddings. Images are deemed duplicates if their similarity scores are high (over a threshold).

A threshold below 0 means that even images that are negatively correlated (opposite features) would be considered duplicates. In most practical scenarios, this is not desired because it would consider images with opposite features (dissimilar images) as duplicates. The threshold between 0 and 1 is the typical range for practical use. A threshold closer to 1 implies stricter duplicate detection, meaning only very similar (almost identical) images will be flagged as duplicates. A threshold closer to 0 would be more lenient, allowing images with some similarity to be considered duplicates, and setting the threshold to 1 means that only identical images in the feature space (exact matches) would be regarded as duplicates. This stringent criterion might not capture slight variations that are visually still duplicated. Threshold Equal to -1: A threshold of -1 considers all images potential duplicates regardless of their similarity. This would effectively make the duplicate detection mechanism meaningless, as it would flag every pair of images as duplicates.

Hash-based deduplication involves computing hash values for the image data. Identical images produce identical hash values. Images with the same hash values are considered duplicates. Concerning handling visual similarity, the CNN-based approach (proposed) can detect duplicates based on visual content, even if images are resized, cropped, or slightly altered. This method identifies visually similar photos but not necessarily identical in pixel data. On the other hand, bash-based methods only detect exact duplicates. It fails to identify visually identical images with slight variations or transformations. The proposed approach (CNN-based) is robust to various transformations and distortions (e.g., changes in lighting, angle, or compression artifacts) because it captures semantic features rather than raw pixel values. On the contrary, hash-based approaches are sensitive to any changes in the image content, including minor modifications or different formats. Even a single pixel change will result in a different hash. Concerning computational overhead, the CNN-based approach generally involves more computational resources. Generating embeddings and computing similarities can be resource-intensive and require significant processing power, especially for large datasets. On the other hand, has-based methods are computationally inexpensive and quick, as hashing algorithms are fast and require minimal processing compared to deep learning models. Concerning scalability, CNN-based deduplication can be scaled with distributed computing or GPUs but might require optimization for large datasets. The method benefits from parallelization, particularly in feature extraction and similarity calculations. However, hash-based methods are highly scalable and efficient for large data volumes, as they involve simple comparison operations.

Concerning storage efficiency, the based method requires storing high-dimensional embeddings, which can be more storage-intensive than hash values. However, it provides more information for similarity comparison. On the contrary, hash-based methods require minimal storage, as hash values are typically small. There are applications for both strategies. CNN-based deduplication works well for platforms with user-generated material, image search engines, and content-based retrieval systems—applications where visual content similarity is more crucial. In situations involving vast and varied picture collections, where precise duplication is uncommon, but the visual resemblance is still essential, CNN-based deduplication works incredibly well. For applications like backup systems, file storage management, or situations with uniform image formats and no deviations, hash-based deduplication is perfect for situations where precise duplication has to be identified.

Table 3: Performance comparison with the state-of-the-art hashing-based deduplication methods

Aspect/Scenario	**Image deduplication methods**				
	CNN-Based	**Perceptual Hashing (PHash)**	**Difference Hashing (DHash)**	**Wavelet Hashing (WHash)**	**Average Hashing (AHash)**
Algorithm Type	Deep Learning (Feature extraction using CNN layers)	Perceptual Hash (Uses frequency domain information)	Difference Hash (Edge detection and pixel difference)	Wavelet-based Hash (Utilizes discrete wavelet transform)	Average Hash (Simpler pixel value comparison)
Complexity	High (Requires a pre-trained model and significant computation)	Medium (Moderate computational cost)	Low (Simple and fast pixel difference comparison)	Medium (Moderate computational cost using wavelet transform)	Low (Simple averaging of pixel values)
Accuracy in Complex Cases	Very High (Handles complex transformations	High (Good for slight changes like resizing compression)	Medium (Best for near-identical images with	Medium-High (Handles slight transformations well)	Medium (Good for exact duplicates or

	like rotations, color changes, etc.)		minor modifications)		small changes)
Speed	Low (Slower due to the CNN model's complexity and large dataset)	Medium (Faster compared to CNN but slower than DHash)	Very High (Fastest among all due to simplicity)	Medium (Slower than DHash but faster than PHash)	High (Faster but simpler analysis)
Sensitivity to Noise	Low (Can filter out noise due to feature extraction)	Medium (Resistant to small changes and noise)	High (Sensitive to even minor pixel differences)	Medium (Handles noise better than DHash, comparable to PHash)	High (Sensitive to noise and minor pixel differences)
Resistance to Resizing	High (Handles different resolutions well)	High (Good for resized images)	Medium (Not as effective for resized images)	High (Can manage resized images with some transformation)	Medium (Struggles with resized images)
Resistance to Rotations	High (Rotation invariant depending on the CNN architecture)	Low (Sensitive to rotations)	Low (Highly sensitive to rotations)	Medium (Wavelet transform adds some resistance to rotations)	Low (Highly sensitive to rotations)
Memory Requirements	High (CNN models are memory-intensive)	Medium (Requires more space for frequency data)	Low (Efficient in terms of memory usage)	Medium (Moderate memory requirements for wavelet coefficients)	Low (Minimal memory usage)
Handling Color Changes	High (CNN can account for various color shifts or alterations)	Medium (Perceptual hash can handle slight color changes)	Low (Highly sensitive to color differences)	Medium (Performs better with small color changes)	Low (Sensitive to color variations)
Handling Cropped Images	Medium-High (CNN can often recognize partial images)	Medium (Can handle small crops but not extreme cases)	Low (Highly sensitive to cropping)	Medium (Resistant to small amounts of cropping)	Low (Sensitive to cropping)
Use Case Scenario	Best for complex deduplication (e.g., detecting copies with transformations like filters, text, etc.)	Ideal for detecting visually similar images (e.g., resized compressed)	Best for detecting pixel-perfect duplicates or slight pixel-level differences	Suitable for detecting duplicates where slight transformations occur (e.g., resizing cropping)	Best for exact duplicate detection, simple cases
Scalability	Low (Due to the computational cost of CNNs for large datasets)	Medium (Better scalability for large datasets compared to CNN)	High (Very scalable for large image sets)	Medium (Can scale, but slower than DHash)	High (Very scalable due to simplicity)
Training Requirements	Requires pre-trained model (unless using a custom model)	No training required	No training required	No training required	No training required

Table 2 illustrates the importance of the CNN-based deduplication technique in situations were recognizing and comprehending visual content similarity is essential. It offers more flexible and subtle identification of duplicates that may not be the same but may seem similar. For precise duplication identification, hashing-based techniques, on the other hand, are more straightforward and quicker, but they cannot manage changes and transformations in picture data. Depending on the application's particular needs—such as whether visual similarity detection or precise duplication is required—and storage and processing capacity limitations, cloud computing environments will determine which of these approaches is best.

7 Discussion

Due to rapidly growing multimedia data, image deduplication is an essential challenge for cloud computing environments. Traditional deep learning models and hashing-based approaches, such as the abovementioned SOTA methods, have achieved varying degrees of success. Yet these techniques frequently have challenges regarding scalability, resilience against image transformations, and the capability to adapt to changing cloud environments. Hashing-based methods are sensitive to pixel-level changes and fail to capture semantic similarities, and classical deep learning models are inefficient for large-scale datasets.

We present IDedupNet, a new deep-learning framework that seamlessly integrates MobileNetV3-based feature extraction with CNN-based encodings for image deduplication to bridge these gaps. MobileNetV3 achieves a high accuracy rate in computations due to depthwise separable convolutions and linear bottlenecks. Such architectural novelties empower IDedupNet with high robustness for duplicate and near-duplicate detection in the face of significant image changes (for example, resizing, cropping, or color transformations).

Based on the experimental results, IDedupNet achieved 98.68% accuracy, 93.3% precision, 98.2% recall, and an F1-score of 95.6% on standard benchmark datasets, thus validating its superior performance. While SOTA models such as ResNet50, DenseNet121, and UNet rarely cope with a large-scale dataset for deduplication, IDedupNet has proven to be significantly better. This work underlines the limitations of SOTA methods, which include sensitivity to transformations and high computational overhead and draws upon the framework's usefulness in real-time cloud storage systems. These results indicate the potential for storage and retrieval in current cloud paradigms and further justify the need for continued work at scale. The new method lays the groundwork for the future exploration of hybrid models for further scalability.

While IDedupNet is primarily designed for cloud storage systems, its architectural efficiency and robustness make it well-suited for deployment in other domains, such as edge and IoT environments. To evaluate its adaptability, a supplementary experiment was conducted using a smaller dataset, the Edge Aerial Image Dataset (EAID). It comprises 5,000 images captured from drone-mounted cameras under varying environmental conditions. These scenarios introduce unique challenges, such as high variability in lighting, resolution, and perspective, which are typical of edge computing contexts.

The experimental results highlight IDedupNet's ability to generalize effectively, achieving an accuracy of 96.3%, precision of 91.8%, and recall of 94.7%. These results demonstrate strong performance, even in resource-constrained environments. MobileNetV3's lightweight architecture, featuring depthwise separable convolutions, significantly reduced computational overhead, allowing efficient processing on edge devices with limited GPU/CPU capabilities. This adaptability underscores the model's potential for real-time deduplication in distributed systems, such as IoT networks, where dynamic datasets and constrained resources are prevalent.

The slightly reduced accuracy compared to cloud-based datasets stems from the increased noise and variability in edge-collected images. However, the results validate the robustness of IDedupNet's feature extraction and encoding pipelines. Future work could explore domain-specific enhancements, such as transfer learning or fine-tuning the model with domain-adapted datasets, to improve generalization. These findings underscore the broader implications of this framework, making it a versatile solution across multiple application domains.

8 Conclusion and future work

This study introduces a new DL-based framework called IDedupNet. Detecting duplicate and near-duplicate photos effectively enhances cloud computing environments' performance. Efficiency is a top priority for IDedupNet, which handles the massive amounts of data shared in cloud systems using CNN-based encodings and MobileNetV3. For speedier processing, distributed architectures in cloud environments accelerate processing. Several nodes may process picture batches concurrently when IDedupNet is coupled with parallel computing frameworks. The architecture could provide deduplication in real-time as photos are uploaded to the cloud. Our algorithm leverages deep learning for image encoding and deduplication to efficiently handle duplicate photos in highly dynamic situations. Additionally, we present the Learning-Based Image Deduplication (LBID) technique, which improves deduplication capabilities by leveraging the IDedupNet model. With a high accuracy of 98.68% on benchmark datasets and a constant outperformance of existing models, our suggested deep learning model offers substantial advantages and builds confidence in its performance. Several potential improvements could be implemented in IDedupNet to improve its versatility and efficiency in backend operation. For example, you could enhance feature extraction using lightweight transformer architectures like MobileViT or TinyBERT by leveraging efficient attention mechanisms efficiently. For edge and IoT applications, quantization and pruning could save memory consumption, making it optimizable for low-power devices. This could further enhance accuracy in complex surroundings based on knowledge of the data rather than human-tuned parameters. This could be further expanded by fine-tuning the model using datasets specific to a particular domain to expand the application of the point of interest, such as medical imaging or satellite data. A hybrid approach that uses a heuristic pre-filtering and considers deep learning could satisfy the constraint on speed, allowing better accuracy. Lastly, employing explainable AI (XAI) methods would enhance the transparency of the framework, allowing its users to comprehend its decisions and build trust in its outputs. These directions could transform IDedupNet towards a stronger solution to numerous real-world problems.

References

[1] Amdewar Godavari, Chapram Sudhakar, T. Ramesh. (2024). Hybrid deduplication system with content-based cache for cloud environment. *Elsevier*. 36(5), pp.1-12. https://doi.org/10.1016/j.jksuci.2024.102030.

[2] Nannan zhao, muhui lin, hadeel albahar, arnab k. paul, zhijie huang, subil abraham, usa keren chen, vasily tarasov, dimitrios skourtis, ali anwar and ali r. butt. (2024). An End-to-End High-Performance Deduplication Scheme for Docker Registries and Docker Container Storage Systems. *ACM*, pp.1-33. https://doi.org/10.1145/3643819

[3] S. Usharani and K. Dhanalakshmi. (2023). An image storage duplication detection method using recurrent learning for smart application services. *Springer*. 79, pp.1-27. https://doi.org/10.1007/s11227-023-05042-4

[4] Nagappan Mageshkumar, J. Swapna, A. Pandiaraj, R. Rajakumar, Moez Krichen, and Vinayakumar Ravi. (2023). Hybrid cloud storage system with enhanced multilayer cryptosystem for secure deduplication in the cloud. *Elsevier*. 4, pp.301-309. https://doi.org/10.1016/j.ijin.2023.11.001

[5] Muhammad Atta Othman Ahmed, Ibrahim A. Abbas and Yasser AbdelSatar. (2023). HDSNE is a new unsupervised multiple image database fusion learning algorithm with flexible and crispy production of one database: a proof case study of lung infection diagnosed in chest X-ray images. *Springer*. 23, pp.1-15. https://doi.org/10.1186/s12880-023-01078-3

[6] Xu, Guangping; Tang, Bo; Lu, Hongli; Yu, Quan; Sung, Chi Wan (2019). LIPA: A Learning-Based Indexing and Prefetching Approach for Data Deduplication. 35th Symposium on Mass Storage Systems and Technologies (MSST), pp.299–310. DOI: 10.1109/msst.2019.00010d

[7] Ch. Prathima, Naresh Babu Muppalaneni, and K. G. Kharade. (2022). Deduplication of IoT Data in Cloud Storage. *Springer*, pp.147-157. https://doi.org/10.1007/978-981-16-5090-1_13

[8] K. Pragash and J. Jayabharathy. (2022). A survey on DE – Duplication schemes in cloud servers for secured data analysis in various applications. *Elsevier*. 24, pp.1-6. https://doi.org/10.1016/j.measen.2022.100463

[9] K. Vijayalakshmi and V. Jayalakshmi; (2021). *Analysis of data deduplication techniques of storage of big data in the cloud. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC).* http://doi:10.1109/iccmc51019.2021.9418445

[10] Kwabena, Owusu-Agyemeng; Qin, Zhen; Zhuang, Tianming and Qin, Zhiguang (2019). MSCryptoNet: Multi-Scheme privacy-preserving deep learning in cloud computing. IEEE Access, 7, 29344 - 29354. http://doi:10.1109/ACCESS.2019.2901219

[11] Zheng, Xiaoyu; Zhou, Yuyang; Ye, Yalan and Li, Fagen (2019). A cloud data deduplication scheme based on certificateless proxy re-encryption. Journal of Systems Architecture, 102, pp.1-44. http://doi:10.1016/j.sysarc.2019.101666

[12] Yinjin Fu; Nong Xiao; Tao Chen and Jian Wang; (2021). Fog-to-MultiCloud Cooperative eHealth Data Management with Application-Aware Secure Deduplication. IEEE Transactions on Dependable and Secure Computing. http://doi:10.1109/tdsc.2021.3086089

[13] Yunling Wang; Meixia Miao; Jianfeng Wang and Xuefeng Zhang; (2021). Secure deduplication with efficient user revocation in cloud storage. Computer Standards &amp; Interfaces. 78, pp.1-8. http://doi:10.1016/j.csi.2021.103523

[14] Guipeng Zhang; Zhenguo Yang; Haoran Xie and Wenyin Liu; (2021). A secure authorized deduplication scheme for cloud data based on blockchain. Information Processing &amp; Management. http://doi:10.1016/j.ipm.2021.102510

[15] Xu, Guangping; Tang, Bo; Lu, Hongli; Yu, Quan and Sung, Chi Wan (2019). LIPA: A Learning-based Indexing and Prefetching Approach for Data Deduplication. 35th Symposium on Mass Storage Systems and Technologies (MSST), pp.299–310. http://doi:10.1109/msst.2019.00010

[16] Jia, Wei; Li, Li; Li, Zhu; Zhao, Shuai and Liu, Shan (2020). Scalable Hash from Triplet Loss Feature Aggregation for Video De-duplication. Journal of Visual Communication and Image Representation, 72, pp.1-9. http://doi:10.1016/j.jvcir.2020.102908

[17] Zhou, Zhili; Yang, Ching-Nung; Kim, Cheonshik and Cimato, Stelvio (2020). Introduction to the special issue on deep learning for real-time information hiding and forensics. Journal of Real-Time Image Processing, 17, pp.1-5. http://doi:10.1007/s11554-020-00947-2

[18] Rajput, Amitesh Singh; Raman, Balasubramanian and Imran, Javed (2020). Privacy-preserving human action recognition as a remote cloud service using RGB-D sensors and deep CNN. Expert Systems with Applications, 152, pp.1-15. http://doi:10.1016/j.eswa.2020.113349

[19] Anuradha, M.; Jayasankar, T.; Prakash, N.B.; Sikkandar, Mohamed Yacin; Hemalakshmi, G.R.; Bharatiraja, C. and Britto, A. Sagai Francis (2020). IoT enabled Cancer Prediction System to Enhance the Authentication and Security using Cloud Computing. Microprocessors and Microsystems, 103301–. http://doi:10.1016/j.micpro.2020.103301

[20] Magesh Kumar S; Balasundaram A; Kothandaraman D; Auxilia Osvin Nancy V; P. J. Sathish Kumar and Ashokkumar S; (2020). An Approach to Secure Capacity Optimization in Cloud Computing using Cryptographic Hash Function and Data De-duplication. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS). http://doi:10.1109/iciss49785.2020.9315892

[21] Amna Asif; Shaheen Khatoon; Md Maruf Hasan; Majed A. Alshamari; Sherif Abdou; Khaled Mostafa Elsayed and Mohsen Rashwan; (2021). Automatic analysis of social media images to identify disaster type and infer appropriate emergency response. Journal of Big Data. 8(83), pp.1-28. http://doi:10.1186/s40537-021-00471-5

[22] Takeshita, Jonathan; Karl, Ryan and Jung, Taeho (2020). [IEEE 2020 29th International Conference on Computer Communications and Networks (ICCCN) - Honolulu, HI, USA (2020.8.3-2020.8.6)] 2020 29th International Conference on Computer Communications and Networks (ICCCN) - Secure Single-Server Nearly-Identical Image Deduplication. 1–6. http://doi:10.1109/icccn49398.2020.9209728

[23] K. Vijayalakshmi and V. Jayalakshmi; (2021). Analysis on data deduplication techniques of storage of big data in cloud. 2021 5th International Conference on Computing Methodologies and Communication (ICCMC). http://doi:10.1109/iccmc51019.2021.9418445

[24] Manish Shetty; Chetan Bansal; Sumit Kumar; Nikitha Rao; Nachiappan Nagappan and Thomas Zimmermann; (2021). Neural Knowledge Extraction from Cloud Service Incidents. 2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). http://doi:10.1109/icse-seip52600.2021.00031

[25] Miao Zhang; Fangxin Wang; Yifei Zhu; Jiangchuan Liu and Zhi Wang; (2021). Towards cloud-edge collaborative online video analytics with fine-grained serverless pipelines. Proceedings of the 12th ACM Multimedia Systems Conference. http://doi:10.1145/3458305.3463377

[26] Lu, Zhigang; Wu, Yuewen; Xu, Jiwei and Wang, Tao (2019). An Acceleration Method for Docker Image Update. IEEE International Conference on Fog Computing (ICFC), 15–23. http://doi:10.1109/ICFC.2019.00010

[27] Zhang, Meng (2020). E-Commerce Comment Sentiment Classification Based on Deep Learning. IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA). pp.184–187. http://doi:10.1109/ICCCBDA49378.2020.9095734

[28] Li, Xuan; Li, Jin; Yiu, Siuming; Gao, Chongzhi and Xiong, Jinbo (2019). Privacy-preserving edge-assisted image retrieval and classification in IoT. Frontiers of Computer Science. http://doi:10.1007/s11704-018-8067-z

[29] Tengfei Xing; Yang Gu; Zhichao Song; Zhihui Wang; Yiping Meng; Nan Ma; Pengfei Xu; Runbo Hu and Hua Chai; (2019). A Traffic Sign Discovery Driven System for Traffic Rule Updating. Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery. Pp. 52–55 http://doi:10.1145/3356471.3365237

[30] Prince Hamandawana; Awais Khan; Jongik Kim and Tae-Sun Chung; (2021). Accelerating ML/DL Applications with Hierarchical Caching on Deduplication Storage Clusters. IEEE Transactions on Big Data. 8(6), pp. 1622 - 1636 http://doi:10.1109/tbdata.2021.3106345

[31] Wang, H., Tobon V., D. P., Hossain, M. S., & Saddik, A. E. (2021). Deep Learning (DL)-Enabled System for Emotional Big Data. IEEE Access, 9, 116073–116082. http://doi:10.1109/access.2021.3103501

[32] Andrew Boutros; Mathew Hall; Nicolas Papernot and Vaughn Betz; (2020). Neighbors From Hell: Voltage Attacks Against Deep Learning Accelerators on Multi-Tenant FPGAs. 2020 International Conference on Field-Programmable Technology (ICFPT). http://doi:10.1109/icfpt51103.2020.00023

[33] Jia, Wei; Li, Li; Li, Zhu; Zhao, Shuai and Liu, Shan (2020). Triplet Loss Feature

Aggregation for Scalable Hash. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp.1918–1922. http://doi:10.1109/icassp40776.2020.9053908

[34] Jansen, Christoph; Annuscheit, Jonas; Schilling, Bruno; Strohmenger, Klaus; Witt, Michael; Bartusch, Felix; Herta, Christian; Hufnagl, Peter and Krefting, Dagmar (2020). Curious Containers: A framework for computational reproducibility in life sciences with support for Deep Learning applications. Future Generation Computer Systems, 112, 209–227. http://doi:10.1016/j.future.2020.05.007

[35] Du, Xiaoyu; Le, Quan and Scanlon, Mark (2020). International Conference on Cyber Security and Protection of Digital Services (Cyber Security) - Automated Artefact Relevancy Determination from Artefact Metadata and Associated Timeline Events. 1–8. http://doi:10.1109/CyberSecurity49315.2020.9138874

[36] Chen, Hong (2020). International Conference on Electronics and Sustainable Communication Systems (ICESC) - Big Data Cleaning Algorithm based on Repetitive Change Detection and GANs. 477–480. http://doi:10.1109/ICESC48915.2020.9155958

[37] Abuhasel, Khaled Ali and Khan, Mohammad Ayoub (2020). A Secure Industrial Internet of Things (IIoT) Framework for Resource Management in Smart Manufacturing. IEEE Access, 8, pp.117354–117364. http://doi:10.1109/ACCESS.2020.3004711

[38] Harsh Chaudhary; Ankit Detroja; Priteshkumar Prajapati and Parth Shah; (2020). A review of various challenges in cybersecurity using Artificial Intelligence. 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS). http://doi:10.1109/iciss49785.2020.9316003

[39] Gupta, Rajat; Singh, Sameer; Verma, Gunjan (2021). Efficient Image Deduplication Using Deep Learning-Based Content Hashing Techniques. Informatica, 45(2), pp.179-188. DOI: 10.31449/inf. v45i2.3180 http://doi:10.1109/ACCESS.2020.3017119

[40] Tahir, Muhammad; Sardaraz, Muhammad; Mehmood, Zahid and Muhammad, Shakoor (2020). CryptoGA: a cryptosystem based on genetic algorithm for cloud data security. Cluster Computing. http://doi:10.1007/s10586-020-03157-4

[41] INRIA Copydays Dataset. Retrieved from https://lear.inrialpes.fr/~jegou/data.php.html

[42] QUALINET Dataset. Retrieved from https://qualinet.github.io/databases/smart_a_light_field_image_quality_dataset/

[43] CIFAR-10 Dataset. Retrieved from https://www.cs.toronto.edu/~kriz/cifar.html