Multimedia Tools and Applications (2022) 81:40793–40826
https://doi.org/10.1007/s11042-022-13182-7

Deep CNN based online image deduplication technique for cloud storage system

Ravneet Kaur[1] · Jhilik Bhattacharya[1] · Inderveer Chana[1]

Received: 13 March 2020 / Revised: 12 January 2021 / Accepted: 19 April 2022 /
Published online: 13 May 2022

Abstract
Online image detection is one of the most critical components of an image deduplication technique for an efficient cloud storage system. Although extensive research has been conducted in this field, the problem still remains challenging. Deep learning techniques have achieved significant success in solving a variety of computer vision issues and have high potential in image deduplication techniques. Deduplication is an efficient method in a cloud storage system that minimizes redundant data at the file or sub-file level using cryptographic hash signatures. Although significant research on offline image deduplication techniques have been reported, yet limited research is available on online image deduplication techniques. Online image matching accuracy and performance has been a major challenge for online image deduplication techniques to detect exact or near-exact images using feature extraction techniques. These first use feature extraction techniques to extract image features and then match these image features to detect duplicate images. In this paper, we have proposed a Deep CNN based online image deduplication technique for a cloud storage system to detect exact and near-exact images using cross-domains, even in the presence of perturbations in the form of blur, noise, compression, lighting variations and many more. The experimental results show that our proposed deep CNN for online image deduplication technique outperforms in terms of image matching accuracy and performance. The paper also proposed a Hot Decomposition Vector (HDV) for image patch generation to store efficiently dissimilar parts of near-exact images. The experimental results demonstrate that HDV exhibits higher and stable image matching accuracy in all three types of image deformations with relatively small computation time.

✉ Ravneet Kaur
ravneet@thapar.edu

Jhilik Bhattacharya
jhilik@thapar.edu

Inderveer Chana
inderveer@thapar.edu

[1] Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology (Deemed to be University), Patiala, Punjab, India

Keywords Convolutional neural networks · Near-exact image detection · Data deduplication · Online image deduplication technique · Storage systems · Cloud computing · Cloud storage

1 Introduction

Recent development in images and videos retrieval systems include methods to extract significant features from multimedia data. Also, with the advent of cloud computing [2, 51] and exponential growth of digital data, a large number of images and videos are being shared by the users across the world. Flickr has around 6 billion images [40] and Facebook has about 0.25 billion images [40] uploaded daily. According to IDC (International Data Corporation), data volume created and copied in the world is expected to reach 40 Trillion GB by 2020 [15]. Especially the image files in the social networking sites [15], are repeated a number of times, which lead to huge usage of storage space in the cloud storage system. Increasing the storage space in the cloud is costly in terms of additional space and bandwidth [33]. Also, the social networking sites are facing issues in storage, retrieval and analysing a large amount of image data volumes. The image data is most widely copied in the cloud storage system. The researchers need to come up with advanced techniques to remove such duplicate copies of images from the storage system. Data deduplication [50, 82] is one such technique that detects duplicate images from the image storage system.

The techniques to find exact or near-exact images have been a challenge in a large storage system. It led researchers to evolve an efficient, scalable, and reliable image deduplication technique. The social networking sites are getting popular due to a large number of images shared around the world by the users. Most of the images [40] uploaded by the users are either slightly modified or exact copies of an original image. This leads to a huge database [52] of duplicate or near-duplicate images on the storage system. These large numbers of duplicate images are impacting the performance of the image storage system and escalating its cost. Further, it is very hard to efficiently index or retrieve images from a very large image cloud storage system. An efficient real-time or online technique that detects duplicates of an original image from a storage system is required to improve the overall performance in terms of system storage efficiency.

Image deduplication is a major challenge for storage systems, and the objective of this paper is to propose an online deduplication technique to detect exact or near-exact duplicate images in a cloud storage system. The near-exact images are variants of an original image with certain alterations and transformations such as scaling, cropping, rotation, color change, compression and enhancement. A large amount of computation and memory is required to detect online duplicates of an original image. Therefore, an online image deduplication technique is a key parameter for a storage system and has a significant impact on the cloud storage in terms of storage efficiency, cost per gigabyte and energy savings. Also, the minute changes in an image may have pertinent information and hence cannot be discarded or ignored. In this research work, we proposed a CNN based technique for online exact or near-exact image deduplication. Hot Decomposition Vector (HDV) is proposed for image patch generation to efficiently store dissimilar parts of near-exact images.

Figure 1 demonstrates a sample of near-exact images in our image database. These images may refer to indoor or outdoor images of a person, object, scene having variations in terms of color, scaling, rotation, cropping etc. These images may also be captured from different viewpoints. In Fig. 1 the last image in each row is a result of zooming in on the original image or capturing a closer view of face and building image.

Fig. 1 Exact or near-exact images

In a large storage system, image detection technique requires additional resources in terms of CPU, memory and bandwidth [23]. The detection of exact or near-exact image duplicates plays an important role in various fields like storage optimization, image copyright enforcement, video copy detection, social network applications, pathological applications, remote sensing applications and performance of image search engines.

In this article, Deep CNN based an online image deduplication technique is proposed for the cloud storage system. The main objective of this research is to find exact or near-exact images using feature extraction techniques in a cloud storage system using normal as well as perturbed cross-domain images. Deep CNN based online image deduplication detects exact or near-exact images and Hot Vector Decomposition (HDV) generates image patches to store dissimilar parts of near-exact images. This research puts forward that while using CNNs, an appropriate fine-tuning and transfer learning is required for getting desirable performance for domain specific data. However, a deduplication task is not domain specific and hence demands a network which can work for cross-domain images without any prior information. At the same time, it should be able to generate a feature vector that is invariant to different perturbations like lighting, compression, blur, sharpness, noise, etc. In this context, we proposed an architecture that deals with both situations without the use of explicit domain specific training. The other goal of the research is to be able to store the patches for near-exact images such that it can be reconstructed using a base image.

The image deduplication techniques can be broadly categorized as exact image deduplication techniques, near-exact image deduplication techniques and real-time image deduplication techniques. These techniques are further presented in section 2. The online image deduplication is a major challenge for a storage system and our objective is to find exact or near-exact duplicate images in a large scale distributed database.

1.1 Data deduplication

Data Deduplication is an enabling technology for efficient big data storage and management [26]. Deduplication is an effective technique to remove the duplicate data automatically from the cloud storage system and helps in reducing storage costs in terms of space and bandwidth. The growing complexity of duplicate data in storage is a challenging issue for researchers. The storage performance optimization is key to a large distributed storage system. The exponential growth of data has led researchers to focus on the development of such a distributed storage system. Removing and managing duplicate data, efficient data delivery from the storage system to users or applications are important objectives of storage.

Deduplication is a special data compression technique for eliminating redundant data and improving data storage utilization. The technique searches for duplicate data and saves only one copy of the data. It uses pointers that point to the unique copy of data to replace the duplicate copies of it [57]. Deduplication reduces the required storage capacity and addresses the increasing demand for storage [48]. The data deduplication techniques are widely used in backup storage systems like Dropbox, Memopal [59] as the data redundancy rates in such storages are exceptionally high. Data deduplication techniques have also been employed in primary storage systems, virtual machines, WAN, and cloud storage systems like Amazon S3, Microsoft Azure and Google Drive [79]. The deduplication techniques depend upon various types of data such as text, image or video. The techniques are developed on different types of data. With the introduction of social networking sites, text, images and videos are highly redundant and exert an extra load on the cloud storage systems. Researchers are focusing their time and energy to develop efficient online deduplication techniques to remove redundant data.

1.2 Online image deduplication

The automatic removal of duplicate images in a cloud storage system is known as image deduplication. It is a special technique to eliminate redundant images, to improve storage utilization and to reduce storage cost. An online image deduplication technique searches for exact or near-exact images and saves only one copy of the image data and dissimilar patches of images. Online image deduplication directly depends upon the performance of detecting exact or near-exact images. Image deduplication techniques are based on image detection techniques and these are further classified as exact image detection and near-exact image detection. Exact image deduplication techniques detect exactly the same duplicate images and do not consider image alteration or transformations. The near-exact image detection deliberates on image transformation and its modification by cropping [1], scaling, rotation, etc. The technique remains the same for exact or near-exact image deduplication. The only difference lies in the storage of image transformation for near-exact images.

This paper contributes to exact or near-exact duplicate image detection for an online image deduplication technique that extracts image features and proposes an online image detection procedure. The existing research on image detection is mainly focused on offline exact or near-exact image detection approaches. Also, the paper proposed a novel online technique to detect near-exact images for a very large distributed, cloud storage system. The main contributions of this paper are:

a) A CNN based online image deduplication technique to detect exact or near-exact images. The paper also proposes multi-classifier decision fusion for exact or near-exact image detection. A Fine-Tuned AlexNet is proposed for cross-domain online image deduplication. The results are presented and key performance indicators are compared with existing CNN techniques.
b) Hot Decomposition Vector (HDV) image feature extraction technique is proposed to improve matching accuracy and efficiency for near-exact images. HDV performance is presented and compared with traditional image feature extraction techniques and their combinations.

This paper is organized into five sections. Section 2 explains the background of exact and near-exact image detection techniques and existing CNN based image detection techniques for deduplication. The state of the art in techniques for image detection are discussed in this section. The challenges in exact or near-exact image detection and methodology of proposed deep CNN based online image deduplication technique is discussed in Section 3. Section 4 demonstrates the experimental setup and analysis of existing techniques with our proposed technique. The comparison and results are discussed in this section. Finally, Section 5 concludes this article by summarizing the contribution and future research scope.

2 Background

In this section, the state-of-the-art research techniques to detect exact or near-exact images are discussed. The existing CNN-based techniques are also presented. This paper categorized contemporary research work based on exact, near-exact and CNN based image detection techniques. Also, we have highlighted existing real-time duplicate image detection techniques.

2.1 Image deduplication process

A basic image deduplication process steps involve chunking, hash calculation, fingerprint matching or index lookup and chunk storage techniques as described below:

- Data Chunking or Data Granularity: Granularity is the smallest and basic unit to detect duplicates. Granularity refers to the method used for partitioning the file into chunks. Chunks are categorized as file-level chunking and sub-file level chunking. The file-level deduplication operates on a file as granularity. It is also known as single-instance storage. The sub-file level or block-level deduplication is performed at a finer granularity level. In this, a file is further divided into fixed or variable size chunks. These unique chunks are stored in the storage system. The variable size chunks provide better matching efficiency to identify redundant data while the fixed-size chunking is easy to implement. In image processing, the entire image is processed as a whole. Few techniques implemented an image management system where the images were divided into static chunking or data blocks [78, 86]. However, [78] particularly focussed on virtual machine images, which are treated at the file level. [86] mainly focuses on family photographs to divide images into different data blocks using Block Truncation Coding (BTC).
- Fingerprint Calculation: Cryptographic hash functions are used to compare duplicate chunks. The hash values are calculated for each chunk of data and the hash keys are stored as an index for future comparisons. Different cryptographic hash algorithms like MD5, SHA-I, Rabin fingerprinting are used to detect duplicate chunks. In the case of images, hashes are applied directly on the intensity of images. The images are pre-processed and image feature extraction techniques like Scale-Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Principal Components Analysis-SIFT (PCA-SIFT) and Binary Robust Invariant Scalable Keypoints (BRISK) are widely used to extract features of images. The hashing algorithms are applied on image features to generate hash values [19].

- Fingerprint Matching: Every chunk's hash value is indexed and compared with a hash value already stored in the existing hash table in the database. The image matching techniques exploit an approximate distance measure matching of image features [7, 53].
- Placement or Chunk Store: Identical chunks having the same hash values are replaced with logical pointers. Such pointers point to already stored chunks in the database. Unique chunks are added in the hash table and the index of chunks is updated. From the point of view of an image, unique images are stored in the database and images having the same hash values are discarded and pointers are used that refer to the original or base image.

2.2 Exact image duplicate detection techniques

Exact Image Duplicate Detection Techniques detect the exact copy of an original image without considering any image transformation. Some of the notable techniques are discussed here. Anum Javeed Zargar et al. [86] discussed a histogram refinement algorithm for image deduplication. The technique was applied for detecting and discarding duplicate copies of electricity cards in the system. The electricity cards contained family photographs, which were exploited for the deduplication procedure. In this work, the block truncation coding technique is used for comparing the images in a large-scale database. This technique exploits a variable size chunking for family photos matching. Images of the same block are placed in the same cluster. This approach exhibits better results in the form of space and bandwidth. N. Pattabhi Ramaiah et al. [61] proposed similarity-based image deduplication using Content-Based Image Retrieval (CBIR). Images are collected from family cards of different districts and feature extraction, clustering and deduplication is done. The features are extracted based on the color and texture content of images. The extracted features are clustered into 23 clusters of each district and these are further clustered into different classes. Histogram refinement of the image is calculated for comparison with other images for the deduplication process. The images with similarity less than 1500 are displayed and checked manually in order to avoid false-positive images. Ming Chen et al. [8] proposed B+ trees for indexing feature vectors of images and Haar wavelet decomposition is used to extract edge information of images. Manhattan distance is used to detect the similarity between two images. Though this technique achieved a high deduplication ratio but is not scalable to a large number of images. Xuan Li et al. [41] proposed a secure Perceptual Similarity Deduplication Scheme (SPSD) for a small-scale database. In this proposed scheme, perceptual hash algorithms generate image signatures to compare similar images. The images and its pHash signatures are encrypted using a group key on Cloud and duplicity is checked by calculating the hamming distance between two image signatures by setting a certain threshold. The method achieves efficient deduplication, in terms of bandwidth and storage. Jyoti et al. [24] proposed a new hybrid compression approach that uses the combination of PCA, Set Partitioning in Hierarchical Trees (SPIHT) and compressive sensing techniques to compress the image. Convergent encryption encrypts the compressed image and the final image is sent to the server for image deduplication to detect duplicate images. The experimental results show the efficiency of the proposed method in terms of saving computational time and resources. T. Dharshini [9] worked on different techniques to detect deduplicate multimedia data such as image and video deduplication. The research work compared existing image and video deduplication based on techniques used and performance metrics to improve the efficiency and performance in image and video deduplication. Jonathan Takeshita [71] presented near-identical image deduplication using a single-server protocol.

Secure Locality Sensitive Hashing with a cryptographic hash function is used to hash the images. The protocol used is computationally feasible and secure against fully malicious and colluding adversaries. Sujatha G et al. [69] analyzed various image hashing techniques to find the unique images using collision resistance property. In this work, Image Deduplication in the cloud is discussed and image hashing methods are compared in terms of robustness and discrimination. Table 1 discusses the findings of existing exact image deduplication techniques and their key features.

2.3 Near-exact duplicate image detection techniques

Near-exact or near-duplicate images are the images having some modifications and editing such as cropping, scaling, rotation of an original image. The literature available on near-exact image detection mainly varies on feature vectors exploited for representing the image and the indexing technique used. K. Velmurugan et al. [77] exploited (Speeded-up Robust Features (SURF) algorithm to extract interest points of images, and KD (k-dimensional) tree is used to index and match similar features of the images. The voting scheme algorithm detects matched

Table 1 Exact image duplicate detection techniques

Reference	Detail	Techniques	Findings
[86]	Photo-based deduplication process based on family photographs to find duplicate electricity bills.	Block Truncation Coding, histogram refinement algorithm.	Eliminate nearly 0.35 million duplicate electricity cards. Better space and bandwidth utilization.
[61]	Proposed histogram refinement features using a K-means clustering algorithm to eliminate duplicate ration cards.	Histogram Refinement	Removes duplicate ration cards based on some threshold value. The deduplication process requires human intervention.
[8]	Constructed B+ tree index using Haar wavelet decomposition to extract the edge information. Manhattan distance detects the similarity between two images.	Haar wavelet	Higher deduplication rate but not scalable to large no. of images.
[41]	Proposed a secure perceptual similarity deduplication scheme (SPSD) using a Perceptual hash algorithm to generate image signatures.	SPSD, Hamming distance	Storage and bandwidth savings for a small-scale database.
[24]	Proposed a new hybrid compression strategy to compress the data at suitable dimensions.	Convergent Encryption, PCA, SPHIT and Compressive Sensing Techniques	Better efficiency in terms of encryption.
[9]	Studied different techniques to deduplicate multimedia data such as image and video deduplication.	Image and video deduplication techniques	Efficiency, performance and accuracy measures are taken for comparison
[71]	Presented a near-identical image deduplication using a single-server protocol.	Single server protocol, Secure Locality Sensitive Hashing	Efficient protocol and secure
[69]	Analysed various image hashing techniques to find unique images using collision resistance property.	Collision resistance property	Image hashing comparison based on robustness.

images from the database. Ke et al. [28] proposed a system to detect near-exact image duplicates. The sub-image retrieval is employed in the Difference of Gaussians (DoG) to detect interest points and PCA-SIFT for distinctive local descriptor representation. In this technique, Locality Sensitive Hashing (LSH) is used for indexing local descriptors and similarity searches for high dimensional data.

Yanqiang Lei et al. [37] proposed a cluster of uniform randomized trees for the fast detection of near-duplicate images. The feature vectors are projected into one-dimensional space, and these features are further implemented using a binary tree. It limits the search space, improves image detection efficiency and is faster than the LSH scheme. Xinghua Yu et al. [85] proposed a SIFT-based robust geometric transformation fingerprinting technique. SIFT keypoints and descriptors are extracted, and homology of two images is detected using area ratio invariance of the affine transformation. KD-trees are used to construct matched keypoints and the best bin algorithm is exploited to check the matching speed. This approach improves the performance to detect homology on the occurrence of geometric transformation. JinS. Seo et al. [62] proposed an image fingerprinting method using a Radon transform. The perceptual hashing is used to obtain bits of multimedia content and a hamming distance is used to detect the similarity of two images. The scheme is robust to affine transformation.

Li Li et al. [39] proposed an image matching algorithm using SURF feature points and daisy descriptor. The euclidean distance is used to match the reference image with the original image. The proposed technique improves matching accuracy, has better speed and robustness. At the same time, it is not suitable for large-scale image variation. Chun-Che Chen et al. [6] proposed a fast image retrieval scheme by transforming 128-DSIFT features into 128-bit binary representations. The hash values are calculated for each of these features, and hamming distance is used to identify similar images. The technique reduces search retrieval time. Jun Jie Foo et al. [14] introduced a similar image collator (SICO) for near-duplicate image detection and PCA-SIFT is used for interest point detectors. In this work, LSH is used to efficiently index the features, and similar images are detected by comparing the matched features. Li et al. [44] proposed a variable length signature for image matching. To match near-duplicates, image patches are generated using probabilistic centre-symmetric local binary pattern new visual descriptors. Earth mover's distance is used to identify similar images.

Yao et al. [84] introduced a new contextual descriptor to measure the contextual similarity of visual words to identify near-duplicates. S. Thaiyalnayaki et.al [72] proposed indexing near-duplicate images using SURF feature extraction. The image is enhanced using a query image followed by features extracted using SURF. The min-hash algorithm is used to measure similarity in images. Locality Sensitive Hashing (LSH) indexing is employed to detect near-duplicate images. The indexing technique used is efficient for a few hundred thousand images. Jingkuan Song et al. [66] proposed a novel self-supervised video hashing (SSVH) framework to capture the temporal nature of videos. The hierarchical binary auto-encoder architecture is designed to address the issues related to temporal dependencies in videos and generates binary codes of videos for accurate video retrieval. The experimental results show that the proposed framework outperforms on unsupervised video retrieval. Jingkuan Song et al. [65] introduced a multiple feature hashing(MFH) method to address the scalability and accuracy issues related to near-duplicate video retrieval (NDVR). The multiple feature hashing method learns a group of hash functions by preserving the local structural information of individual features. The binary codes are generated by mapping the video keyframes into a hamming distance. The proposed method outperforms in terms of accuracy and efficiency. Jingkuan Song et al. [67] proposed a unified binary generative adversarial network (BGAN+) framework that convert images into binary codes in unsupervised fashion for both image compression and retrieval. The proposed BGAN+ framework simultaneously learns two binary representations per image, one for image retrieval and one for image compression. The experimental results show that BGAN+ attains competitive performance for image compression and outperforms the existing retrieval methods with significant margins. Jingkuan Song [64] proposed an unsupervised hashing technique using generative adversarial network (GAN). A novel loss function is designed to retrieve the images accurately by generating the binary codes from the images. The experimental results show that the proposed approach outperforms as compared to existing hashing methods.

Dengzhi Liu et al. [45] proposed a secure real-time image protection system to detect exact and near-exact duplicate images stored in the Cloud. The convergent encryption technique encrypts the images. The image deduplication is executed before uploading the images to detect duplicate images. Deep learning is exploited to improve the efficiency of near-duplicate detection. The proposed scheme is more efficient in terms of performance for image storage, deduplication checking, and near-duplicate detection and security. S. Thaiyalnayaki et al. [73] proposed a technique to detect near-duplicate images using SURF and segmented minhash algorithm. The image features are extracted using a SURF algorithm and the segmented minhash algorithm is used to index the similarity of extracted images. Locality Sensitive Hashing is used to index the near-duplicate images. The proposed approach is highly effective for near-duplicate image detection. K. K. Thyagharajan et al. [74] reviews near exact image detection techniques that focus mainly on feature extraction techniques. The main challenges in near-duplicate image detection are addressed efficiently. Fang Huang et al. [20] proposed a new method to detect near-duplicate images by tracing the original image of each near-duplicate image cluster. Image clustering is done using both local and global features. The experimental results indicate the effectiveness of our method.

Diankun Zhang et al. [89] classified near-duplicate videos using temporal and spatial keypoints. The key frames are extracted from videos and video segments are recognized to identify the approximate videos or near-duplicate videos. The proposed method is quite effective to detect near-duplicate videos. Giorgos Kordopatis-Zilos et al. [29] proposed a method to detect near-duplicate video retrieval using an unsupervised scheme based on a modified Bag-of-Words (BoW) video representation and supervised method based on Deep Metric Learning (DML). The features are extracted using Convolutional Neural Networks and frames are generated for compact image representation. The proposed method shows better results. Siying Liang et al. [42] proposed a hierarchical detection method based on the Convolutional Neural Network (CNN) model. Semantic and label descriptors are obtained from deep semantic features of CNN extracted from video frames. Hierarchical matching detects near-duplicate videos and mean average precision is obtained on the dataset. Table 2 discusses a few findings of existing near-exact duplicate image detection techniques and their key features.

2.4 Real-time duplicate image detection technique

Yu Hua et.al [19] proposed Smart-Eye in-network coarse-grained deduplication in cloud-assisted disaster recovery system. In this work, DoG and PCA-SIFT are used for feature extraction, bloom-filter is used to hash features of an image and serves as input to LSH which maps the similar images in a DiffServ aggregated flow. The technique obtains energy savings

Table 2 Near-exact duplicate image detection techniques

Reference	Detail	Techniques	Findings
[77]	Utilized SURF algorithm to extract interest points of images and KD-tree with the best bin is used to index and match similar features of images. Voting Scheme algorithm retrieve matched images from the database	SURF and Kd-tree	Improves average precision to detect the matched images.
[28]	Detect near-duplicates and sub-image retrieval using DoG for interest point detector and PCA-SIFT for distinctive local descriptor representation. LSH is used for indexing local descriptor and an efficient similarity search for high dimensional data	DoG, PCA-SIFT and LSH	High computational overhead when the database grows.
[37]	Proposed a cluster of Uniform Randomized trees for fast detection of near-duplicate images and feature vectors are projected into one-dimensional space.	Kd trees	Improves search space and detection efficiency and is faster than LSH scheme
[85]	Proposed SIFT-based robust geometric transformation fingerprinting technique. SIFT keypoints and descriptors are extracted, and homology of two images is detected using area ratio invariance.	SIFT, KD-trees	Better performance to detect homology when the geometric transformation occurs. KD-trees are used to construct matched key points, and the best bin algorithm is used to check the matching speed.
[62]	Proposed image fingerprinting method using Radon transform and perceptual hashing is used to obtain bits of multimedia content. Hamming distance detects the similarity between two images.	Hamming Distance	Robust to affine transformation
[39]	Proposed Image matching algorithm using SURF feature points and Daisy descriptor to compute the principal direction of feature points. Euclidean distance matches the reference image with the original image.	SURF and DAISY	Improves matching accuracy, better speed and robustness. But is not good for large-scale image variation
[6]	Proposed a fast image retrieval scheme by transforming 128-DSIFT features into 128-bit binary representations. Hash values are calculated of each of the features, and Hamming distance is used to identify similar images.	SIFT	Reduces search retrieval time using binarized features.
[14]	Introduces SICO (Similar Image Collator) for near-duplicate image detection and PCA-SIFT used for interest point detectors. LSH is used to efficiently index the features, and similar images are detected by comparing the matched features.	PCA-SIFT	An effective and efficient method as LSH efficiently hashes the values.
[44]	Proposed variable-length signature for image matching. To match		Patch-based near-duplicate detection, flexible even in case of image

Table 2 (continued)

Reference	Detail	Techniques	Findings
	near-duplicates, image patches are generated using probabilistic centre-symmetric local binary pattern new visual descriptors.	Earth mover's distance on PCSLBP	distortions. Earth movers distance cope with variable length signatures and is used to identify similar images.
[84]	Introduced a new contextual descriptor to measure the contextual similarity of visual words to identify near-duplicates, efficiently encodes the neighbors' local descriptor.	SIFT descriptor	Efficient for large-scale near-duplicate image retrieval, takes less query time and robust image deformation.
[72]	Proposed Indexing of Near-Duplicate Images in the Web Search using SURF feature extraction. Min-hash algorithm is used for similarity measurement between images. Locality Sensitive Hashing (LSH) index near-duplicate images from query images.	SURF, Min-Hash	Efficient for the small dataset.
[66]	Proposed a novel self-supervised video hashing (SSVH**)** framework to capture the temporal nature of videos.	Hierarchical Binary Auto-Encoder model	Better performance for unsupervised video retrieval with lesser computations.
[65]	Introduced the Multiple Feature Hashing (MFH) method to address the accuracy and scalability issues related to Near-duplicate video retrieval (NDVR).	Hamming distance, Hashing technique	Better performance in terms of accuracy and efficiency.
[67]	Proposed a unified binary generative adversarial network (BGAN+) framework and convert images into binary codes in unsupervised fashion for both image compression and retrieval.	BGAN+ encoder-decoder	Better Performance for Image Compression
[64]	Proposed an unsupervised hashing technique using generative adversarial network (GAN). A novel loss function is designed to retrieve the images accurately by generating the binary codes from the images.	Loss function, binary codes and sign-activation function.	Proposed approach outperforms as compared to existing hashing methods.
[45]	Proposed a secure real-time image protection system to detect exact and near-exact duplicate images stored in the Cloud.	Convergent encryption, Deep Learning	Secure and efficient in terms of performance for near-duplicate detection
[73]	Proposed a technique to detect near-duplicate images using SURF and segmented minhash algorithm. Locality Sensitive Hashing (LSH) indexes the near-duplicate images.	SURF, Locality Sensitive Hashing	Highly effective for near-duplicate image detection.
[74]	The different feature extraction techniques to detect near-duplicate images are reviewed. Near-duplicate image detection are addressed efficiently.	Feature extraction techniques	Efficiently address the challenges related to near-duplicate images.
[20]	Proposed a new method to detect near-duplicate images. Image	Image relational graph,	An effective and efficient method to detect near-duplicate images

Table 2 (continued)

Reference	Detail	Techniques	Findings
	clustering is done using both local and global features.	PageRank algorithm	
[89]	Proposed near-duplicate videos using temporal and spatial keypoints.	Temporal and spatial keypoints	An effective method to detect near-duplicate videos.
[29]	Proposed a method to detect near-duplicate video retrieval using an unsupervised scheme based on a modified Bag-of-Words (BoW) video representation and supervised method based on Deep Metric Learning (DML).	Convolutional Neural Network	Efficient Method To Detect Near-Duplicate Videos
[42]	Proposed a hierarchical detection method based on the CNN model.	Hierarchical matching technique	Mean average precision obtained

and improves bandwidth efficiency [86]. [53, 93] also reports on real-time image detection schemes. Fudong Nian et al. [53] proposed online near-duplicate detection based on local-based binary representation (LBR). The binary representation uses a local and global representation of an image to detect near-duplicates in real-time. Pengfei Zuo et al. [93] proposed bandwidth and energy-efficient image sharing system (BEES) in case of disasters. Approximate Image Sharing (AIS) explores feature extraction and redundancy detection in real-time to reduce the consumption of bandwidth and energy efficiency. Table 3 discussed a few findings of real-time duplicate image detection techniques and their key features to detect exact or near-exact images in a storage system.

The techniques discussed above have shown their own unique features to detect exact or near-exact images in a storage system. On the other hand, the scalability and performance are a major concern for techniques to detect the exact and near-exact images. The efficiency of exact or near-exact image detection decreases with the increase of image quantity and size. There must be an efficient technique to detect exact or near-exact images with a large database of images that extract most pertinent features to get more accurate and efficient results. The paper

Table 3 Real-time duplicate image detection technique

Reference	Detail	Techniques	Findings
[19]	Proposed Smart-Eye, in-network coarse-grained deduplication in cloud-assisted disaster recovery system. DoG, PCA-SIFT are used for feature extraction and Bloom-filter for hashing the features.	SIFT and DoG	Obtain energy savings and improve bandwidth efficiency.
[93]	Proposed bandwidth- and energy- efficient image sharing system (BEES) in case of disasters. Approximate Image Sharing (AIS) explore feature extraction and redundancy detection in real-time.	BEES, AIS	Reduce the consumption of bandwidth and energy efficiency
[53]	Proposed online near-duplicate detection based on local-based binary representation (LBR). The binary representation uses a local and global representation of an image to detect near-duplicates in real-time.	LBR	The local and global binary representation used for near-duplicate detection

proposed feature extraction techniques using HDV and CNN to detect exact or near-exact images for online deduplication.

3 Exact and near-exact image detection

Image deduplication techniques are based on image detection techniques that are further classified as exact image detection and near-exact image detection. Image features extraction is the most important and crucial step in image deduplication. The major challenge of image feature extraction and detection of exact and near-exact images in a cloud storage system is time and accuracy. The basic feature extraction techniques such as SURF [4, 25, 55], SIFT [17, 22, 27], BRISK [38], features from Accelerated Segment Test (FAST) [5, 49], Haar [7, 92], Harris [18, 46], DWT [88] or hybrid techniques such as binarized SIFT [58], LBP [3], Local-based Binary Representation (LBR) [53], Probabilistic Center-symmetric Local Binary Pattern (PCSLBP) [44], DoG and PCA-SIFT [28, 90] and DSIFT [80] can be replaced with state-of-the-art CNN based feature extraction techniques for improving accuracy. CNN based image detection techniques are evolving. The CNN based feature extraction techniques [21, 60, 75, 76, 83, 87] have been reported for object detection [16], face detection [10] and plant identification [36]. In our proposed technique, we also used deep CNN based feature extraction from a fully connected neural network to extract image features and kept it as hash values.

3.1 Proposed online image deduplication technique

In image deduplication technique, the query image features are extracted using different image feature extraction techniques and its fingerprints or signatures are generated. Deduplication algorithms focus on these image signatures for further processing. Newly generated fingerprints are compared with the earlier stored fingerprints in the signature database. If it already exists, then a pointer would be created to reference the image with an already stored image in the storage database. Else, the fingerprint will be considered as a new and both the fingerprint and the image will be stored in the storage database.

3.1.1 Architecture of proposed online image deduplication technique

Our proposed technique has three components: client-side pre-processing of the query image, intermediate computation node for signature matching of images with image signature database of existing hash tables and distributed image storage. In proposed technique, the query image features are extracted on client-side and its signatures are matched in the image database stored on an intermediate node. The server-side stores the database of original image. The architecture components of the proposed technique are presented below.

- Client-Side Pre-Processing: The client is the source of an image or new query image that needs to be analysed for an exact or near-exact image match. This image is queried for a match on the signature database of an intermediate node. In order to achieve this, features are extracted from a query image using image feature extraction techniques. In our proposed technique, CNN based feature extraction technique is used to detect exact or near-exact image detection for online deduplication as further described in section 3.2.1. The extracted features of a query image are forwarded to the intermediate computation nodes for further processing. The extracted features sent to an intermediate node are processed for a duplicacy test to check whether these are unique features of an image or the features of this image are already stored in the database. Here, the client-side is also responsible for making a decision to accept or discard the near-exact version of a query image. The exact match of a query image with the image database results in duplicate images and is discarded. For a near-exact image match, the user is queried for taking a final decision to store or discard the near-exact image. The image patches are generated using our proposed HDV technique discussed in section 3.3 for a near-exact version of an image and are stored in the database. The image patches contain information regarding color synchronizations, affine transformations and cropping.

- Intermediate Computation: The intermediate node is responsible to maintain the image signatures or image hash values to process the image matching. On intermediate computation nodes, the features extracted from query image are looked up in the signature database of images. The query image is a new image when there is no matching of query image with the signature database. The signature of this new image is added to the signature database and the image is directly stored on the distributed storage on the server-side. The exact matching of query image results in duplicate image detection and a pointer of an existing duplicate image is returned. Initially the database is empty. Every time a new image enters the system, its features are extracted using our trained CNN. These features are provided to the LSH hash table for indexing.

- Distributed Image Storage: Distributed Image storage has distributed nodes on the storage system that are used to store the complete images. The original image is stored on distributed storage nodes. Any match of the near-exact image is presented to the end-user on the client-side for approval to retain the original image or its near-exact match.

The architecture of proposed Deep CNN based exact or near-exact image detection for online deduplication is demonstrated in Fig. 2 The image patches are generated using our proposed

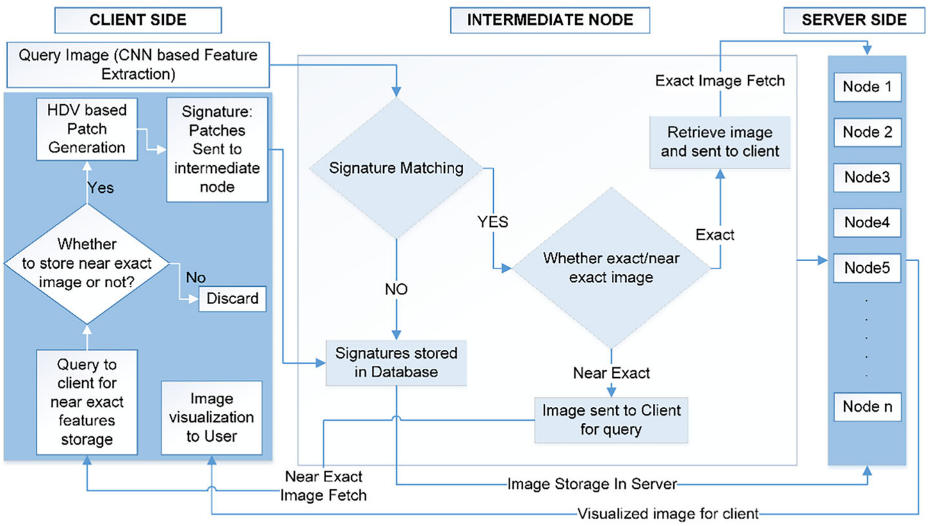


Fig. 2 Proposed online image deduplication technique

HDV technique for a near-exact version of an image and are stored in the database. It contains information regarding color synchronizations, affine transformations and cropping.

The Deep-CNN based feature extraction for image matching and retrieval is discussed in section 3.2 while patch generation for near-exact images is discussed in section 3.3.

3.2 DEEP CNN for exact or near-exact images deduplication

The convolution neural network [56, 63, 68] has attained high success in the large-scale image and video recognition. A deep neural network is divided into feature extraction and classification layers. The feature extraction layers of a CNN consist of convolution, activation and pooling layers whereas the classification layers use FC/ACTV layers. Deep learning neural networks use feature extraction layers to learn a deep hierarchy of features. The deep architecture in CNN makes it more powerful for image recognition. These features termed as deep CNN (DCNN) is translational, rotational and illumination invariant. This has achieved much success in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). There are several models for deep network techniques, such as the AlexNet model [56], VGGNet [63] and GoogleNet [70].

- Convolutional layers: These layers contain significant features in input data, apply a convolution operation to the input and pass the result to the next layer.
- Activation layers: It is an element-wise function to make the network more powerful and is applied to the input feature maps. A most widely used activation function is the Rectified Linear Unit (ReLU) function. It is implemented in hidden layers of the Neural network and the purpose of ReLU [43] function is to increase the non-linearity in the images.
- Pooling layers: The pooling layers use average or max operation on the input data which results in reducing the spatial size. It reduces the number of parameters and computation in the network by reducing the dimensions. The max pooling is the most popular pooling which takes maximum value in each window or from each of a cluster of neurons at the prior layer. It decreases the feature map size and extracts important features of an image.
- Fully-connected layers: Also known as the classifier layer is the last layer of a CNN. The main function of this layer is to take neurons from the previous layer so that every neuron in one layer is connected to every neuron on another layer. FC/ACTV [35] activation function transformed the summed weighted input from the node into the output of that input.

3.2.1 DEEP CNN for exact or near-exact image detection

Figure 3 presents the architecture of a fine-tuned AlexNet for exact or near-exact image detection for online deduplication. The input of the CNN network is a 256*256 image size which is convolved by a series of convolutional layers. We evaluated AlexNet and VGGNet architecture trained on the ILSVRC-2012 dataset for feature extraction where features are extracted from the second-last layer of both the nets giving us a vector of 4096 for AlexNet and VGGNet respectively. We further used pre-trained AlexNet and fine-tuned it using Pascal VOC 2007. Here, we modified the fc7 layer and added 2 more fc layers of sizes 1024 and 128 as shown in Fig. 3.

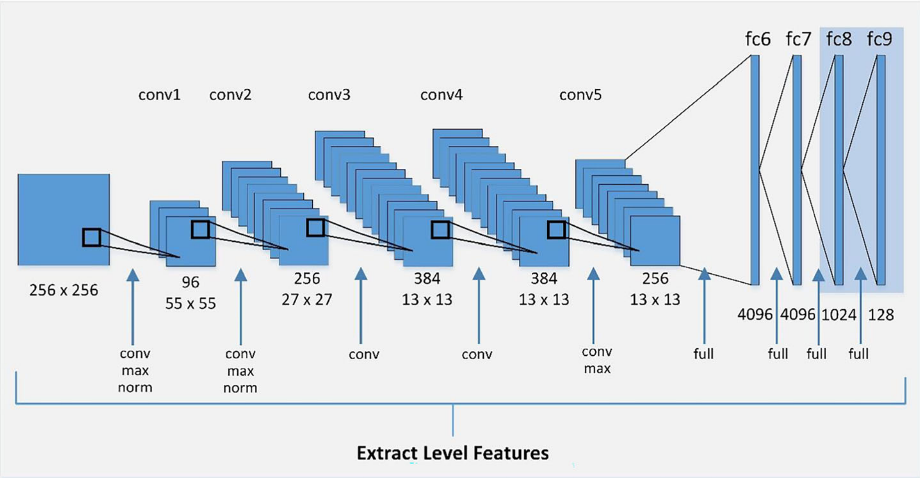


Fig. 3 Architecture of fine-tuned AlexNet based exact or near-exact image detection for online deduplication

Our proposed improved CNN reduces the feature dimension to 1024 and 128 respectively. The output of the fine-tuned network is a 128-dimensional vector for a single input image. This paper denotes it by *Net-0* henceforth. This facilitates a low memory and time requirement for feature matching. We later demonstrate the time variations in indexing depending on the network chosen and length of feature vector extracted. These networks are referred to as *Net-1, Net-2* etc. and are described later in the paper.

It should be noted that although CNNs provide a superior performance compared to handcrafted features, the training data plays an important role. Most applications use a domain specific fine-tuning to obtain superior results for their task at hand. Most of these works start by taking pre-trained features obtained using ImageNet dataset from AlexNet, VGGNet and Resnet. A transfer learning or fine-tuning is next applied. There are hence two points to deal with (a) the network to be chosen (b) the layer from which pre-trained features are to be extracted. We perform all further experiments with the AlexNet keeping in mind the time constraint for our online search. As we will move to deeper networks like VGG or Resnet, we will increase memory and computational requirements for our system. AlexNet proves to be an optimal choice between Resnet and low-version MobileNet. The second issue to be considered is the layer to be selected for feature extraction. It is a well-known fact that deeper layers generate high-level features whereas lower layers provide low-level features. Our task at hand involves a data deduplication without the knowledge of the data domain. For example, the data can belong to satellite images (PatternNet), handwritten characters (OMNI) or 3D model image data (ModelNet40). All these are very different from natural scenes in ImageNet or CIFAR100. We hence aim at proposing a network, which can not only deal with cross domain images but also with perturbations in each of them.

We introduced a network architecture of cross-domain images with perturbations as shown in Fig. 4. Output from the fifth convolution layer after maxpooling is obtained using pre-trained AlexNet. This is then shared with five different modules which learns weights individually to generate a 2048 sized feature vector. This vector is then used for generating hash tables using LSH. One of the five modules first learns to recognize the image type, following which the corresponding feature vector is generated for that image type. In this work, we consider four types of images. As mentioned before, we take satellite images (PatternNet), handwritten

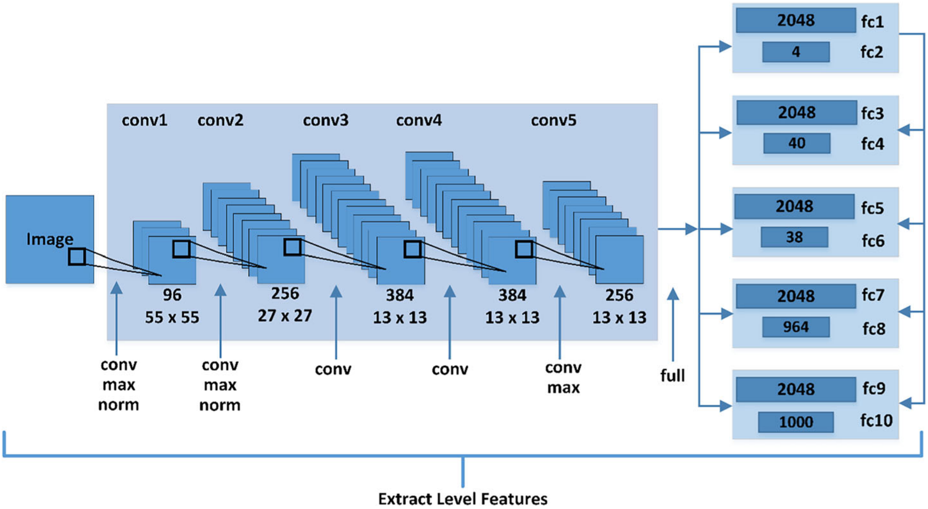


Fig. 4 Architecture of cross-domain images with perturbations

characters (OMNI), 3D model image data (ModelNet40) and natural images with indoor and outdoor scenes. Extensive experiments using these are later provided in the results section.

3.3 Patch generation for near-exact images using hot decomposition vector

An important decision on obtaining a near-exact match involves deciding whether to discard the image or store it. In this research, we propose a mechanism where we can opt to store the near-exact image not to its entirety but as patches, which differ from the original. Image patch generation to store dissimilar parts of near- exact images can further storage the storage requirement. In such cases instead of storing two images where one is 45 degrees rotated with respect to the first, we store only a single image and information required in order to reconstruct it. We use the proposed HDV technique to compare these images and generate the necessary information.

The Hot Decomposition Feature Vector (HDV) is an application of orthogonality on DCT features and SURF on wavelet coefficients. For an input image I(x, y), n SURF feature points bw are detected on the wavelet transformed image as shown in Eq. 1

$$D = \left(bw_i{}^1 \ldots, bw_i{}^2 \ldots\ldots\ldots\ldots bw_i{}^{128}\right) \; (i = 1, .., n) \tag{1}$$

This is referred to as Wavelet SURF Transformation (WST) henceforth. The image is divided into non-overlapping blocks. Each DCT block D has a size $d_r \times d_c$. Hence x blocks are composed of $D\epsilon R^{x \times d_r \times d_c}$. The data is represented as a 3rd order tensor, and a hot decomposition is carried out using a 2nd order mode matrix W_r and third-order mode matrix W_c. W_r is the DCT of the mode 2 flattened matrix of A defined as

$$A = D \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T \tag{2}$$

W_r and W_c are optimized using

$$\| W_r^{k+1\top} W_r^k \| > (1-\sigma) d_r \tag{3}$$

$$\| W_c^{k+1\top} W_c^k \| > (1-\sigma) d_c \tag{4}$$

Updated matrix W_r^{k+1} and W_c^{k+1} can be obtained from DCT on $A_{\times 3} W_c^{k\top}$ and mode 2 flattened and mode 3 flattened $A_{\times 2} W_r^{k\top}$ respectively. The final vector is represented as $[A_{\times 2} W_r^{\top} \times_3 W_c^{\top}, D]$.

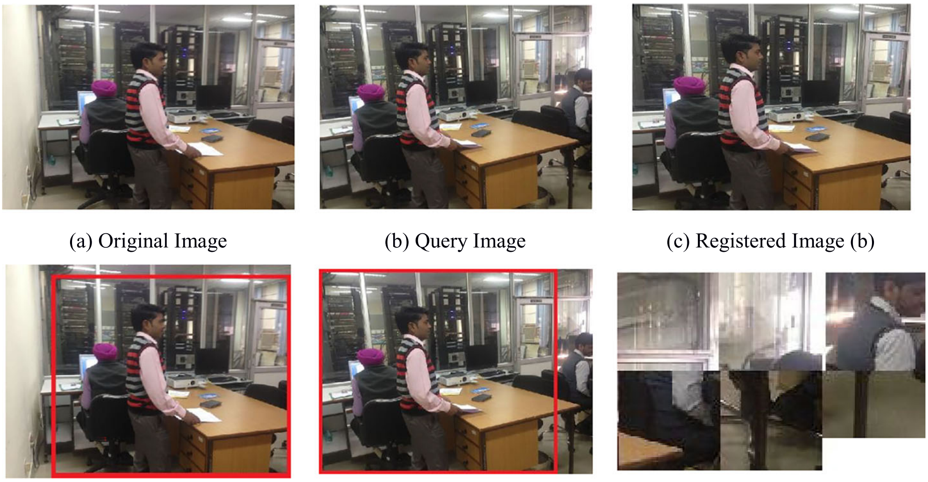
3.3.1 Near-exact image patch generation

For a near-exact image match, the user is queried for taking a final decision to store or discard the near-exact image. The proposed patch-based near-exact storage scheme using HDV features can be used to reconstruct images in the future. The differences of query image with the base or matched image may be stored in a B+ tree. The value of each key consists of affine, color and cropping information. This information may be used for reconstruction of the image. HDV features of base and query image is used to obtain image alignment (affine) information. HDV features are used to generate image correlation and patches with poor correlation factor are treated as extra patches. The color information stored is generated using the DC component of the Fourier transform of the query image. This stores the average color component and can be used to restore the color. All these techniques are considered individually. For example, each pair of images are either checked for color component or cropping or affine transformation. Also, a single set of transformation is considered currently. The hash table stores the key of each image which refers to the B+ tree. The leaf stores data $\tau_{B - \rightarrow Q}$, $dp^c(x_i, y_i)$ (for i = 1 to 4) required for near-exact reconstruction. $\tau_{B - \rightarrow Q}$ gives the transformation between the base and the query image. $dp^c(x_i, y_i)$ denotes the corner points of the base image matched with the query image. $dp^c(x_i, y_i)$ is computed from $Q_{\tau B - \rightarrow Q}$. Image patch p^{dp} obtained from $dp^c(x_i, y_i)$ is transformed as $p^{dp}_{\tau B \longrightarrow Q}$ and mapped with Q to obtain p^Q. p^Q is represented as $(x_k, y_k)\, p_k^Q$Q having the general form $[\,(x_1, y_1)\, p_1^Q,\ (x_2, y_2)\, p_2^Q,\ (x_3, y_3)\, p_3^Q \ldots]$. (x_k, y_k) denotes the starting position of a sub-patch and p_k is the sub-patch itself. Sub-patches can be square or rectangular depending on the size of the query image and the patch $p^{dp}_{\tau B - \rightarrow Q}$.

Figure 5 shows generation of image patches. Figure 5a is taken as the original image and Fig. 5b is a near-exact image of Fig. 5a. Figure 5b is the query image and has been registered as shown in Fig. 5c. Figure 5d demonstrates the correlation of Fig. 5a original image and Fig. 5c registered image. Figure 5e represents the correlation of image in Fig. 5b with Fig. 5c. Finally, Fig. 5f shows the patches are generated from Fig. 5d and 5e that reflect the dissimilar part of the image. Figure 6 shows another example of an original image, its near-exact images, image patches are generated for dissimilar part and transformation matrix is used for reconstruction is stored as follows $\tau_{a - \rightarrow b}$, (0.97870.02470, −0.02470.97870, −39.379313.57521.0000) $\tau_{a - \rightarrow c}$, (1.0032 − 0.00830, 0.00831.00320, −83.66525.61711.0000)2$dp^c(53_1, 1_1)$ $(359_2, 1_2)$ $(53_3, 478_3)$ $(359_4, 478_4)$, 3$dp^c(88_1, 1_1)$ $(359_2, 1_2)$ $(88_3, 478_3)$ $(359_4, 478_4)$.

4 Experiments and results

The experimental work of this research has been implemented in Matlab, Torch and PyTorch. To evaluate the behavior of different image detection algorithms for deduplication, we considered PASCAL VOC dataset and our dataset of 70,000 images. Our collected dataset



(a) Original Image (b) Query Image (c) Registered Image (b)

(d) Correlation of (a) with (c) (e) Correlation of (b) with (d) (f) Image patches Generated

Fig. 5 Near-exact Image detection for image deduplication and generation of patches for dissimilar part

(a) Base Image (b) Nearly similar image to base image (c) Nearly similar image to base image

(d) Image patches formed from image(b) (e) Image patches formed from image(c)

Fig. 6 Near-exact Image detection for image deduplication and generation of patches for dissimilar part consists of indoor and outdoor images taken on the University campus. These include scenes like classrooms, laboratories, car parking, stairs, cafe, one or more people etc. Also, these were captured on different days and time to accommodate the same scene with scale and illumination changes. The dataset images have various augmentations such as scale, rotation, and illumination. The subsets from DS1 to DS2 are random groups of these images with varied scale, illumination and capture angle. The purpose of this grouping is to test the robustness of our algorithm against each type of variation. To reduce the biasness, the experiments are modeled and executed on the same machine and the same dataset. The dataset is further divided into ten subsets and referred to as DS1 to DS10 with different image variants and their combinations for comparison. All the experiments are executed on a workstation with Intel i5 2.53 GHz processor, 12.0 GB RAM and Linux operating system. The experimental results of image matching techniques are evaluated in terms of time and accuracy and are compared. The experiment takes input image parameters such as the size of an image (256*256, 240*240, 220*220) and image file types (JPG). The image sizes vary based on the network used. In this paper, the experimental results of techniques are repeated thrice, and averages of all parameters are used for various comparisons.

The article presents experimental results of the following:

(i) Concurrent handcrafted feature extraction techniques are implemented with our dataset. The results in terms of accuracy and time are presented and compared in Figs. 7 and 8.
(ii) Performance comparison of individual key point feature descriptors is mentioned in Figs. 9, 10, 11 and 12. HDV image matching technique as a combination of key-point features with DWT and DCT-block is also presented. The choice of SURF key-point descriptors is validated.
(iii) Fine-tuned AlexNet is implemented, and results of this proposed network are compared with existing CNN based AlexNet and VGGNet techniques to detect exact or near-exact images. The performance of CNN based image detection with a dataset of scaled images, illumination images and rotated images are presented in Figs. 13 and 14.
(iv) The classification is based on image clusters. Image classifiers are evaluated based on different image variations to detect exact or near-exact images, and findings are discussed in Tables 4, 5 and 6.
(v) Our proposed network trained with cross-domain images and perturbations is tested in terms of hash computation time, image query time and hits obtained. Results are presented in Tables 7, 8, 9, 10, 11, 12 and 13.

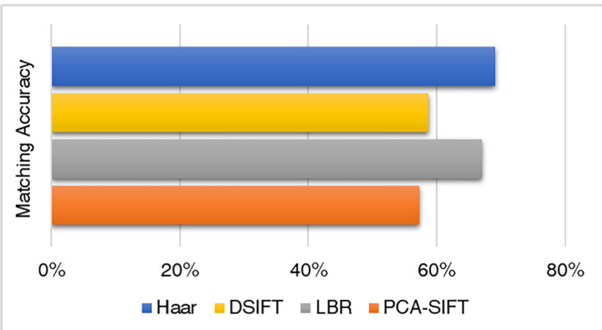


Fig. 7 Average accuracy of concurrent exact or near-exact techniques

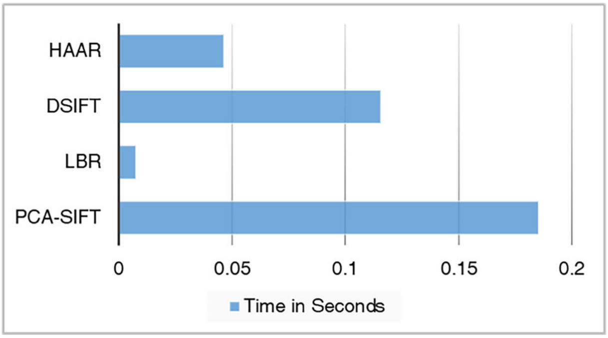


Fig. 8 Time in seconds of some concurrent exact or near-exact techniques

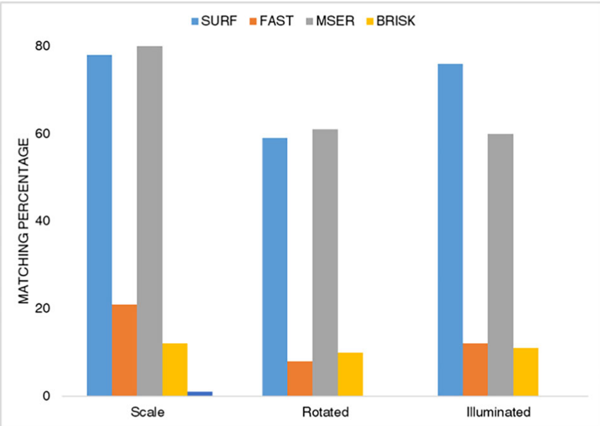


Fig. 9 Image matching accuracy of individual key-point feature descriptors on image deformation

4.1 Comparison of concurrent feature extraction techniques

Using our dataset, Figs. 7 and 8 reports the performance of concurrent exact or near-exact techniques [19, 53]. The analytical comparison of these techniques is based on image matching

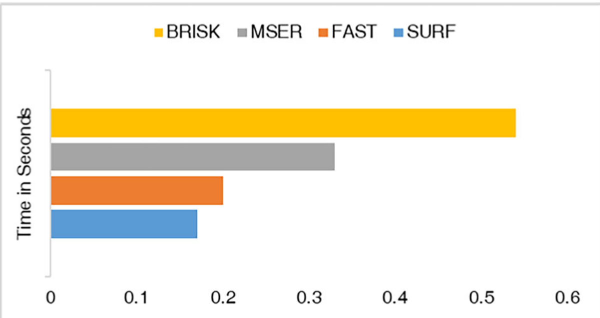


Fig. 10 Time in seconds of individual key-point feature descriptors on image deformation

Fig. 11 Matching accuracy of feature extraction algorithms with DWT on image deformation

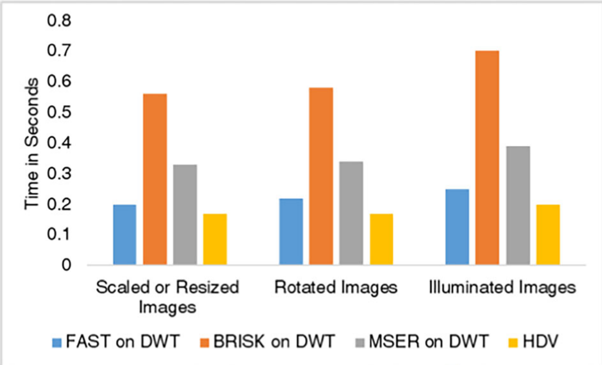


Fig. 12 Time in seconds based on feature extraction algorithms with DWT on image deformation

accuracy and time in seconds. It was observed that LBR and Haar have better image matching accuracy as compared to PCA-SIFT, DSIFT for a mixture of scale, illumination and pose variant images. The time taken by LBR and Haar is also comparatively less than other image matching techniques. DSIFT and PCA-SIFT achieve almost the same matching accuracy but DSIFT takes comparatively less time than PCA-SIFT.

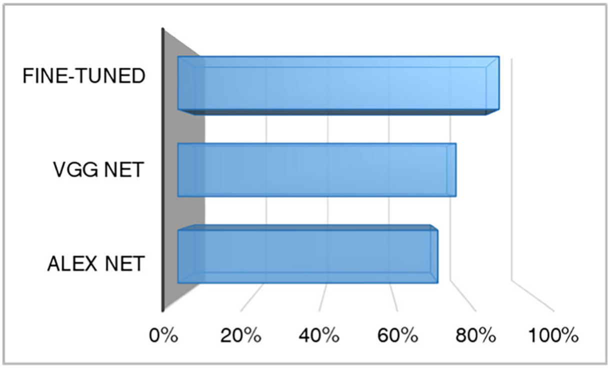


Fig. 13 Matching accuracy of CNN features extractors

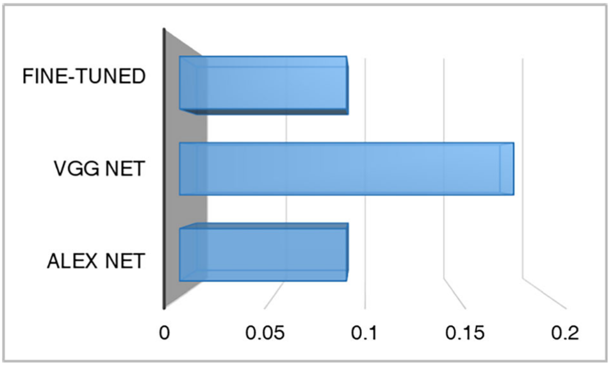


Fig. 14 Performance of CNN features extractors in seconds

4.2 HDV feature performance

Figures 9 and 10 depict the performance of individual key-point feature descriptors particularly on each type of variation based on matching accuracy and time in seconds. As shown in Fig. 8, SURF and MSER performed well on all the three image deformations scaled, rotated and illuminated images applied on the dataset. It was observed that SURF not only performed better among the other key-point feature extractors but also took the least computation time. MSER performed a little better in scaled and rotated images in terms of image matching accuracy but takes relatively higher computation time than SURF. BRISK takes the maximum computation time with the least image matching accuracy. FAST and BRISK both give a low number of keypoints. For example, a subset of 20,000 images showed less than 10 points in

Table 4 Top-1 and Top-3 recognition on the datasets

Dataset	Size	Image Variations	Top-1 Recognition(TN)	Top-3 Recognition(TN)
DS1	850	Illumination	28%	22%
DS2	3500	Illumination + Scale	11%	10%
DS3	6500	Illumination + Scale	3%	1%
DS4	4000	Illumination + Scale	8%	8%
DS5	1000	Pose	15%	4%
DS6	3400	Pose + Scale	3%	0.5%
DS7	1600	Pose + Scale	3%	2%
DS8	4400	Pose + Scale	5%	4%
DS9	8500	Illumination + Pose + Scale	11%	8%
DS10	1400	Illumination + Pose + Scale + Outliers	20%	16%

Table 5 Image detection performance of Distance Classifiers

Classifiers	DS1	DS5	DS10
Class Mean Euclidean	78	87	82
Euclidean	72	85	80
Bayesian	93	94	74

Table 6 Number of true-negatives for Bayesian classifier with a different sample size

Dataset	30(TN)	60(TN)
DS1	28%	26%
DS2	11%	11%
DS3	0.03%	0.02%
DS4	0.078%	0.067%
DS5	0.19%	0.16%
DS6	0.034%	0.012%
DS7	0.09%	0.03%
DS8	0.04%	0.04%
DS9	0.12%	0.11%
DS10	0.20%	0.17%

Table 7 Feature computation time in milliseconds (ms) for VGG-Flower for Net-1, Net-2

Network	Net-1 (ms)	Net-2 (ms)
Feature Extraction Time	7	8

Table 8 Hash computation time in milliseconds (ms) using different bits and hash tables for VGG-Flower with per image time in ms. Each entry represents values for (Net-1, Net-2)

Hash Length (Bits)	Hash-Table 2 (Net-1, Net-2)	Hash-Table 4 (Net-1, Net-2)	Hash-Table 8 (Net-1, Net-2)	Hash-Table 16 (Net-1, Net-2)	Hash-Table 32 (Net-1, Net-2)
16	(127.8, 150)	(125.4, 151.6)	(126.2, 150.8)	(130.2, 153.9)	(133.3, 155.5)
24	(141.3, 158.7)	(140.5, 159.5)	(138.9, 164.3)	(145.2, 167.5)	(149.2, 176.2)
32	(159.5, 176.9)	(157.9, 180.2)	(157.9, 179.4)	(161.9, 185.7)	(173, 195.2)
64	(196.0, 217.5)	(196.8, 218.3)	(203.97, 223.8)	(211.1, 235.7)	(229.4, 255.6)
128	(199.2, 217.9)	(202.4, 223.8)	(210.3, 233.3)	(226.9, 254.8)	(262.7, 289.1)

Table 9 Query time in milliseconds (ms) for VGG-Flower with per image time (Net-1, Net-2)

Hash Length (Bits)	Hash-Table 2 (Net-1, Net-2)	Hash-Table 4 (Net-1, Net-2)	Hash-Table 8 (Net-1, Net-2)	Hash-Table 16 (Net-1, Net-2)	Hash-Table 32 (Net-1, Net-2)
16	(60, 240)	(28, 87)	(480,2200)	(1210, 3490)	(2200, 5180)
24	(30, 30)	(30, 40)	(40, 80)	(50, 190)	(140, 280)
32	(30, 30)	(30, 40)	(40, 50)	(50, 60)	(80, 100)
64	(60, 60)	(70, 70)	(100, 120)	(130, 150)	(190, 210)
128	(60, 60)	(70, 80)	(100,110)	(130, 140)	(210, 230)

Table 10 Eight different perturbations for cross-domain dataset

p=0	Original
p=1	Gaussian-noise
p=2	Poisson-noise
p=3	Compression
p=4	Blur
p=5	Sharpen
p=6	Gamma
p=7	Adversarial
p=8	Solarize

Table 11 Classification performance metrics using Omni dataset

	Precision	Recall	F-Score
Omni w/o_P	0.46	0.54	0.48
Omni w_P	0.76	0.82	0.78

Table 12 Average and minimum hash computation per image time in milliseconds (ms) using cross-domain net

Hash Length (Bits)	Hash-Table 2	Hash-Table 4	Hash-Table 8	Hash-Table 16	Hash-Table 32
16	Avg=18.9	Avg=19.1	Avg=15.4	Avg=21.2	Avg=22.8
	Min=14.7	Min=14.7	Min=15.1	Min=15.9	Min=16.7
24	Avg=19.8	Avg=19.8	Avg=22.4	Avg=23.1	Avg=24.6
	Min=16.3	Min=17	Min=18	Min=19	Min=23.7
32	Avg=22	Avg=22.6	Avg=25.6	Avg=26.9	Avg=29.5
	Min=17.5	Min=19.8	Min=19.8	Min=19.8	Min=27.8
64	Avg=37.8	Avg=39.4	Avg=45.1	Avg=48.3	Avg=53.8
	Min=31	Min=32	Min=34.1	Min=34.9	Min=49.2
128	Avg=38.5	Avg=39.4	Avg=47.6	Avg=50.1	Avg=60.6
	Min=31.3	Min=31	Min=35.7	Min=38.9	Min=56.4

Table 13 Average and minimum query per image time in milliseconds (ms) using cross-domain net

Hash Length (Bits)	Hash-Table 2	Hash-Table 4	Hash-Table 8	Hash-Table 16	Hash-Table 32
16	Avg=216.8	Avg=451.7	Avg=776.1	Avg=2133.4	Avg=2897.2
	Min=40	Min=90	Min=170	Min=420	Min=1040
24	Avg=19.8	Avg=20.6	Avg=59.3	Avg=87	Avg=200.9
	Min=10	Min=20	Min=20	Min=30	Min=80
32	Avg=5.5	Avg=7.9	Avg=11.1	Avg=20.6	Avg=30.9
	Min=10	Min=4	Min=4	Min=20	Min=15
64	Avg=18.2	Avg=26.1	Avg=28.5	Avg=34.8	Avg=49.1
	Min=9.3	Min=18.7	Min=30	Min=24.8	Min=30
128	Avg=22.2	Avg=21.4	Avg=30.9	Avg=45.9	Avg=77.5
	Min=14.5	Min=14.5	Min=20	Min=35	Min=68

80% images for FAST features and 70% images for BRISK features. Therefore, overall SURF performs better than all other feature extraction techniques based on computation time and image matching accuracy in illuminated, scaled and rotated images for individual keypoint feature extraction.

Based on the results shown in Figs. 9 and 10, the proposed HDV image matching technique utilizes SURF keypoints and DCT-block. The HDV is based on an application of orthogonality on DCT features and SURF on wavelet coefficients. Our proposed HDV has performed better in terms of image matching accuracy in all three image deformations and takes relatively less computation time.

For empirical comparison of HDV with other features extraction techniques on DWT, the combination of existing techniques on DWT are implemented and the results of HDV and other existing feature extractors on DWT are shown in Figs. 11 and 12. The results demonstrate that HDV exhibits higher and stable image matching accuracy in all three types of image deformations with relatively least computation time. FAST, Harris, MSER and BRISK on DWT with a single approximate coefficient is applied on all three deformations on the same image database. MSER on DWT is second next to HDV in terms of image matching accuracy with demonstrated higher computation time than HDV. MSER on DWT performed well on scaled image matching accuracy in comparison to its image matching accuracy in rotated and illuminated image deformations. BRISK on DWT demonstrated no matching accuracy in scaled and rotated images and very small matching accuracy in illumined images. The computation time of BRISK on DWT is relatively much higher in all three image deformations. FAST on DWT exhibited relatively less matching performance than HDV, MSER on DWT. In addition, FAST on DWT shows inconsistent image matching accuracy on scaled, rotated and illuminated images deformations.

Overall HDV exhibits improved performance on all three types of image deformations with better matching accuracy and less computation time as compared to other four feature extraction techniques as shown in Figs. 11 and 12.

4.3 Comparison of deep CNN based feature extraction

As discussed earlier in section 3.2, Deep CNN for exact or near-exact images detection technique has been proposed which is a fine-tuned AlexNet (Net-0). The performance of proposed fine-tuned AlexNet is compared with existing AlexNet and VGGNet. In fine-tuned AlexNet, the original features extracted were 4096 which were later reduced to 1024 and finally to 128 features. The purpose of reducing the features to 128 feature vectors is to reduce the memory requirement to make this technique highly scalable for a large-scale image storage system. VGGNet takes large network space and time as compared to AlexNet which is simple and scales well for online image matching.

The empirical comparison is based on image matching accuracy and computation time in seconds. The results of these techniques are depicted in Figs. 13 and 14. These techniques are applied to the same dataset, which has all three types of deformations scaled, rotated and illuminated images. As shown in Figs. 13 and 14, the proposed fine-tuned technique has exhibited better image matching accuracy than existing VGGNet and AlexNet. VGGNet has shown better matching accuracy as compared to AlexNet with higher computation time than fine-tuned and AlexNet.

4.4 Performance comparison of image classifiers

Image classifiers are evaluated based on different image variations to detect exact or near-exact images and findings are discussed in Tables 4, 5 and 6. To evaluate the performance of CNN, ten different image subsets with different image size, variants and its combinations are used. The parameters to measure the performance of CNN is based on image size, image variations and true negative (TN) rate of Top 1 and Top 3 recognition.

The data set of the image was compared with different subsets DS1-DS10 with different image variants. Table 4 demonstrates the performance of CNN features extracted from fine-tuned net particularly for scaled, illumination and rotated image variants. The detected images are accepted or discarded by the client as discussed in Fig. 2. Euclidean distance feature classifiers are used to extract the matching of Top-1 and Top-3 image recognition. The True-Negative (TN) in Top-1 and Top-3 recognition provide a percentage of true negatives in different subsets of image variations. This True-Negative percentage value varies for different image variations in the subsets.

In Table 5, three different types of datasets are chosen randomly, such as DS1, DS5 and DS10 with sample size 850, 1000 and 1400 image and image variations to compare the image detection performance of distance-based classifiers. Table 5 depicts the image detection performance of Class Mean Euclidean, Euclidean and Bayesian image classifiers. It has been observed that the Bayesian image classifier performs better in illumination, pose image variations and exact images. Therefore, the Bayesian image classifier performs better to find near-exact images.

There is another important issue to be considered for the selection of classifiers. Class Mean Euclidean and Bayesian classifiers require mean and standard deviation respectively to be stored along with the signatures. Euclidean distance has no such requirement. Table 6 depicts True-Negative (TN) variations when the mean and standard deviation is computed with two different sample sizes 30 and 60 for each class.

When there is no match, and the image signature is to be stored for the first time then the client will provide the mean and standard deviation of the query images using different pose augmentations of the image. This means the user will provide the mean and standard deviation by using illumination, scaled and rotation image variants of a query image.

As results are shown in Table 6, DS2 and DS8 give the same results of true negatives in 30 and 60 sample sizes. However, in some cases like in DS6 and DS7, 60 sample size has less error rate as compared to 30 sample size. In Tables 4, 5 and 6 ten different subsets are taken. Clustering is done on our 70,000 image dataset. The main aim is to map the archive images into the cluster such that images of the same class are nearby to each other regardless of irrelevant characteristics and variations. To measure the similarity, the pixels in the feature space are clustered together. Here Class Mean Euclidean distance classifier, Euclidean Distance classifier and Bayesian Classifier are taken to measure the similarity.

4.5 Performance evaluation of cross-domain net

In this section, we make use of different Datasets such as VGG-Flower, Aircraft, Omni, CIFAR100 along with the PatternNet, ModelNet40 datasets. VGG-Flower has 1362 images [54], Aircraft has 10,200 images [47], Cifar100 has 60,000 images [30], Omniglot has 17,853 images [34], PatternNet has 38,400 images [91], ModelNet40 has 48,000 images [81]. These datasets are used for hash computations and queries.

We begin by demonstrating the feature computation time and hash computation time and query time required while using the output of the second last feature layer of pre-trained AlexNet and VGG-16. The results are given in Tables 7, 8 and 9. This paper involves the use of VGG-Flower Dataset for this purpose. In this paper, these features are denoted by Net-1 and Net-2. The minimum hash computation time and query time in milliseconds using different bits and hash-tables are shown in Table 8 for VGG-Flower dataset. Each entry represents values for (Net-1, Net-2). For 16-bits hash length, minimum hash computation time with Net-1 is 125.4 ms for hash-table 4 and with Net-2 is 150 ms for hash-table 2. Similarly, the results for varied hash bits lengths and different number of hash-tables are presented in Table 8. Here, hash-table 2 represents two number of hash-tables.

The query time is in milliseconds for Net-1 and Net-2 for VGG-Flower dataset. For different hash bits, query time with Net-1 and Net-2 features are presented in Table 9. For 16-bits hash length, minimum query time with Net-1 is 28 ms and with Net-2 is 87 ms for hash-table 2. The results of other hash bits with varied number of hash-tables are presented in Table 9.

We draw two conclusions from the results: a) Alex-Net will reduce the time complexity to a certain extent and increases the image retrieval speed. b) We need to reduce the feature dimensions further, in order to get a significantly lower hash computation time. It should be noted that we extract features from the 5th convolutional layer and perform a cross-domain training including perturbed data. We selected features from fully connected layers for reducing the feature vector size instead of retraining. We promote the fact that as we want a cross-domain feature extraction network, we prefer selecting features from lower layers.

For training the cross-domain net, we used 18 k original images each from datasets OMNI, ModelNet40, PatternNet and ImageNet. We included eight different kinds of perturbations in these images as given in the Table 10.

We configured a learning rate of .001 to learn the non-shared parameters in each case using a cross-entropy loss with an adam optimizer for 30 epochs. Table 11 shows classification performance metrics using the Omni dataset. We presented two scenarios, w/o_P denotes cross-domain net trained without perturbations but tested with perturbed images while w_P denotes both training and testing using perturbations. It is observed that the latter provides better results and is hence more robust to all kinds of perturbations for any data.

Tables 12 and 13 presented the hash computation time and image query time using different combinations. The experiments are conducted on a cross-domain net dataset with different hash bits and number of hash-tables. We use VGG-flower dataset, OMNI, CIFAR, PatternNet, Aircraft for this purpose. The average and minimum hash computation time in ms using different bits and hash-tables is shown in Table 12 for using cross-domain Net. For 16-bits hash-length, lowest average hash computation time is 15.4 ms for hash-table 4 and lowest minimum hash computation time is 14.7 ms for hash-tables 2 and hash-Table 4.

For cross-domain net, average and minimum query time is presented in Table 13. For 16-bits hash length, lowest average query time is 216.8 ms for hash-table 2 and lowest minimum query time is 40 ms for hash-table 2. The results of other hash bits with varied number of hash-tables are presented in Table 13.

In Table 14, we presented the number of hits obtained while querying different perturbed images. We observedthree cases, first we created the signature database with perturbed images and then queried perturbed images. Incase 2 and 3, we created signatures using the original images and queried perturbed images. However, in case 2,we used the w/o_P version of the network. The best results are obtained in case 3 with the w_P cross-domain net.

Table 14 Hits table using cross-domain net

Perturbations	Case 1	Case 2	Case 3
0	0.458	1.000	1.000
1	0.186	0.126	0.670
2	0.198	0.588	0.914
3	0.452	0.890	1.000
4	0.340	0.670	0.902
5	0.204	0.106	0.570
6	0.434	0.960	0.998
7	0.458	0.452	0.008
8	0.176	0.018	0.364

Many interesting results along the same line were found in [71]. The authors utilize a ResNet architecture for extracting features. They show that the number of misses increases with varying perturbations. In our work, we focus on the fact that a cross-domain learning with a perturbed dataset can handle this impact to a large extent.

It is observed that for solarization and sharpening, we got a lower number of hits when indexed as in case 3. Many experiments with varying parameters showed the same results. It should be noted that blur, compression, gaussian noise can be varied in the image. A low impact of these perturbations results in higher number of hits whereas higher amount results in lower hits. For example, with a variance of 0.005 for Gaussian noise the hit is 0.96. While training the cross-domain net we fixed these parameters. We took a blur window of 15 pixels, a compression ratio of 50% and gaussian noise variance of 0.1. The hits shown here are using the same extremes. If we lower these, then the hits increase. Although we get a good classification accuracy with blurred images, yet they return 57% hits when indexed. This indeed is a very interesting observation and requires more attention. One way to deal with the situation is to perform filtering operations before feature computation. It is also evident that the hits for noisy gaussian images will further increase if a denoising is performed. However, as it is not known beforehand whether the query image is blurry, noisy or not, two approaches can be taken. One is to build a classifier which identifies blur or noise and apply a deblur or denoising filter on them. The selection of the deblur or denoising filter will again be data dependent as filters working on normal images may fail to work on medical images [11–13, 31, 32]. A better approach will be to pass all images through a network possibly a cross-domain one trained in such a way that it cleans noisy images or blurry ones but leaves all other images untouched.

Another area that requires further attention is the Image reconstruction using the B+ tree information. Our current experiments were limited to single step crops or rotations. We are yet to explore scenarios where multiple operations are involved on the same image.

5 Conclusion and future research

The research article proposed a deep CNN based image detection technique for online deduplication to detect exact or near-exact images. We proposed Hot Vector Decomposition (HDV) for image patch generation of near- exact images that performs better in terms of higher and stable accuracy. The main objective of this research is to find exact or near-exact images using feature extraction techniques in a cloud storage system using normal as well as perturbed cross-domain images in the form of blur, noise, compression, lighting variations and many more. The performance of concurrent exact or near-exact image detection techniques is evaluated and compared in terms of matching accuracy and time. The performance of the individual key point feature extractor is compared based on matching accuracy and time. HDV image matching technique for deduplication is proposed with a combination of SURF with DWT and DCT-block based on the superior performance of SURF compared to other key-point descriptors. The performance comparison of proposed HDV with other features extraction techniques on DWT is implemented. The results demonstrate that HDV exhibits higher and stable image matching accuracy in all three types of image deformations with relatively least computation time. The proposed fine-tuned CNN based feature extraction technique for online deduplication provides better accuracy when compared to AlexNet and VGGNet. Image classifiers are evaluated based on different image variations to detect exact or near-exact images, and various findings are discussed. It has been observed that the Bayesian image classifier performs better in illumination, pose image variations and exact images. Also, we deal with cross domain images along with eight perturbations in each of them and tested in terms of hash computation time, image query time and hits obtained. From our experiments, we can conclude that our cross-domain perturbed dataset can handle all perturbations better as compared to a general network. Also, the average minimum hash computation time reduces significantly by reducing the feature dimensions for cross-domain network. To further improve the performance of deduplication, image patches may be used to reconstruct the images using a data structure such as B+ tree for efficient storage of near-exact images. The difference with the base or original image in the form of transformations of near-exact images may be stored. It may further reduce the requirement of storage and storage cost.

Acknowledgements This research is supported by the Department of Science and Technology, Government of India under WOS (Women Scientists Scheme) sponsored research project entitled “Distributed Data Deduplication Technique for efficient Cloud-Based Storage System” under File No: SR/WOS-A/ET-119/2016.

Declarations

Conflict of interest The authors declare that they have no conflict of interests.

References

1. Alkawaz MH, Sulong G, Saba T, Rehman A (2018) Detection of copy-move image forgery based on discrete cosine transform. Neural Comput & Applic 30(1):183–192. https://doi.org/10.1007/s00521-016-2663-3
2. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M (2010) A view of cloud computing. Commun ACM 53:50–58
3. Banerji S, Sinha A, Liu C (2013) New image descriptors based on color, texture, shape, and wavelets for object and scene image classification. Neurocomputing 117:173–185. https://doi.org/10.1016/j.neucom.2013.02.014
4. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (SURF). Comput Vis Image Underst 110(3):346–359. https://doi.org/10.1016/j.cviu.2007.09.014
5. Biadgie Y, Sohn KA (2014) Feature detector using adaptive accelerated segment test. In: 2014 international conference on information science and applications (ICISA) 2014, (pp. 1-4). IEEE. https://doi.org/10.1109/ICISA.2014.6847403
6. Chen CC, Hsieh SL (2015) Using binarization and hashing for efficient SIFT matching. J Vis Commun Image Represent 30:86–93. https://doi.org/10.1016/j.jvcir.2015.02.014

7. Chen M, Wang Y, Zou X, Wang S, Wu G (2012) A duplicate image deduplication approach via Haar wavelet technology. In: 2nd International Conference on Cloud Computing and Intelligence Systems, IEEE 2: 624–628. https://doi.org/10.1109/CCIS.2012.6664249
8. Chen M, Wang S, Tian L (2013) A high-precision duplicate image deduplication approach. J Comput 8(11): 2768–2775. https://doi.org/10.4304/jcp.8.11.2768-2775
9. Dharshini T, Angelina JJR (2019) Review and analysis of image and video deduplication techniques. J Inf Comput Sci 9(12)
10. Ding C, Tao D (2015) Robust face recognition via multimodal deep face representation. IEEE Trans Multimedia 17(11):2049–2058. https://doi.org/10.1109/TMM.2015.2477042
11. Diwakar M, Kumar M (2018) A review on CT image noise and its denoising. Biomed Signal Process Control 42:73–88. https://doi.org/10.1016/j.bspc.2018.01.010
12. Diwakar M, Kumar M (2018) CT image denoising using NLM and correlation-based wavelet packet thresholding. IET Image Process 12(5):708–715
13. Diwakar M, Singh P (2010) CT image denoising using multivariate model and its method noise thresholding in non-subsampled shearlet domain. Biomed Signal Process Control 57:101754. https://doi.org/10.1016/j.bspc.2019.101754
14. Foo JJ, Sinha R, Zobel J (2007) SICO: a system for detection of near-duplicate images during search. Multimedia and Expo, IEEE International Conference on, 595–598. https://doi.org/10.1109/ICME.2007.4284720
15. Gang H, Yan H, Xu L (2015) Secure image deduplication in cloud storage. In: Khalil I, Neuhold E, Tjoa A, Xu L, You I (eds) Information and Communication Technology ICT-EurAsia, Lecture Notes in Computer Science, vol 9357. Springer, pp 243–251. https://doi.org/10.1007/978-3-319-24315-3_25
16. Girshick R, Donahue J, Darrell T, Malik J (2016) Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans Pattern Anal Mach Intell 38(1):142–158. https://doi.org/10.1109/TPAMI.2015.2437384
17. Grabner M, Grabner H, Bischof H (2006) Fast approximated SIFT. Lecture Notes in Computer Science, LNCS. In: Asian conference on computer vision, Springer, Berlin, 3851: 918–927. https://doi.org/10.1007/11612032_92
18. Harris C, Stephens M (1988) A combined corner and edge detector. Procedings of the Alvey Vision Conference 1988, 23.1-23.6. https://doi.org/10.5244/C.2.23
19. Hua Y, He W, Liu X, Feng D (2015) SmartEye: real-time and efficient cloud image sharing for disaster environments. IEEE Conference on Computer Communications (INFOCOM) 1616-1624, https://doi.org/10.1109/INFOCOM.2015.7218541
20. Huang F, Zhou Z, Yang CN, Liu X, Wang T (2019) Original image tracing with image relational graph for near-duplicate image elimination. Int J Comput Sci Eng 18:294–304. https://doi.org/10.1504/IJCSE.2019.098540
21. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia ACM, pp. 675-678. https://doi.org/10.1145/2647868.2654889
22. Juan L, Gwun O (2009) A comparison of sift, pca-sift and surf. Int J Image Process (IJIP) 3(4):143–152
23. Jung HM, Park WV, Lee WY, Lee JG, Ko YW (2011) Data deduplication system for supporting multi-mode. In Asian Conference on Intelligent Information and Database Systems, Lecture Notes in Computer Science, Springer 6591:78-87. https://doi.org/10.1007/978-3-642-20039-7-8
24. Jyoti RR (2019) A hybrid approach for effective image deduplication using PCA, SPIHT and compressive sensing. Women Institute of Technology Conference on Electrical and Computer Engineering (WITCON ECE) 129-134. https://doi.org/10.1109/WITCONECE48374.2019.9092894
25. Kalia R, Lee KD, Samir BVR, Je SK, Oh WG (2011) An analysis of the effect of different image preprocessing techniques on the performance of SURF: speeded up robust features. 17th Korea-Japan joint workshop on Frontiers of computer vision, FCV 2011. https://doi.org/10.1109/FCV.2011.5739756
26. Kaur R, Chana I, Bhattacharya J (2018) Data deduplication techniques for efficient cloud storage management: a systematic review. J Supercomput 74(5):2035–2085. https://doi.org/10.1007/s11227-017-2210-8
27. Ke Y, Sukthankar R (2004) PCA-SIFT: "a more distinctive representation for local image descriptors, in proc. Conf. Computer vision and Pattern Recognition, pp. 511-517, 2004. https://doi.org/10.1109/CVPR.2004.1315206
28. Ke Y, Sukthankar R, Huston L (2004) Efficient near-duplicate detection and sub-image retrieval. In Proceedings of ACM International Conference on Multimedia (MM), 4(1).
29. Kordopatis-Zilos G, Papadopoulos S, Patras I, Kompatsiaris I (2019) Finding near-duplicate videos in large-scale collections. In: Mezaris V, Nixon L, Papadopoulos S, Teyssou D (eds) Video Verification in the Fake News Era. Springer. https://doi.org/10.1007/978-3-030-26752-0_4

30. Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Computer science department, University of Toronto. Tech Rep 1(4):7
31. Kumar M, Diwakar M (2018) CT image denoising using locally adaptive shrinkage rule in tetrolet domain. J King Saud Unive-Comput Info Sci 30(1):41–50. https://doi.org/10.1016/j.jksuci.2016.03.003
32. Kumar M, Diwakar M (2019) A new exponentially directional weighted function based CT image denoising using total variation. J King Saud Univ-Comput Inf Sci 31(1):113–124. https://doi.org/10.1016/j.jksuci.2016.12.002
33. Kumar PM, Devi GU, Basheer S, Parthasarathy P (2020) A study on data de-duplication schemes in cloud storage. Int J Grid Util Comput 11:509–516. https://doi.org/10.1504/IJGUC.2020.108450
34. Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. Science 350:1332–1338. https://doi.org/10.1126/science.aab3050
35. Latif A, Rasheed A, Sajid U, Ahmed J, Ali N, Ratyal NI, Zafar B, Dar SH, Sajid M, Khalil T (2019) Content-based image retrieval and feature extraction: a comprehensive review. Math Probl Eng. https://doi.org/10.1155/2019/9658350
36. Lee SH, Chan CS, Wilkin P, Remagnino P (2015) Deep-plant: plant identification with convolutional neural networks. Proceedings - International Conference on Image Processing, ICIP, pp.452–456. https://doi.org/10.1109/ICIP.2015.7350839
37. Lei Y, Qiu G, Zheng L, Huang J (2014) Fast near-duplicate image detection using uniform randomized trees. ACM Trans Multimed Comput Commun Appl (TOMM) 10(4):1–15. https://doi.org/10.1145/2602186
38. Leutenegger S, Chli M, Siegwart RY (2011) BRISK: binary robust invariant scalable keypoints. Proceedings of the IEEE International Conference on Computer Vision, 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542
39. Li L, Zic J (2014) Image matching algorithm based on feature-point and DAISY descriptor. J Multimed 9(6):829–834. https://doi.org/10.4304/jmm.9.6.829-834
40. Li J, Qian X, Li Q, Zhao Y, Wang L, Tang YY (2015) Mining near duplicate image groups. Multimed Tools Appl 74(2):655–669. https://doi.org/10.1007/s11042-014-2008-0
41. Li X, Li J, Huang F (2016) A secure cloud storage system supporting privacy-preserving fuzzy deduplication. J Soft Comput 20(4):1437–1448. https://doi.org/10.1007/s00500-015-1596-6
42. Liang S, Wang P (2020) An efficient hierarchical near-duplicate video detection algorithm based on deep semantic features. In: Ro Y et al (eds) MultiMedia modeling. MMM lecture notes in computer science, vol 11961. Springer. https://doi.org/10.1007/978-3-030-37731-1_61
43. Liu YH (2018) Feature extraction and image recognition with convolutional neural networks. J Phys Conf Ser 1087(6):062032
44. Liu L, Lu Y, Suen CY (2015) Variable-length signature for near-duplicate image matching. IEEE Trans Image Process 24(4):1282–1296. https://doi.org/10.1109/TIP.2015.2400229
45. Liu D, Shen J, Wang A, Wang C (2020) Secure real-time image protection scheme with near-duplicate detection in cloud computing. J Real-Time Image Proc 17:175–184. https://doi.org/10.1007/s11554-019-00887-6
46. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94
47. Maji S, Rahtu E, Kannala J, Blaschko M, Vedaldi A (2013) Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151
48. Mao B, Jiang H, Wu S, Fu Y, Tian L (2014) Read-performance optimization for deduplication-based storage Systems in the Cloud. ACM Trans Storage 10(2):1–22. https://doi.org/10.1145/2512348
49. Miksik O, Mikolajczyk K (2012) Evaluation of local detectors and descriptors for fast feature matching. In: Proceedings of 21st international conference of pattern recognition (ICPR), 2012, pp. 2681-2684. IEEE
50. Mohamed SMA, Wang Y (2020) A survey on novel classification of deduplication storage systems. Distrib Parallel Databases. https://doi.org/10.1007/s10619-020-07301-2
51. Mohapatra S, Bajpai N, Swarnkar T, Mishra M (2020) Raw data redundancy elimination on cloud database. In: Das A, Nayak J, Naik B, Dutta S, Pelusi D(eds), Computational intelligence in pattern recognition. Adv Intel Sys Comput 1120:395–405. https://doi.org/10.1007/978-981-15-2449-3_34
52. Nbt Y, Ismail A, Majid NAA (2016) Deduplication image middleware detection comparison in standalone cloud database. Int J Adv Comput Sci Technol (IJACST) 5:12–18
53. Nian F, Li T, Wu X, Gao Q, Li F (2016) Efficient near-duplicate image detection with a local-based binary representation. Multimed Tools Appl 75(5):2435–2452. https://doi.org/10.1007/s11042-015-2472-1
54. Nilsback ME, Zisserman A (2006) A visual vocabulary for flower classification. In: IEEE computer society conference on computer vision and Pattern Recognition (CVPR'06), 2:1447-1454. https://doi.org/10.1109/CVPR.2006.42

55. Pang Y, Li W, Yuan Y, Pan J (2012) Fully affine invariant SURF for image matching. Neurocomputing 85: 6–10. https://doi.org/10.1016/j.neucom.2011.12.006
56. Paulin M, Douze M, Harchaoui Z, Mairal J, Perronin F, Schmid C (2015) Local convolutional features with unsupervised training for image retrieval. In: Proceedings of the IEEE international conference on computer vision, pp. 91-99
57. Paulo J, Pereira J (2014) A survey and classification of storage deduplication systems. ACM Comput Surv 47(1):1–30. https://doi.org/10.1145/2611778
58. Peker KA (2011) Binary sift: fast image retrieval using binary quantized sift features. In: Content-based multimedia indexing (CBMI), in 9th international workshop on 2011, pp. 217-222, IEEE. https://doi.org/10.1109/CBMI.2011.5972548
59. Pietro RD, Sorniotti A (2016) Proof of ownership for deduplication systems: a secure, scalable, and efficient solution. Comput Commun 82:71–82. https://doi.org/10.1016/j.comcom.2016.01.011
60. Raghavendra U, Fujita H, Bhandary SV, Gudigar A, Hong TJ, Acharya UR (2018) Deep convolution neural network for accurate diagnosis of glaucoma using digital fundus images. Inf Sci 441:41–49. https://doi.org/10.1016/j.ins.2018.01.051
61. Ramaiah NP, Mohan CK (2011) De-duplication of photograph images using histogram refinement. IEEE Recent Advances in Intelligent Computational Systems 391–395. https://doi.org/10.1109/RAICS.2011.6069341
62. Seo JS, Haitsma J, Kalker T, Yoo CD (2004) A robust image fingerprinting system using the radon transform. Signal Process Image Commun 19(4):325–339. https://doi.org/10.1016/j.image.2003.12.001
63. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
64. Song J (2017) Binary generative adversarial networks for image retrieval. arXiv preprint arXiv:1708.04150
65. Song J, Yang Y, Huang Z, Shen HT, Luo J (2013) Effective multiple feature hashing for large-scale near-duplicate video retrieval. IEEE Trans Multimedia 15:1997–2008. https://doi.org/10.1109/TMM.2013.2271746
66. Song J, Zhang H, Li X, Gao L, Wang M, Hong R (2018) Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Trans Image Process 27:3210–3221. https://doi.org/10.1109/TIP.2018.2814344
67. Song J, He T, Gao L, Xu X, Hanjalic A, Shen HT (2020) Unified binary generative adversarial network for image retrieval and compression. Int J Comput Vis 128:2243–2264. https://doi.org/10.1007/s11263-020-01305-2
68. Srinivas S, Sarvadevabhatla RK, Mopuri KR, Prabhu N, Kruthiventi SS, Babu RV (2016) A taxonomy of deep convolutional neural nets for computer vision. Front Robot AI https://doi.org/10.3389/frobt.2015.00036
69. Sujatha G, Raj JR (2020) A study on image hashing techniques for implementing deduplication. In: Hemanth D., Vadivu G., Sangeetha M., Balas V. (eds), In Artificial Intelligence Techniques for Advanced Computing Applications. Lecture Notes in Networks and Systems 130:529-534. https://doi.org/10.1007/978-981-15-5329-5_49
70. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9, https://doi.org/10.1109/CVPR.2015.7298594
71. Takeshita J, Karl R, Jung T (2020) Secure single-server nearly-identical image deduplication. arXiv preprint arXiv:2005.02330
72. Thaiyalnayaki S, Sasikala J, Ponraj R (2018) Indexing near-duplicate images in web search using minhash algorithm. Mater Today: Proc 5(1):1943–1949. https://doi.org/10.1016/j.matpr.2017.11.297
73. Thaiyalnayaki S, Sasikala J, Ponraj R (2019) Detecting near-duplicate images using segmented minhash algorithm. Int J Adv Intell Paradigms 12:192–206. https://doi.org/10.1504/IJAIP.2019.096963
74. Thyagharajan KK, Kalaiarasi G (2020) A review on near-duplicate detection of images using computer vision techniques. Arch Comput Methods Eng 6:1–20. https://doi.org/10.1007/s11831-020-09400-w
75. Tian Y, Luo P, Wang X, Tang X (2015) Deep learning strong parts for pedestrian detection. Proceedings of the IEEE International Conference on Computer Vision, 2015 Inter, 1904–1912. https://doi.org/10.1109/ICCV.2015.221
76. Vedaldi A, Lenc K (2015) MatConvNet - Convolutional Neural Networks for MATLAB. In: Proceedings of the 23rd ACM international conference on Multimedia, 2015 pp. 689-692. https://doi.org/10.1145/2733373.2807412
77. Velmurugan K, Baboo LDSS (2011) Content-based image retrieval using SURF and colour moments. Global J Comput Sci Technol 11(10)
78. Wan J, Han S, Zhang J, Zhu B, Zhou L (2013) An image management system implemented on open-source cloud platform. Proceedings - IEEE 27th International Parallel and Distributed Processing Symposium Workshops and PhD Forum, IPDPSW 2064–2070. https://doi.org/10.1109/IPDPSW.2013.176

79. Wang J, Chen X (2016) Efficient and secure storage for outsourced data: a survey. Data Sci Eng 1(3):178–188. https://doi.org/10.1007/s41019-016-0018-9
80. Wang JG, Li J, Lee CY, Yau WY (2010) Dense SIFT and Gabor descriptors-based face representation with applications to gender recognition. In: 11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010, 1860–1864. https://doi.org/10.1109/ICARCV.2010.5707370
81. Wu Z, Song S, Khosla A, Yu F, Zhang L, Tang X, Xiao J (2015) 3D shapenets: a deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 1912–1920
82. Xia W, Zou X, Jiang H, Zhou Y, Liu C, Feng D, Hua Y, Hu Y, Zhang Y (2020) The Design of Fast Content-Defined Chunking for data deduplication based storage systems. IEEE Trans Parallel Distrib Syst 31:2017–2031. https://doi.org/10.1109/TPDS.2020.2984632
83. Yang S, Luo P, Loy CC, Tang X (2015) From facial parts responses to face detection: a deep learning approach. In: Proceedings of the IEEE International Conference on Computer Vision 2015, pp. 3676-3684. https://doi.org/10.1109/ICCV.2015.419
84. Yao J, Yang B, Zhu Q (2015) Near-duplicate image retrieval based on contextual descriptor. IEEE Signal Process Lett 22(9):1404–1408 https://digitalcommons.unomaha.edu/compscifacpub/25
85. Yu X, Huang T (2009) A SIFT-based image fingerprinting approach robust to geometric transformations. In: Circuits and systems, ISCAS 2009. IEEE international symposium on, pp. 1665-1668. IEEE. https://doi.org/10.1109/ISCAS.2009.5118093
86. Zargar AJ, Singh N, Rathee G, Singh AK (2015) Image data-deduplication using the block truncation coding technique. IEEE International Conference on Futuristic Trends in Computational Analysis and Knowledge Management 154–158. https://doi.org/10.1109/ABLAZE.2015.7154986
87. Zeng X, Wen S, Zeng Z, Huang T (2018) Design of memristor-based image convolution calculation in convolutional neural network. Neural Comput & Applic 30(2):503–508. https://doi.org/10.1007/s00521-016-2700-2
88. Zhang J, Feng Z, Su Y (2008) A new approach for detecting copy-move forgery in digital images. 11th IEEE Singapore International Conference on Communication Systems, ICCS 2008, 362–366. https://doi.org/10.1109/ICCS.2008.4737205
89. Zhang D, Sun Z, Jia K (2020) Near-duplicate video detection based on temporal and spatial key points. In: Kountchev R, Patnaik S, Shi J, Favorskaya M (eds) Advances in 3D image and graphics representation, analysis, computing and information technology. Smart innovation, systems and Technologies, vol 180. Springer, Singapore, pp 129–137. https://doi.org/10.1007/978-981-15-3867-4_16
90. Zhao WL, Ngo CW, Tan HK, Wu X (2007) Near-duplicate keyframe identification with interest point matching and pattern learning. IEEE Trans Multimedia 9(5):1037–1048. https://doi.org/10.1109/TMM.2007.898928
91. Zhou W, Newsam S, Li C, Shao Z (2018) PatternNet: a benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS J Photogramm Remote Sens 145:197–209. https://doi.org/10.1016/j.isprsjprs.2018.01.004
92. Zuo F, de With PH (2004) Fast facial feature extraction using a deformable shape model with haar-wavelet based local texture attributes. In: IEEE international conference on image processing, ICIP'04, 3:1425-1428. https://doi.org/10.1109/ICIP.2004.1421330
93. Zuo P, Hua Y, Liu X, Feng D, Xia W, Cao S, Wu J, Sun Y, Guo Y (2017) BEES: bandwidth- and energy-efficient image sharing for real-time situation awareness. Proceedings - International Conference on Distributed Computing Systems, 1510–1520. https://doi.org/10.1109/ICDCS.2017.36