Distributed Data Deduplication Techniques for Efficient Cloud Storage System

A Thesis

submitted in partial fulfilment of the requirements for the award of the degree of

Doctor of Philosophy

in

Computer Science and Engineering

Submitted by:

Ravneet Kaur
(Registration No: 901403012)

Under the guidance of

Dr. Inderveer Chana
Professor, CSED
Dean (Student Affairs)

Dr. Jhilik Bhattacharya
Associate Professor, CSED

Thapar Institute of Engineering and Technology
Patiala-147004, Punjab, India
October 2023

Certificate

I hereby certify that the work, which is being presented in the thesis, entitled "Distributed Data Deduplication Techniques for Efficient Cloud Storage System", in partial fulfillment of the requirements for the award of the degree of Doctor of Philosophy and submitted to the institution is an authentic record of my work carried out under the supervision of Dr. Inderveer Chana and Dr. Jhilik Bhattacharya. I have cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.

The matter presented in this thesis has not been submitted either in-part or full to any other University/Institute for the award of any other degree.

Ravneet Kaur

(Ravneet Kaur)
Registration No. 901403012

This is to certify that the above statements made by the candidate are correct and true to the best of my knowledge.

Verified by:

(Dr. Inderveer Chana)
Supervisor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala, Punjab, India

(Dr. Jhilik Bhattacharya)
Supervisor
Computer Science and Engineering Department
Thapar Institute of Engineering and Technology, Patiala, Punjab, India

DEDICATED TO MY LOVING
PARENTS AND
KIDS

Acknowledgements

This Ph.D. thesis is the outcome of a challenging journey, upon which Waheguruji has given me wisdom and perseverance to accomplish this work successfully. First and foremost, I am eternally grateful to Sri Guru Granth Sahib Ji for his enduring blessings and unwavering support during my tough times.

I wish to extend my heartfelt gratitude to my esteemed supervisors, Dr. Inderveer Chana, Professor and Dean (Student Affairs) and Dr. Jhilik Bhattacharya, Associate Professor, Computer Science and Engineering Department, Thapar Institute of Engineering and Technology (Deemed University), Patiala (India), for their invaluable guidance, enthusiastic encouragement, and valuable feedback throughout the course of this research endeavor.

I want to express my sincere gratitude to Dr.Inderveer Chana for her constant advice and encouragement at every step of my Ph.D. studies. Her constant support, encouragement, and immense knowledge provided solace during my most challenging times. I am indebted to her insightful discussions and enlightening suggestions that helped me shape the direction of this research. The meticulous and diligent proofreading of this work contributed to the successful presentation of this thesis at the desired level of quality. Her constant pursuit of perfection, guidance, and valuable suggestions have guided me at every step of my professional journey. I sincerely thank her from the bottom of my heart and will be indebted to her throughout my lifetime.

Sincere thanks must also go to Dr. Jhilik Bhattacharya for her continuous support, immense knowledge, and always being willing to answer my questions. The observations and comments provided valuable guidance for determining the research's direction and conducting a thorough investigation. I have greatly benefited from her extensive knowledge and careful editing. Her dedication and effort have significantly enhanced the research's quality. In particular, I appreciate her enthusiasm and endless energy that inspired me to put maximum effort into research.

I offer my sincere gratitude to our director Prof. Padmakumar Nair, Dr. N. Tejo Prakash, Dean (Research Sponsored Projects), and Dr. Shalini Batra, Professor and Head, Computer Science and Engineering Department for providing the necessary academic and administrative assistance in the completion of this work. I sincerely thank the members of my Doctoral committee - Dr. Rinkle Rani, Dr. Prashant Rana, and Dr. Kulbir Singh for their academic support and invaluable comments.

I am also thankful to Ph.D. Coordinator Dr. Sushma Jain and all the Computer Science and Engineering Department faculty members, Thapar Institute of Engineering and

Technology, Patiala for their cooperation and support. I also acknowledge the cooperation rendered to me by the office and the laboratory staff of this department. I would also like to thank all my friends for their continuous motivation and moral support. I especially thank S. Aiemanpreet Singh and CITM Department, Thapar Institute of Engineering and Technology for helping and supporting me in my tough times.

I would like to express my gratitude to the Department of Science and Technology, Government of India for providing financial assistance through a project under Women Scientist Scheme (SR/WOS-A/ET-119/2016) for my doctoral research.

This thesis would not have been possible without the unconditional support of my family. I want to acknowledge my pillars of strength, my grandfather Late Capt. Sampuran Singh, my grandmother Smt. Rawel Kaur and my parents S. Bakhshish Singh and Smt. Jaswinder Kaur for their blessings and eternal support. I owe a lot to my parents for their unwavering support and guidance throughout my personal and academic life who always lived each day from beginning till completion with a dream of their daughter getting a degree of Doctorate. I will always be grateful to them for their patience and compassion in every sphere of life. My mother is my role model and has always been an eternal source of inspiration. I gratefully acknowledge the patience and love of my sister, brother, bhabhi, and nieces and nephews who extended their cooperation and encouragement to me. Special thanks to my sister Supreet Kaur for her selfless love, care, and dedicated efforts which contributed a lot for the completion of my thesis.

Immense gratitude to my husband S. Harcharanjit Singh for his understanding, tremendous moral support, and the help he extended to me all the time. He inspired me in all dimensions of life and instilled confidence in me to complete this journey successfully. I would always be indebted to him for his love and commitment. I want to express my sincere gratitude to my mother-in-law for her support and encouragement. I gratefully acknowledge her cooperation and blessings for me. These acknowledgments would remain incomplete if I do not mention the names of my daughter Gunveen Kaur and son Ranfateh Singh, who have suffered a lot due to lack of attention during this journey. I owe everything to them, and this thesis would not have been completed without their everlasting support.

Besides this, several near and dear ones and relatives have helped me in the successful completion of this work. I wholeheartedly thank every person not mentioned above but who contributed to my thesis directly or indirectly.

Ravneet Kaur

Ravneet Kaur
August, 2023
Patiala

Abstract

Cost-effective storage management has emerged as a critical challenge for cloud storage systems given the the exponential growth of digital data in contemporary times. Storing vast amounts of internet-generated data efficiently requires substantial computing and storage resources. This issue is further exacerbated by significant, redundant data, significantly impacting storage requirements.

This thesis investigates and proposes deduplication techniques to reduce duplicate data in cloud storage systems. Data deduplication is crucial for large-scale distributed systems, particularly in dynamic infrastructures like cloud storage. The performance of deduplication directly affects the overall efficiency and cost of the system. By reducing data volumes, storage providers can mitigate the costs of running large storage systems and conserve energy consumption.

This work proposes an efficient data deduplication technique that effectively manages and eliminates duplicates in cloud storage systems. A comprehensive investigation of various deduplication techniques has been undertaken to study their efficacy in storage systems. Data-based deduplication techniques are categorized into text, image, and video-based methods. Scalability, reliability, distributed environment techniques, and fingerprint indexing emerge as key challenges for distributed data deduplication in cloud storage systems. This research work addresses these challenges and explores measures to overcome them.

The thesis focuses on image deduplication techniques in cloud storage systems, with the aim of minimizing exact or near-exact image duplicates. A novel CNN-based online image deduplication technique is proposed to detect such duplicates. A Fine-Tuned AlexNet for cross-domain online image deduplication is proposed for exact near exact image detection. Comparative analysis with existing CNN techniques demonstrates better accuracy in the proposed fine-tuned CNN-based feature extraction technique, surpassing AlexNet and VGGNet by 24% and 17%, respectively. Additionally, the research introduces the Hot Decomposition Vector (HDV), which optimally stores dissimilar parts of near-exact images for efficient reconstruction using a base image. HDV outperforms traditional image feature extraction approaches in terms of image-matching accuracy and computing time.

Furthermore, a novel EsDeDUP energy-saving technique is proposed to analyze the impact of exact or near-exact image deduplication techniques on energy savings and storage reduction. Fine-tuned CNN-based image deduplication techniques have been proposed to compute the effectiveness of image deduplication techniques and compared with existing

hash-based image deduplication techniques. The technique assesses the power consumption and performance of various deduplication approaches to ascertain their energy-saving potential. The work evaluates the performance and power consumption of four hash-based duplicate image detection techniques: phash, whash, ahash, and dhash. Additionally, this research proposed fine-tuned CNN-based deduplication techniques using neural structures such as fine-tuned AlexNet, fine-tuned VGG-NET-16, and fine-tuned VGG-NET-19 for extracting exact and near-exact duplicate images.

Empirical results demonstrate the effectiveness of the proposed fine-tuned CNN-based deduplication techniques, showcasing top-5 accuracy rates of 83.1%, 93.2%, and 92.8% for fine-tuned AlexNet, VGGNet-19, and VGGNet-16, respectively, using augmented ImageNet-Min dataset. Furthermore, the CNN-based deduplication method achieves storage reduction of 37.2% to 42.4% when applied to augmented ImageNet-Mini datasets. Conversely, the hash-based techniques (aHash, dHash, pHash, and wHash) exhibit top-5 accuracy rates of 23.4%, 21.8%, 38.3%, and 37.1%, respectively, using augmented ImageNet-Min dataset, thereby achieving storage reduction of 11% to 17.4%. Fine-tuned CNN-based deduplication techniques exhibit promising results in terms of accuracy but require higher power consumption compared to hash-based techniques for exact and near-exact image detection.

This research work contributes to the advancement of cloud storage efficiency through innovative deduplication techniques, with a particular focus on image data. The proposed methods offer potential cost savings, energy conservation, and improved performance in cloud storage systems.

Fellowship Awarded

Awarded **DST-WOS-A/ET-119/2016 Fellowship** under **File No: SR/WOS-A/ET-119/2016 (G)** with Project Entitled **"Distributed Data Deduplication Technique for efficient Cloud-Based Storage System"** from December 2017 to May 2021.

Women Scientists Scheme-A (WOS-A)" awarded by the Department of Science and Technology(DST), Govt. of India, availed at Thapar University, Patiala, India. This fellowship by GOI helps in providing opportunities to women scientists and technologists to empower Women Scientists struggling through their individuality. This fellowship helped in the successful completion of this research work.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

IDC	International Data Corporation
ZB	Zettabytes
I/O	Input/Output
IT	Information Technology
VM	Virtual Machine
HDV	Hot Decomposition Vector
CPU	Central Processing Unit
LAN	Local Area Network
DRAM	Dynamic Random Access Memory
SSD	Solid State Drive
CNN	Convolutional Neural Network
GPU	Graphics Processing Unit
RAPL	Running Average Power Limit
WAN	Wide Area Network
DDFS	Data Domain File System
POD	Performance-Oriented I/O Deduplication
MD5	Message Digest
SHA1	Secure Hash Algorithm
CDGT	Content Deduplication with Granularity Tweak
TDDFS	Tier-aware Data Deduplication-based File System
LDFS	Low latency in-line data deduplication file system
DARE	Deduplication Aware Resemblance detection and Elimination scheme
DupAdj	Duplicate-Adjacency
HAR	History-Aware Rewriting algorithm
CAF	Cache-Aware Filter
RevDedup	Reverse deduplication
NED	Near Exact Defragmentation
SBBS	Sliding blocking algorithm with backtracking sub-blocks
HANDS	Heuristically Arranged NonBackup Inline Deduplication System
NNP	N-Neighborhood Partitioning
CBR	Context Based Rewriting
CFL-SD	Cacheaware Chunk Fragmentation Level and Selective Deduplication
DeFFS	Duplicated eliminated Flash File System
FLOMD	Fast and Low Overhead Memory Deduplication

RCE	Reference-Count based Eviction Bitmap
BHI	Bitmap based Hotness Identification
CDAC	Content-driven Deduplication-Aware Cache
LIPA	Learning-based Indexing and Prefetching Approach
TLE-LRU	Temporal Locality Estimation- Least Recently Used
SLADE	Stream Locality Aware DEduplication
AppDedupe	Application-Aware Deduplication
SiLo	Similarity-locality based Indexing
CRP	Commonly Repeated Patterns
LT-LH	Lightweight Triple-leveled Hashing
DSHA	Distributed Storage Hash Algorithm
TTTD	Two Threshold Two Divisor
CB-TTTD	Content Based Two Threshold Two Divisor
DBA	Dynamic Bloom Filter Array
RMD	Resemblance and Mergence based Deduplication
RAM	Rapid Asymmetric Maximum
HDFS	Hadoop Distributed File System
ILB	Index Lookup in-Batch
IUB	Index Update in-Batch
INC-K	Incremental Modulo-K
CP-ABE	Ciphertext-Policy Attribute Encryption
BIAD	Blockchain-based Cloud Storage Integrity Auditing with Secure Deduplication
AES-CBC	Advanced Encryption Standard- Cipher Block Chaining
RSE-PoW	Role Symmetric Encryption algorithm and proof of ownership
MM	Meta Data Manager
RSSS	Ramp Secret Sharing Scheme
NCS	Neighbourhood Correlation Sequence
IDDTLC-CHS	Intelligent Data Deduplication with Deep Transfer Learning Enabled Classificat
IBPRE	Identity-based Proxy Re-encryption
IBPoW	Identity-based Proof of ownership
CLPRE	Certificateless Proxy Reencryption
IBPoW	Identity-based Proof of ownership
PoW-CLS	Proof of Ownership based on Certificateless Signature
MECC	Modified Elliptic Curve Cryptography
CE	Convergent Encryption
FC-LID	File classifier based Linear Indexing Deduplication
LHRG	Linear Hashing with Representative Group
NED	Near Exact Defragmentation Scheme

SAR	SSD (solid-state drive) - Assisted Read Scheme
ALG	Application-Aware Local Global
DeDU	Data Deduplication over Engineering oriented Cloud Systems
SIFT	Scale-Invariant Feature Transform
SURF	Speeded Up Robust Features
PCA-SIFT	Principal Component Analysis -SIFT
BRISK	Binary Robust Invariant Scalable Keypoints
FAST	Features from Accelerated Segment Test
BRIEF	Binary Robust Independent Elementary Features
ORB	Oriented FAST and Rotated BRIEF
NCS	Neighbourhood Correlation Sequence Algorithm
(SGO	Seagull Optimization Algorithm
ANNS	Approximate Nearest Neighboring Search
LSH	Locality Sensitive Hashing
VGG-16	Visual Geometry Group
SPIHT	Set Partitioning in Hierarchical Trees
CSP	Cloud Service Provider
SVC	Scalable Video Coding
CBIR	Content-based Image Retrieval
BTC	Block Truncation Coding
DoG	Difference-of Gaussian
LDP	Local-DifferencePattern
NDVC	Near Duplicate Video Clip
DCT	Discrete Cosine Transform
AKM	Approximate k-means
LBR	Local-based Binary Representation
BBF	Best Bin First Algorithm
LDA	Linear Discriminant Analysis
FMT	Fourier-Mellin Transform
SICO	Similar Image Collator
DWT	Discrete wavelet transform
LBP	Local Binary Patterns
PCSLBP	Probabilistic Center-symmetric Local Binary Pattern
SIFT	Dense Scale-Invariant Feature Transform
DCNN	Deep CNN
ReLU	Rectified Linear Unit
ILSVRC	ImageNet Large Scale Visual Recognition Challenge

ResNet	A residual neural network
MSER	Maximally Stable Extremal Regions
TN	True Negative
EsDeDUP	Energy-Saving Deduplication
SSDH	Supervised Semantics-Preserving Deep Hashing
aHash	Average Hashing
dHash	Difference Hashing
pHash	Perceptual Hashing
wHash	Wavelet Hashing
ED	Euclidean Distance
HD	Hamming Distance

Chapter 1

Introduction

In the era of "Big Data," data size keeps growing and comes from various sources like social media sites, sensor networks, etc. This has led to the development of Cloud Computing technology and various computational methods to manage the ever-growing volume of big data. Data deduplication is required to improve the Cloud's data storage efficiency and lower the cost of managing these huge amounts of multimedia data.

Data deduplication is one of the most efficient data reduction techniques that can effectively reduce storage space and manage network bandwidth overhead. It is a data compression technique that decreases storage capacity by retaining only a single instance of data and eliminating all duplicate copies of data files using logical pointers. Thus, it mitigates storage overhead and saves upload bandwidth. Data deduplication is used by many cloud storage services like Amazon S3, Bitcasa, and Microsoft Azure, as well as backup services like Dropbox and Memopal, to make storage more efficient. The research work presented in this thesis focuses on developing efficient data deduplication techniques for online image deduplication and measuring its performance and energy efficiency.

This chapter provides an overview of Data Deduplication, including its evolution, need, and working of Data Deduplication in cloud storage systems. It further provides a comprehensive analysis of data deduplication, specifically focusing on its classification at different levels and evaluating its advantages and disadvantages. Furthermore, this thesis has also provided an overview of the research motivation and objectives. The Chapter concludes with the contribution of this research work and its organization.

1.1 Data Deduplication: An Overview

The amount of data is growing at an unprecedented rate due to the rapid and widespread expansion of digitalized information systems. In 2008, the volume of digital information was approximately 500 Exabytes, with a fivefold growth every year [1]. In 2011, International Data Corporation (IDC) reported that data volume created and copied worldwide would be 35 zettabytes (ZB) by 2020[2]. A recent report from IDC predicted that the data volume generated and duplicated in the world is expected to grow from 33 ZB in 2018 to 175 ZB in 2025 [3, 4, 5], representing a tenfold increase from the amount of data created in 2016. One-third of the data comes from businesses and is unstructured, like office papers, web pages, electronic mail, digital photos, and audio and video files. Businesses retain such data for business administration, legal compliance and lawsuits, and data management. Businesses have difficulty storing and processing data because of the sheer volume of data constantly generated. In addition, data is duplicated across storage sites to improve the availability of hardware and software systems and minimize the implications of data loss. The majority of this redundant data adds to the strain on storage systems. To save storage space and network traffic, researchers are now working on ways to delete or reduce the amount of redundant data.

Cloud computing helps to deliver internet-based business requirements. The enterprise business requires computing and storage resources on demand to deliver services on the internet. So, cloud storage is a very popular service for enterprise businesses in cloud computing [6]. Customers use cloud storage to reduce their expenditure on purchasing and maintaining storage infrastructure. Customers want to pay for the amount of storage required that can be scaled up and down upon demand [7]. As the data size demand of customers in cloud computing is increasing, any reduction in data storage can help cloud storage service providers reduce the costs of running large storage systems and energy consumption. On the other hand, customers' digital data storage is growing rapidly, and they expect to meet the on-demand for cloud storage services anywhere and at any time. Customers need to reduce their payment overheads in procuring and continuing storage requirements. Cloud storage service providers have mandatory compliance to maintain high system availability. Cloud storage providers replicate the data to achieve availability and charge its cost to customers. So customers and providers of cloud storage services require a process to reduce the replicated data to reduce the storage cost. So, data deduplication techniques are designed to improve storage efficiency in cloud storage. Duplicate data is automatically removed from storage systems through a process known as deduplication, which saves space and bandwidth. Therefore, cost-effective storage solutions are required as the amount of data expands. Since deduplication [8] eliminates

redundant data and reduces storage space needs, it has become increasingly prevalent. Regarding deduplication, backup data is an ideal candidate for reducing storage. Deduplicating backup data can reduce storage space requirements by 10 to 20 times[9]. As a result, data deduplication techniques are employed in cloud storage to enhance storage efficiency.

Also, data has multiplied by orders of magnitude across all sources in the age of big data. Because of the rapid increase in data volumes, system designers face significant difficulty in providing highly efficient storage infrastructure support for upper-layer software, which is necessary to ensure consumer service quality. Storage efficiency is critical, and data deduplication is a key enabler for large-scale data storage [10]. Deduplication techniques have been implemented in both primary and backup storage systems to reduce the amount of data that must be kept. Commercial storage systems, such as the Dell-EMC Data Domain and the NetApp ONTAP, include the deduplication technique as a standard feature. Data deduplication can reduce the data kept on storage devices, and I/O and network traffic can also be conserved.

Deduplication can reduce storage requirements by up to 95 percent in some circumstances, according to NetApp [11]. Data deduplication is a specific compression technique for removing redundant data and increasing storage utilization. The approach identifies and removes any redundant data, saving only one copy of each piece of information. To eliminate duplicate copies, it uses pointers pointing to the unique copy [12]. Deduplication can reduce the storage space needed and handle the ever-increasing demand for storage capacity [13].

Microsoft conducted a file system analysis to determine the space savings tradeoff between whole-file and block-based deduplication on a Windows desktop[14]. Additionally, data deduplication was implemented on virtual machines. Along the same lines, Liquid, a distributed file system for Virtual Machine(VM) Images, was proposed. It addresses the problems faced in large-scale VM deployment as VM plays a crucial role in cloud computing and supports low storage consumption [15]. Data deduplication was also applied to a digital library that used similarity techniques to identify bibliographic records [16].

1.1.1 Evolution of Data Deduplication

Early in the 1950s, lossless and lossy data reduction techniques [17, 18] were introduced. Towards the end of the 1990s, space-efficient Intelligent Delta Compression algorithms were proposed to compress highly similar files or chunks. Data deduplication was coined in the early 2000s, as shown in Figure 1.1 to help large storage systems at high granularity levels [2]. It identifies duplicates by matching their cryptographically secure hash-based

signatures. Unlike typical data compression approaches, it reduces inter-file and intra-file redundancy for big datasets over multiple distributed storage servers, eliminating redundancy for small groups of files based on intra-file redundancy.

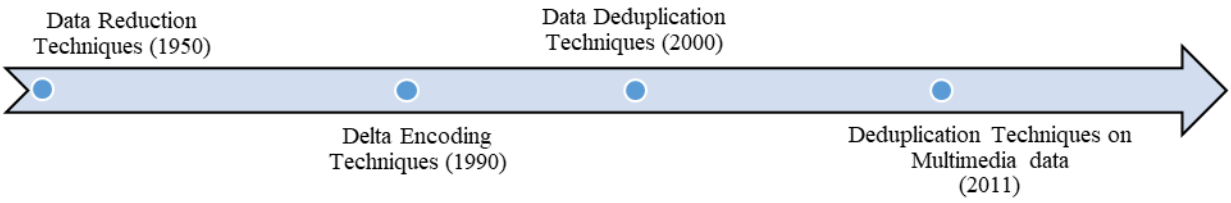


Figure 1.1: Evolution of Data Deduplication

Each file or chunk's cryptographic hash is calculated to identify duplication. These techniques were also applied to multimedia content in 2008, and multimedia deduplication was introduced in 2011. It assessed the similarity of images or video frames using feature extraction and hashing algorithms to search for duplicate multimedia content. These data deduplication techniques were developed to solve the issue of storage systems' growing data volumes [19]. However, image-based duplication techniques can be applied to video content as video frames are extracted, and image deduplication can be applied. Image deduplication, particularly for feature extraction using SURF, BRISK, or CNN techniques, may not directly apply to other data types. However, the principles of data chunking, hashing, and comparison can be adapted to various data types. This adaptability allows for efficient storage and data transfer improvements across a wide range of digital content. Finding an ideal technique for all data type contents is an open challenge. There is significant potential for further research and extension of this work to address this open and compelling research problem.

Huffman and dictionary coding are standard data compression techniques that operate at the byte or text level [17, 18, 19], whereas deduplication techniques reduce redundancy at the file or chunk level.

1.1.2 Need for Data Duplication

Information protection is critical to the daily operations of a business. Information is one of the company's most valuable assets in the digital era, and an effective and controllable backup and recovery strategy has become an IT(Information Technology) imperative. The information is saved in a computer storage system; even a single storage error or power outage can put a substantial quantity of data at risk. Several solutions that improve the accessibility and dependability of digital data, such as lower-level disk mirroring and data replication at the application level, have been introduced to safeguard storage against

this risk. Historically, backups of the original data were made to recover it in the event of data loss [20] or failure. Due to the exponential growth of digital data, the backup operation is no longer helpful. Businesses attempt to back up or snapshot their data regularly, such as every 24 hours. During these instances, duplicates of the primary data are created and stored on a new disk or tape. For disaster recovery, a backup copy must be moved or copied offshore to assure the safety of the data. Backup and recovery allow businesses to safeguard and preserve their data. Thus, the data is divided into small chunks for backup and distributed among storage systems. These blocks are duplicated on distinct nodes, and clusters using replication [21] policies and replication factors.

1.1.3 Why Data Deduplication?

The strategic removal of redundant data from storage considerably reduces a storage system's time and space requirements. Hence, Data Deduplication is required. Cloud Computing provides users with on-demand, low-cost computing, and storage services via the Internet. It offers scalability and flexibility in terms of capacity and performance. The volume of enterprise data is rising as organizations acquire and retain more information. Most data in storage consists of duplicated data, i.e., actual data and the number of copies. The data is duplicated for availability, localization, and improved throughput but causes system overheads. Therefore, Data Deduplication techniques can increase storage efficiency, reduce the cost of data management, and decrease the bandwidth required for data transfer.

Also, scalable deduplication techniques in large-scale storage systems, such as those in cloud storage are designed to handle vast amounts of data and adapt to the growing storage demands. Scalable deduplication techniques typically incorporate distributed algorithms, efficient data indexing, and load-balancing mechanisms to work effectively in large and dynamic storage environments. Scalable deduplication techniques are crucial in maintaining the cost-effectiveness and performance of cloud storage services, where data growth is continuous and substantial.

1.1.4 How Data Deduplication works?

Figure 1.2 illustrates process of data deduplication which reduces redundant data segments to distinct data segments. The entire file is divided into chunks that can be fixed or variable. Only one copy of each segment is kept during deduplication, and pointers are used for duplicate segments. If the deduplication engine discovers a piece of data already in the storage system, it stores a pointer to the original copy. So, it helps to free up storage space in the storage system. Figure 1.2 illustrates the deduplication process.

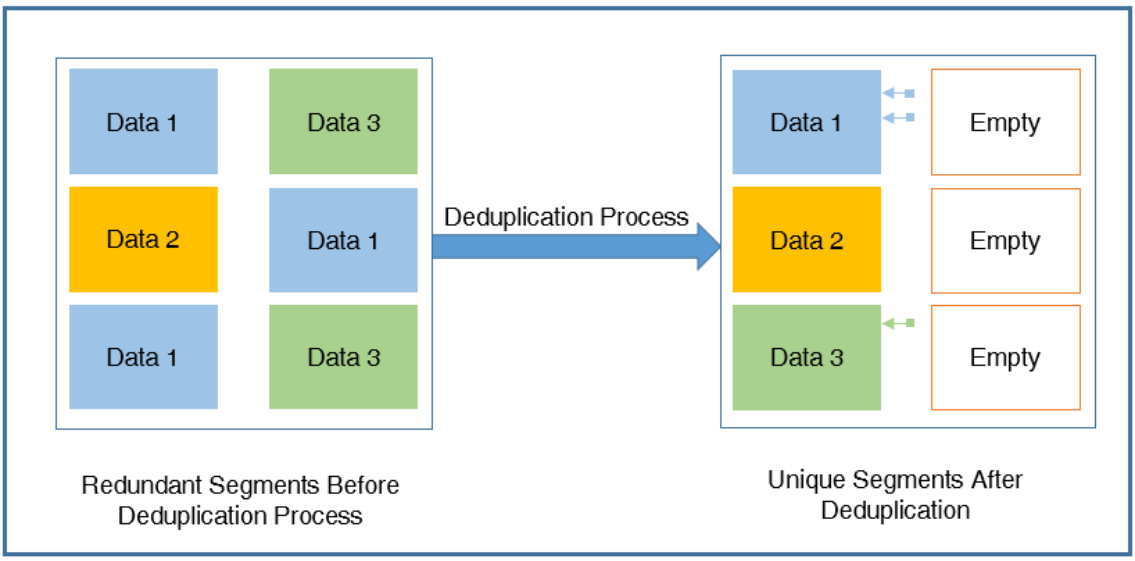


Figure 1.2: How Data Deduplication Works

The file is processed for deduplication and divided into fixed or variable-size blocks. Data deduplication compares objects and eliminates duplicates from a data set. The deduplication procedure removes unique blocks. The deduplication process generally involves four steps:

- First, a hash value is created for each block of data.
- Next, the hash values are compared.
- If the hash values are identical, the duplicate data is replaced with a pointer to the existing objects in the database.
- When data is chunked, an index can be established, and duplicates can be identified and removed [22].

As depicted in Figure 1.3, the generic data deduplication process comprises data chunking, fingerprinting calculation, index lookup, and chunk storage. Index lookup is crucial for locating duplicate chunks. In Figure 1.3, the file is processed for deduplication and is first divided into fixed or variable-size blocks. Data deduplication compares and eliminates identical fingerprint blocks. The unique block is stored, and the index is modified. Figure 1.4 demonstrates the operation of deduplication when a file is broken into chunks and its hash values are generated.

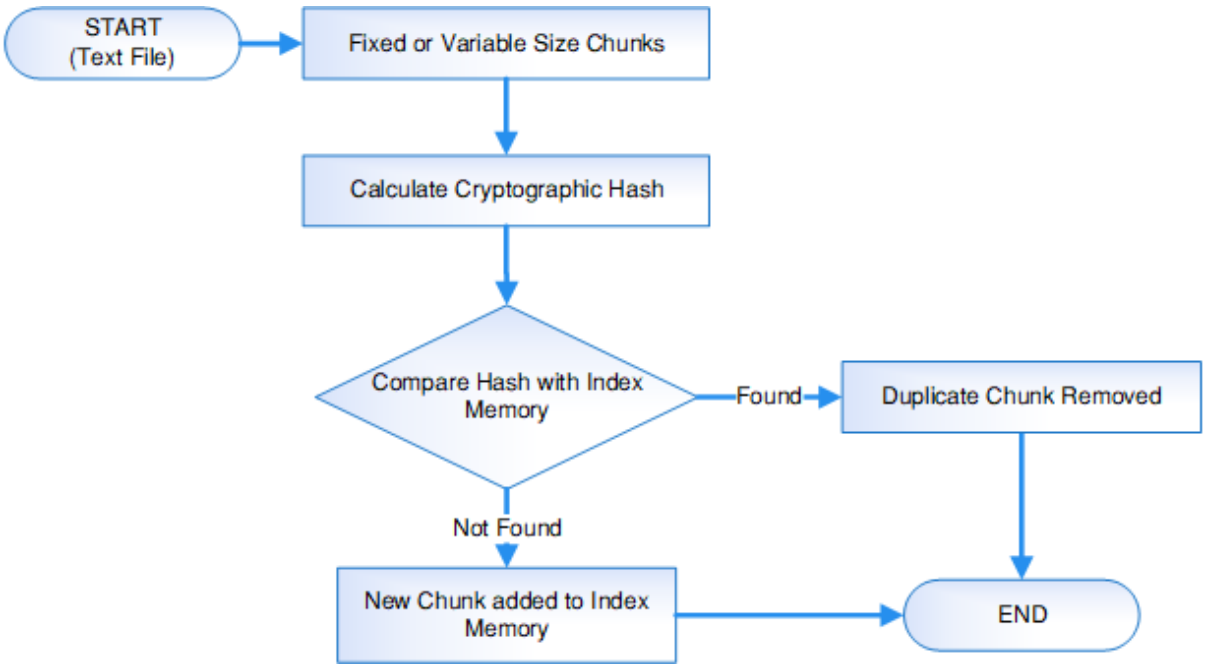


Figure 1.3: Steps of Generic Data Deduplication Process

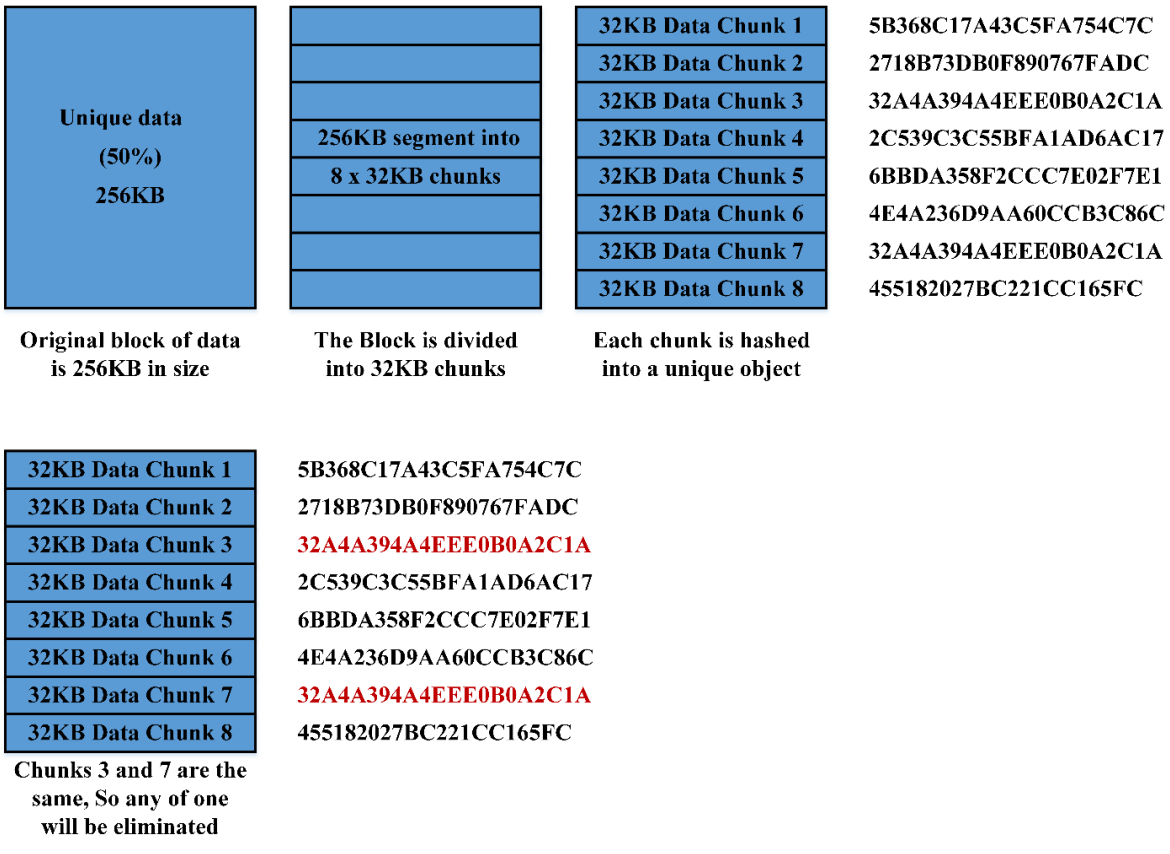


Figure 1.4: Deduplication Chunking Technique

1.2 Classification of Data Deduplication Techniques

The existing deduplication techniques have been classified based on storage type i.e. primary and secondary deduplication; source and target-based; processing time i.e. inline and post-process deduplication, based on level i.e. local and global level deduplication. Cloud-based deduplication is also discussed as deduplication is widely used in cloud storage systems. Figure 1.5 represents the taxonomy of Data Deduplication categorization based on four parameters like Storage-based, Type-based, Timing-based, and Level-based.

These are further classified as following [23]:

- Storage-based - Primary and Secondary storage-based Deduplication
- Type-based - Source and Target Level-based Deduplication
- Timing-based - Inline and Post process-based Deduplication
- Level-based - Local and Distributed or Global based Deduplication

1.2.1 Storage-based Deduplication

Deduplication has been categorized according to storage type. Deduplication is applied to primary [24] or secondary storage [25] as discussed below:

- Primary storage: Primary storage-based deduplication operates on main memory or active storage directly accessible to the CPU(Central Processing Unit). The CPU executes instructions as needed. Primary storage-based deduplication is primarily utilized for latency-sensitive primary workloads [24]. The mail servers' in-memory data is an example of primary storage.
- Secondary Storage: It is an external storage system [25] with no direct access to the CPU. It stores backups of primary storage data for data protection and recovery. These databases are accessible mainly for data archiving and retrieval. Archival storage, snapshots, and backups are all good examples.

1.2.2 Type-based Deduplication

This deduplication process is executed either on the source or target. Based on these two types, deduplication is characterized as Source-based and Target-based.

- Source-based Deduplication: Before being sent to the backup target, the data at the source side is completely deduplicated [26]. Before sending data to the backup

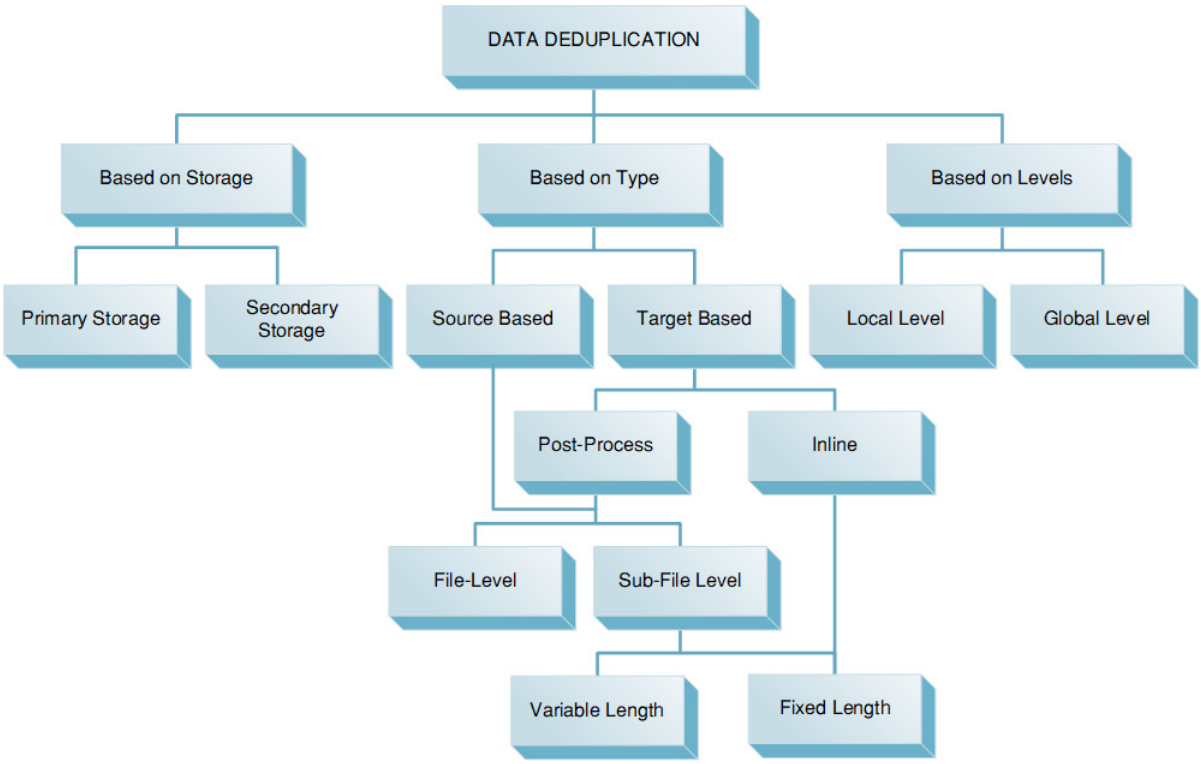


Figure 1.5: Taxonomy of Data Deduplication Techniques

server, the software installed on the servers uses the CPU and memory of the source side and checks for duplicates. As a result, it also decreases the bandwidth, storage, and time needed to back up data. At the same time, it consumes additional CPU and I/O resources to detect duplicates.

- Target Side Deduplication: Deduplication is carried out on backup servers on the targeted storage device [23]. Specialized hardware deduplication appliances manage all deduplication tasks [26]. It improves storage utilization with the additional cost of dedicated appliances. It does not impact the data source and is used for large storage systems. It is further categorized as inline or post-process and demands more network resources, as discussed in the next section.

1.2.3 Timing-based Deduplication

Timing-based deduplication refers to the time at which the deduplication technique is applied. It limits the amount of time required to execute deduplication [12]. The crucial strategy in timing-based deduplication is deduplication processes such as searching for duplicates. It can be performed synchronously/in-band or asynchronously/out-of-band. Timing-based deduplication is subdivided into Inline deduplication and Post-Process deduplication discussed below:

- Inline Deduplication: The data is deduplicated at the source or before being written to disk [26]. Therefore, there is no need for additional disk space to hold and protect the data to be backed up. As a result, it boosts efficiency because the data is only transmitted and processed once. Inline deduplication demands additional computation.
- Post-Process Deduplication: Deduplication occurs after backup data is temporarily written to a storage system, such as a disk. Also referred to as offline deduplication [26]. It is typically faster than inline deduplication since it reduces backup time.

1.2.4 Level-based Deduplication

As described below, Data Deduplication can also be characterized as local level-based or global level-based.

- Local Level Deduplication: Local data deduplication is supported at the Local Area Network (LAN) level. Local deduplication only applies to a single virtual machine and finds duplicates on a single node. It has a negative impact on performance as it cannot completely remove all the duplicates [27]. It offers marginally higher speed in multi-node deployments because it can exploit parallelism and indexing with a high number of nodes and also maintains data availability.
- Global level Deduplication: Common File Elimination, or global-level deduplication, is done across numerous data sets in a distributed environment. It is also known as multi-node deduplication and consists of a cluster of several nodes that function as a single entity. The data sent to one node in the cluster is compared to data previously sent to the same appliance and any other cluster node. The primary objective is to implement deduplication on distributed storage that uses multiple storage servers. It reduces duplicate disk accesses and possibly virtual machine replicas. Moreover, it incurs additional hashing costs [27]. The main goal is to apply deduplication on distributed storage that uses multiple storage servers.

1.2.5 Cloud-based Deduplication on Storage System

Data Deduplication has been widely used in cloud storage [28, 29] environments for backup [29] and archive storage systems because it reduces storage space needs and costs. Deduplication reduces internet bandwidth over the network or the quantity of data transferred to the cloud by storing only a single physical copy instead of redundant data copies. It contributes to the acceleration of cloud backup [29], resulting in faster and more effective data protection operations. Deduplication to Cloud Storage can be configured via Direct Deduplication to Cloud, Deduplication to Cloud on Secondary

Storage Copies, or Deduplication via Cloud Gateway. Additionally, deduplication can be used in various storage systems, from primary to secondary, virtual machines [28] to cloud storage. Private, public, and hybrid cloud storage systems leverage the benefits of deduplication.

1.3 Merits and Demerits of Data Deduplication

Data Deduplication offers significant storage system benefits. These techniques require resources to employ and draw benefits. The significant advantages and disadvantages of deduplication approaches are as follows:

1.3.1 Merits of Data Deduplication

i. Reduce Storage Space: Deduplication reduces the storage capacity for backups, files, and other data applications. As just one copy of the data is stored and duplicates are eliminated[12]. Therefore, it creates more free space to store additional data.

ii. Improves Network Bandwidth: As the unique copies are written to disk, logical pointers for duplicate data are created. As a result, sending duplicates over the network is unnecessary. Deduplication helps to reduce network bandwidth requirements [30].

iii. Reduce Energy Consumption: Deduplication can minimize IO access and storage demand, hence decreasing energy usage. As a result of the decrease in storage space caused by deduplication, additional disks are released and can be placed in a power-saving condition such as standby, spin-down, or off, hence reducing the overall energy usage.

iv. Reduce Storage Cost: Deduplication results in substantial time, space[30], network bandwidth, human resources, and financial savings. It increases the effectiveness and efficiency of the storage system.

1.3.2 Demerits of Data Deduplication

i. Impact on storage performance: In primary storage systems, a fixed-size approach stores many chunks at various memory locations. It causes fragmentation issues that negatively affect performance. The execution of the deduplication technique demands more CPU, memory, and bandwidth resources. So, any inefficient deduplication technique negatively affects the performance of an extensive storage system.

ii. Loss of data integrity: The data blocks are indexed through hash values for better lookup. The identical hashes can be generated for different data blocks due to hash collision, resulting in a loss of data integrity. Hence, hash collisions must be handled carefully to prevent data loss and integrity compromise.

iii. Backup appliance issues: Data deduplication may necessitate a separate hardware device to transfer and process data. Such backup equipment may incur additional expenses and impact storage performance.

iv. Privacy and Security: The entire storage is available to the deduplication techniques. It can be used to gain full access to the storage. Deduplication security measures should be properly planned to protect systems from security breaches and the loss of sensitive data.

v. Reduced Availability: The data is replicated to enhance the high availability of a vast distributed storage system. Any reduction of these duplicates will affect the system's accessibility. A minimum number of copies must be retained to ensure the high availability of a storage system.

1.4 Research Motivation

This work is motivated by the fact that data deduplication [31] is the essential step that must be taken to reduce storage space needs [32, 33, 34] and contribute to a greener environment. Deploying a deduplication strategy in a storage system is not free, and it can severely impact system performance by incurring additional computation and I/O overheads. Thus, the primary focus is on novel distributed data deduplication techniques for large-scale data storage systems that reduce space, bandwidth, the number of disks used in operations, and disk energy consumption costs [35, 36]. Following a comprehensive analysis of data deduplication techniques and research gaps, it has been observed that multimedia deduplication, particularly image-based deduplication in large distributed storage systems, is a potential field for further research. Extensive research has been done on text-based deduplication techniques; however, research on image-based deduplication techniques has not been addressed to a large extent. This issue must be addressed as many duplicate images have been stored in the cloud storage system due to the exponential growth of digital data like Twitter, Facebook, Instagram, etc. Also, duplicate images need massive storage, affecting storage system performance and cost[37].
The real-time duplicate image detection and subsequent removal of duplicate images is a major challenge in cloud-based storage systems as it requires excessive computation and memory. The problems analyzed above necessitate devising an efficient real-time

duplicate image detection technique to detect duplicates from the cloud storage system, which helps improve storage utilization and reduce storage costs.

1.5 Thesis Statement

Data deduplication has been applied to diverse datasets, encompassing text, images, and video content. However, there's a notable absence of a comprehensive and systematic review of existing data deduplication methods. Such a review should thoroughly assess and evaluate the various techniques employed in this field, with a particular emphasis on text and multimedia deduplication, particularly image-based methods.

In the field of online data deduplication within cloud storage systems, limited attention has been given to this area, despite extensive research in the field. One significant challenge is real-time duplicate image detection and subsequent removal of exact or near-exact duplicates, which demands substantial computational and memory resources. To address this challenge in cloud-based storage, an imperative solution is the deployment of a Deep Convolutional Neural Network (CNN)-based online image deduplication technique.

Deduplication techniques play a pivotal role in enhancing the cost-effectiveness and resource efficiency of storage systems by eliminating or minimizing data redundancy. This optimization approach not only conserves energy but also reduces the expenses associated with storing and processing extensive datasets. Handling deduplication on such a massive scale is intricate and challenging. Consequently, there is a notable absence of analyses addressing the impact of these scalable deduplication techniques on energy consumption, cost savings, and system performance. Bridging this gap by devising a common analytical framework is vital for assessing the influence of deduplication techniques on energy savings and cost efficiencies within storage systems.

1.6 Thesis Objectives

The objectives of the proposed work are as follows:

1. To study and analyze the existing deduplication techniques in large-scale storage systems.
2. To propose and design a deduplication technique for scalable cloud storage services.
3. To optimize the proposed deduplication technique for big data storage, performance enhancement, and energy savings.

4. To test and validate the proposed technique for cloud storage service and big data storage.

To achieve the first objective: A comprehensive investigation has been conducted on various existing data deduplication methods for large storage systems in Chapter 2, analyzing their advantages and disadvantages. Techniques were categorized into storage-based, point of application-based, and level-based approaches, with further classifications for text, image, and video deduplication. The survey primarily focused on text and multimedia deduplication, breaking down text deduplication into file and sub-file levels. Multimedia deduplication techniques were divided into image and video categories, with image deduplication further specified as exact and near-exact methods. Video deduplication was referred to as frame-based. Based on the analysis, deduplication poses many challenges in text and multimedia-based deduplication.

To achieve the second objective: A CNN-based online image deduplication technique has been proposed for a cloud storage system in Chapter 3. The technique identifies exact or near-exact images by extracting features on normal and perturbed cross-domain images. Deep CNN-based online image deduplication identifies exact or near-exact images, whereas Hot Vector Decomposition (HDV) generates image patches to store dissimilar parts of near-exact images. The performance of the proposed method has been evaluated based on image-matching accuracy and the time required to match the images.

To achieve the third objective: A novel EsDeDUP, image deduplication technique for energy-saving has been proposed to detect exact or near-exact images. CNN-based and hash-based image deduplication techniques have been proposed to compute the effectiveness of image deduplication techniques on energy saving and storage reduction in Chapter 4. The proposed EsDeDUP analyzes three levels of energy requirements for the computation of images. The CNN model is developed and analyzed further for image feature extraction. Images are pre-processed using data augmentation techniques such as random rotation and resize.

To achieve the fourth objective: Extensive experiments were performed to validate (i) Deep CNN-based online image deduplication technique for cloud storage systems and (ii) the CNN-based energy-efficient deduplication technique in Chapter 3 and Chapter 4. The experiments are carried out in Matlab, Torch, PyTorch, and Tensorflow. PASCAL VOC and a self-collected image dataset of 70,000 images are used to evaluate the performance of several image detection methods for deduplication. CIFAR-10, CIFAR-100, and SVHN are also used for image query energy estimations. The experimental results of image matching techniques are assessed in terms of time and accuracy and compared.

1.7 Thesis Contributions

The thesis makes the following contributions:

- A comprehensive analysis has been undertaken to examine existing data deduplication techniques in cloud storage systems, their challenges, and future research areas. The necessity of a deduplication technique and its pros and cons have also been investigated.
- One of the most important contributions of this research work is developing a CNN-based online image deduplication technique to recognize exact or near-exact images for a massively distributed cloud storage system. Multi-classifier decision fusion for exact or nearly exact image detection and a Fine-Tuned AlexNet is proposed for online cross-domain image deduplication.
- Another major contribution of this work is the Hot Decomposition Vector (HDV) proposition to extract the efficiency of near-exact images. HDV generates image patches to store dissimilar parts of near-exact images such that they can be reconstructed using a base image. HDV exhibits higher and more stable image-matching accuracy with image deformations with comparatively less computing time.
- This work also contributes significantly by designing EsDeDUP, an image deduplication technique for energy-saving. Using deduplication, the strategy conserves energy in terms of CPU consumption, along with conserving storage space. Consequently, the storage area is optimized, requiring less electricity, coolants, etc.
- Another contribution of this research is evaluating the performance of CNN-based and hash-based image deduplication techniques and their effect on energy consumption and energy saving to extract the exact and near-exact duplicate image on a workload.

1.8 Thesis Organisation

Following the thesis's introduction in this chapter, the remaining chapters are as follows:

Chapter 1: Introduction: Chapter 1 is partially derived from :

- Ravneet Kaur, Inderveer Chana, Jhilik Bhattacharya *"Data deduplication techniques for efficient cloud storage management. A Systematic Review."*, The Journal of Supercomputing, Springer, 74(5):2035-2085, 2020. [SCI, IF 3.3]

The rest of the thesis is organized into the following chapters:

Chapter 2: Literature Survey: This chapter provides a review of the literature on various Data Deduplication techniques. Traditional data compression and deduplication approaches have also been discussed. It further summarizes the current deduplication techniques based on storage, point of application, and level-based. It also provides a detailed survey on existing data deduplication techniques based on text, images, and videos, i.e., based on the data type. The discussion is then extended by incorporating other important energy-saving techniques in the cloud storage system. Chapter 2 is derived from

- Ravneet Kaur, Inderveer Chana, Jhilik Bhattacharya *"Data deduplication techniques for efficient cloud storage management. A Systematic Review."*, The Journal of Supercomputing, Springer, 74(5):2035-2085, 2020. [SCI, IF 3.3]

Chapter 3: Proposed Image based Deduplication Technique: This chapter describes a Deep CNN-based online image deduplication technique for detecting exact and near-exact images in a cloud storage system using normal and perturbed cross-domain images as a blur, noise, compression, illumination fluctuations, etc. The work additionally addresses the storage of near-exact images. Hot Decomposition Vector(HDV) is proposed, and image patches are generated and stored as a transformation matrix in case of near-exact images so that it can be reconstructed using a base image. The performance of HDV is compared with traditional image feature extraction techniques and their combinations. The chapter also describes the experimental details and results of the proposed work obtained from different image datasets. Various performance metrics for detecting duplicate images have been evaluated to validate the performance of the proposed approach. Chapter 3 has been derived from:

- Ravneet Kaur, Jhilik Bhattacharya, Inderveer Chana *"Deep CNN based online image deduplication technique for cloud storage system"*, Multimedia Tools and Applications, Springer, 1-34, 2022. [SCI, IF 3.6]

Chapter 4: Data Deduplication Technique for Energy Saving Optimization: This chapter proposes EsDeDUP, a novel image deduplication technique for energy-saving. The proposed approach analyzes the impact of image deduplication techniques on energy-savings, accuracy, and storage reduction using data augmentation techniques such as resizing, random rotation, and extracting image features using the CNN model. The chapter also evaluated the performance of hash-based i.e., PHash, WHash, AHash, and DHash-based image deduplication techniques and their effect on energy-saving to extract the exact and near-exact duplicate images on a workload. Also, fine-tuned CNN-based

deduplication techniques with latent layers to learn the binary codes and F7 features are employed to detect exact or near-exact images. The proposed latent layer has neurons to learn binary code representations. These are used for fine-grained search and reduce the high dimensional vector comparison during exact and near-exact duplicate image detection. The binary codes extracted from the latent layer can easily be compared using Hashing or Hamming distance methods. The experimental results show that the proposed CNN-based deduplication approaches detect duplicate images with greater accuracy, higher storage reductions, and relatively higher power usage than hash-based deduplication techniques. Chapter 4 has been derived from:

- Ravneet Kaur, Jhilik Bhattacharya, Inderveer Chana, *"EsDeDUP: An Energy-saving image deduplication technique for Scalable Exact or Near-Exact Image Duplicate Detection"*.....[Communicated]

Chapter 5: Conclusions and Future Directions: This chapter highlights the thesis's conclusions and recommends further directions.

Chapter 2

Literature Survey

The previous Chapter presented the evolution of data deduplication and its significance in the cloud storage system. It further describes the need for deduplication and its classification, which might prove to be useful for exploring its various techniques in the cloud storage system. Henceforth, it motivates the research and objectives and culminates with the discernment of the contributions and the organization of the rest of the thesis.

This Chapter is intended to review the literature, starting with an overview of the current situation in the realm of data reduction approaches. Furthermore, a comparative study of the state-of-the-art methodologies for data deduplication has been conducted. Existing work on data deduplication has been classified according to several factors, such as data reduction methods, deduplication techniques based on the data type, energy-saving-based deduplication techniques, etc. Further, the chapter traverses through an extensive survey of various existing deduplication techniques based on text and multimedia-based data, with in-depth learning of image-based deduplication techniques.

This Chapter begins with outlining the current scenario of data deduplication in section 2.1. This is followed by data reduction techniques in section 2.2 and the classification of data deduplication techniques in section 2.3, respectively. Text-based deduplication techniques are discussed in section 2.4. Further, Multimedia based deduplication techniques including the structure and components of image-based deduplication are discussed in section 2.5. Section 2.6 discusses key findings of Multimedia based deduplication techniques, followed by Energy-saving techniques in Section 2.7. The chapter concludes with a discussion in section 2.8

2.1 Data Deduplication: State of Art

With the rapid growth of data [38], the term "big data" emerged, and it is now mostly used to characterize huge databases. It often comprises unstructured data that requires more real-time analysis than traditional datasets. The frequency with which data is created is critical and a significant problem. This chapter summarizes existing data deduplication approaches based on text, image, and video data.

2.1.1 Data Deduplication Applications

As data grows dramatically in Cloud storage [39, 40] services, so does the need for more storage capacity. This data is stored on a distributed storage system to ensure high availability and disaster recovery. The replication factor, which refers to the minimum number of data replications, is crucial for disaster protection and system availability. Any number that exceeds the replication factor must be deleted from the storage system. Otherwise, duplicate data places additional demands on the storage system regarding space and bandwidth [41]. As a result, researchers are concentrating their efforts on developing efficient deduplication solutions for storage systems.
Deduplication is a technique for removing duplicate data from storage systems automatically. Deduplication techniques make the storage system more cost- and resource-effective by eliminating or reducing data duplication. This optimization technique also reduces the energy and funding needed to store and process large-scale data. The deduplication process for such a massive dataset is challenging and complex. Thus, analyzing the energy consumption, savings, and performance achieved by such scalable deduplication techniques is important. Deploying data deduplication techniques depends on the data type, including structured, unstructured, and semi-structured data.
Table 2.1 lists existing data deduplication applications used in different Storage Systems, their significance, advantages, and tools and techniques. The different storage systems are categorized into Primary Storage Systems, Secondary Storage Systems, Virtual Machine Systems, Network Systems, SSD(Solid State Drive)-based Multimedia Systems, and Cloud Storage Systems.

2.2 Data Reduction Techniques

The first redundant data reduction techniques appeared in the 1950s [17] as lossless and lossy data compression techniques, followed by delta compression in the 1990s. In 2000, data deduplication techniques were introduced, followed by multimedia deduplication in 2011. These techniques are discussed in the following sections.

Table 2.1: Deduplication Applications Used in Different Storage Systems

Deduplication-Categories	**Significance**	**Deduplication-Advantages**	**Tools and Techniques**
Primary Storage System	Main or active storage system used for primary workloads.	Reduce primary storage space and cost, improve storage energy efficiency	iDedup, SDFS, Ocarina, Permabit, ZFS, POD
Secondary Storage System	Auxiliary storage system, infrequently accessed.	Reduce secondary/backup storage space and cost.	Sparse Indexing, DDFS, HydraStor, RevDedup, Silo, DEDE
Virtual Machine System	Virtual Storage and processing through VMs.	Virtual machine storage efficiency and reduce VM migration time	Liquid, HOPE, VMware ESX , VMflock, DEDE
Network System	Distributed storage and caching for Wide Area Network (WAN) storage optimization.	Reduce time to store and process network storage on WAN	SmartRE, EndRE
SSD-based Multimedia Storage System	SSD Fast storage and access to multimedia content.	Reduce storage space and cost of Solid State Device(SSD), make storage energy efficient	ViDedup, Nitro, UQLIPS, CAFTL
Cloud Storage Systems	Cloud Storage provides access to private, public, and a combination of both users.	Improves Cloud storage efficiency and cost and reduce bandwidth utilization.	Cumulus, NED, SAR, CABdedupe

2.2.1 Redundant Data Reduction Techniques

Several redundant data reduction methods have been developed to manage the rapidly expanding digital data. These techniques identify redundancy from byte to string and chunk to file levels. The organization of redundant data reduction techniques and their evolution are shown in Figure 2.1.

Data compression is a bit-rate reduction approach that represents data compactly. It minimizes storage space requirements and searches for redundant data. Data compression algorithms are widely categorized as lossless or lossy compression techniques[17] [18]. Lossless compression reconstructs original data from compressed data. Lossy compression removes unnecessary data and reconstructs an approximation of the original data. Videos and audios use lossy compression techniques [18]. This section provides background on redundant data reduction strategies, including the evolution of lossless data compression, delta compression, and data deduplication. The taxonomy and evolution of all techniques are described below:

2.2.1.1 Lossless Data Compression Techniques

Claude E. Shannon developed the data compression theory. Lossless data compression techniques include entropy encoding, run-length encoding, and dictionary encoding [42]. A few bit sequences represent the string of characters. In such strings, a substantial amount of redundant data is found and deleted using such data patterns.

- **Byte Level**: Early data compression algorithms used entropy encoding to identify redundancy at the byte level. Entropy encoding techniques like Huffman and arithmetic coding represent frequently occurring patterns with fewer bits. David A. Huffman developed Huffman Coding, which employs a frequency-sorted binary tree to construct the optimum prefix code [18]. It substitutes fixed-length code with variable-length code [18, 43]. More frequently used symbols have shorter encodings [18] than less frequently used symbols. An entire message was encoded into a fixed floating-point number using Arithmetic Coding. A character string is represented by a constant number of bits per character [18, 43, 44]. Less frequently used characters are saved with fewer bits than frequently used characters. Entropy encoding-based techniques have limited scalability and are ineffective for large storage systems.
- **String Level**: A string-level technique was suggested to find and remove repetitive strings. LZ77/LZ78 [17] and LZW/LZO [19, 42] are the two most common byte-level compression methods. LZ77/LZ78, proposed in the 1970s [17] by Lempel and Ziv, are dictionary-level approaches to support sliding windows that identify and delete repeated sets of strings. In the 1980s, Terry Welch introduces LZW/LZO compression versions to improve or optimize the compression process. Figure 2.1 depicts the organization of Redundant Data Reduction Techniques.

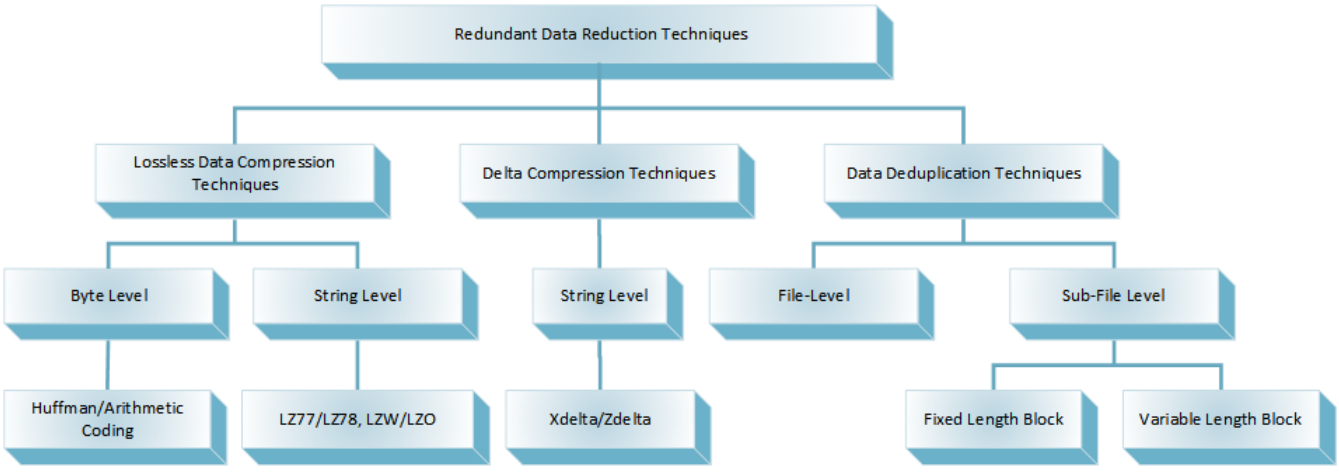


Figure 2.1: Organization of Redundant Data Reduction Techniques

2.2.1.2 Delta Compression Techniques

In the 1990s, the delta compression technique was developed to compress similar files or chunks [17, 19]. Most widely used in applications like remote synchronization and backup storage system. The algorithm employs a sliding window approach at the byte level to identify matching substrings among similar chunks. The differences between sequential files and complete files are represented as "delts" or "diffs" [17, 19]. The string level is one of the delta compression techniques.

- String Level: Xdelta and Zdelta are delta compression algorithms that employ byte-wise sliding windows [19] to detect identical strings between the source chunk and target chunk to calculate the delta. However, this method is extremely time-consuming and not scalable.

2.2.1.3 Data Deduplication Techniques

Data deduplication was proposed in 2000 to support coarse-grained global compression [17]. The earlier methods are not scalable and need considerable time to find similar chunks. However, data deduplication can be applied to the file or sub-file level. It compresses data by using fixed or variable-sized chunks. The hash values of these chunks are generated using cryptographic hash functions, and duplicates are detected by matching hash values.

- **File Level Deduplication**: These techniques are used at the file level, and each file is treated as a separate entity. The attributes stored in the file are compared to those in the backup file index [45]. A pointer is added if the file exists; otherwise, the index value is updated. Single Instance Storage saves only one copy of a file. The whole file hashing method is simple to apply here. Since file hash numbers are simple to generate, it requires a relatively small amount of processing power. Changing a few bytes in a file generates a new or different hash value, which requires new storage. This issue with file-level deduplication prompted the development of block-level deduplication methods.

- **Sub-File or Block Level Deduplication**: These techniques divide a file into fixed or variable-size chunks [46]. Similar hashing algorithms, such as MD5, SHA-I, and Rabin fingerprinting, detect identical blocks. Therefore, a unique block is written to the disc, and its index is modified. Otherwise, a pointer is added to the original location of the same data block. It requires more processing power because the number of identifiers that need to be processed increases significantly. Block-level deduplication is further categorized as fixed-length or variable-length

deduplication.

- **Fixed-Length Block Deduplication**: Fixed-length approaches examine data blocks with a predetermined length. Files are divided into fixed-size blocks [46], and the same data block isn't backed up twice. The primary benefit of this method is its simplicity. A single character insertion in data necessitates a one-byte shift. All subsequent data blocks are again backed up. This limitation of fixed-length block approaches inspires the development of variable-length methods.
- **Variable-Length Block Deduplication**: The deduplication method divides the file into variable-sized data blocks. Variable block algorithms employ a variety of methods for determining the block length. This allows the boundaries of data blocks to "float" inside the data stream, such that changes to one portion of a block do not affect the boundaries of adjacent blocks [46]. The file is partitioned in a content-dependent manner, with segments of arbitrary bytes within a specified range. It offers better control over granularity and flexibility for inserting data in blocks.

2.2.2 Comparison of Lossless Compression, Delta Compression, and Data Deduplication

Table 2.2 compares redundant data reduction techniques based on target data, granularity level, approaches, scalability, evolution, and processing time.

Table 2.2: Comparison of Redundant Data Reduction Technologies

Chunking Techniques	**Lossless Data Compression**	**Delta Compression**	**Data Deduplication**
Target Data	All Data	Similar Data	Duplicate Data
Granularity	Byte or String Level	String Level	Chunk or File Level
Approaches	Huffman coding,Dictionary coding	KMP based Copy and Insert	Content defined chunking, Hashing or fingerprint
Scalability	Weak	Weak	Strong
Evolution Year	1950s	1990s	1990s
Processing Time	High processing time Huffman/dictionary coding	Optimized by using Rabin–Karp string matching	Less processing time as file or sub-file level deduplication is applied.

Since lossless compression, delta compression, and data deduplication target different data types for redundancy, they employ distinct data reduction techniques.

2.3 Classification of Data Deduplication Techniques

Data deduplication techniques are specific to each data type. Text, images, and videos are the primary data types with distinct storage formats and implicit characteristics. The information format is essential for locating the matched data. Therefore, the data type is a crucial consideration while developing deduplication techniques. Data Deduplication techniques are currently a prominent area of research, with researchers dedicating their time and attention to efficiently applying these techniques to remove duplicates or redundant data. Text, images, and videos are highly redundant [47] on the Internet. Such redundancy has increased since the introduction of social networking platforms and exerts an additional strain on the cloud storage systems [47]. Finding duplicate data on such a large heterogeneous platform is a huge challenge for researchers in industry and academia.

Figure 2.2 illustrates deduplication techniques based on the data type. Text data deduplication is categorized into file level and sub-file level. Multimedia deduplication techniques have been categorized as image and video deduplication. Image-based deduplication is further categorized as exact and near-exact image deduplication. Video deduplication can be classified as frame-based deduplication.

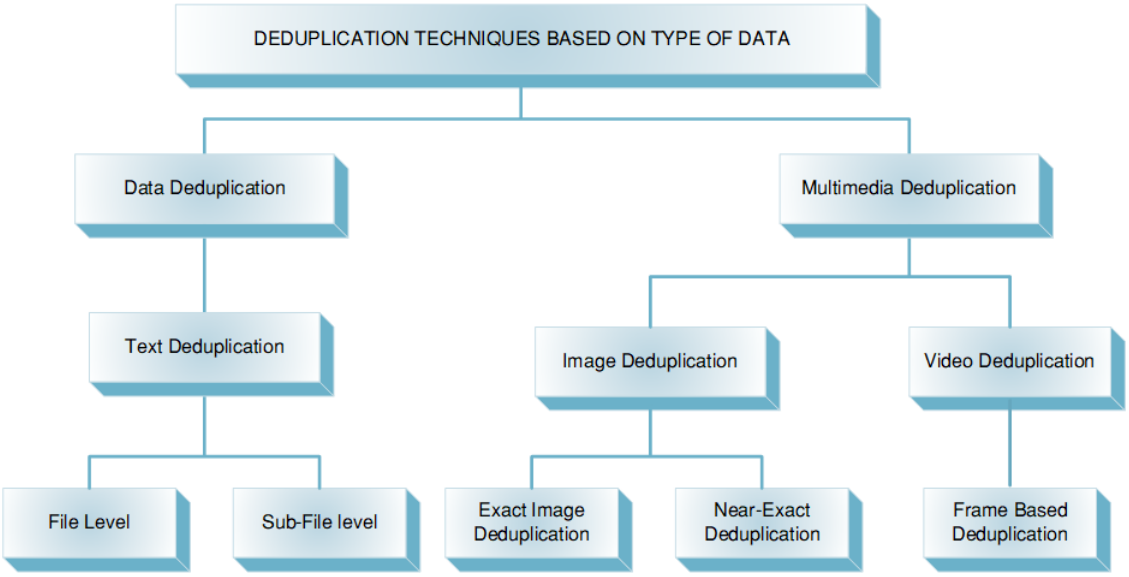


Figure 2.2: Classification of Deduplication Techniques

2.3.1 Data or Text Deduplication

In text-based deduplication techniques, exact text matches are determined through byte-by-byte comparison, and duplicates are found. It operates at both file and sub-file levels. File-level deduplication, or single instance storage, eliminates duplicate files at the file level. Blocks at the sub-file level, also known as block-level deduplication, can be either fixed or variable in size. The fixed-length method examines data chunks of a predetermined length. It separates the files into fixed-size blocks [46]. Varied-Length Deduplication breaks the file into data chunks of variable length.

2.3.2 Multimedia Deduplication

Image-based deduplication and video-based deduplication are subcategories of multimedia deduplication. Image deduplication techniques are based on image detection techniques and categorized into exact and near-exact image detection. Exact image deduplication approaches are based on exact duplicate images and do not consider image transformation. Near-exact images are modified or copied versions of the original image. The images are altered by cropping, modifying, scaling, adding noise, compressing, rotating, etc. There are some strategies and accuracies for identifying such duplicates.
Video deduplication employs frame-based deduplication algorithms. First, a video is divided into frames (or images) representing the visual features [48, 49]. Each video keyframe generates a hash function based on visual descriptors to detect duplicate sequences between queried video and the video library [48].

2.3.3 Comparison of Text-based Deduplication and Image-based Deduplication

Text-based deduplication techniques are compared to image-based deduplication techniques based on partitioning methods, indexing or hashing techniques, lookup methods, storage format, matching strategy, and accuracy. As video is split into frames, image-based deduplication methods can be utilized because video and images are based on visual content. Table 2.3 illustrates the parameters and comparative descriptions for text-based and image-based deduplication. This table has been modified and improved based on tabular data in [50].

Table 2.3: Comparison between Text and Image Based Deduplication

Parameters	Text Based Deduplication	Image Based Deduplication
Partition Method	Data is divided into chunks of varying sizes.	The image is preprocessed, and features are extracted.
Techniques Used	Cryptographic hash functions calculate hashes of various chunks.	Different feature extraction approaches to extract features and hashing techniques on image features.
Index Lookup	It uses an exact match for index lookup to find duplicate files or chunks of data in storage systems.	Similar images require an approximate match for index lookup to be matched.
Storage	A file is only stored once, and duplicates are eliminated.	Centroid selection is made, centroid-image is stored, and the near-exact image transformation is stored as a transformation matrix.
Matching Technique	A byte-by-byte comparison is made to find an exact text match, and duplicates are found.	Duplicate images are identified by comparing the number of identical elements extracted from images.
Accuracy	Complete matching is done here.	Exact or near exact images are detected here.

2.4 Text Based Deduplication Techniques

Several studies have been reported on text-based deduplication techniques for different storage systems, such as backups and secondary storage systems. Numerous authors have discussed text-based deduplication techniques, which are broadly categorized based on granularity, locality, indexing, security, and cloud. Figure 2.3 presents a taxonomy of text-based deduplication techniques.

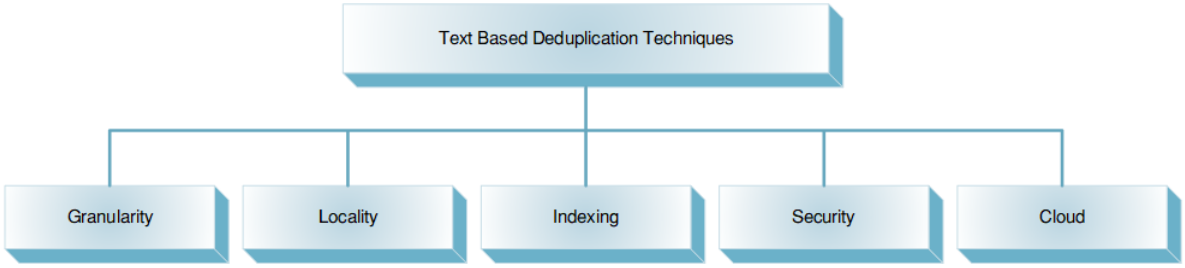


Figure 2.3: Taxonomy of Text Based Deduplication Techniques

2.4.1 Taxonomy of Text Deduplication

This section presents a taxonomy that categorizes deduplication systems based on granularity, locality, indexing, security, and cloud. This taxonomy defines and characterizes the distinct methodologies employed for each approach. Granularity divides data into

chunks and eliminates duplicates. The file and sub-file are two levels of granularity. File-level granularity, often called object granularity, deduplicates files. A file is split into chunks or blocks of fixed or variable size using the sub-file level granularity [12]. Sub-file variable chunks improve matching. Duplicate detection is faster and more accurate with finer granular chunks[51]. The storage systems also exploit locality in caching strategies and on-disk layout [12]. Storage systems use two types of locality, namely temporal and spatial. In temporal locality, chunks are referenced at a specific memory location during a particular moment. They will likely be referenced in the same memory place in the near future [12]. Therefore, duplicate chunks appear multiple times within a short time span. Memory locations with adjacent addresses are referenced in close succession in the spatial locality. Spatial locality refers to a certain memory location that is referenced at a specific time; thus, its nearby memory locations will likely be referenced soon.
Indexing provides an efficient data structure to look up duplicated data [12]. Hashing summarises the content and identifies signatures to find exact duplicates. Hashing computation needs additional CPU resources and hash collision avoidance methods. Collisions can be avoided by comparing the contents of the two chunks with similar signatures. Rabin's fingerprint [52] is another technique used to compare the similarity of two chunks. The security of cloud storage systems is critical. The deduplication technique requires authorization and authentication to access the entire storage system. A security framework is necessary for deduplication to thwart attackers and stop system invasions. Cloud storage security includes data sharing, confidentiality, integrity, avoiding data leaks, and offsite storage [53]. Data Deduplication is also extensively used in cloud-based storage systems to increase storage efficiency and reduce costs. Network bandwidth, high throughput, computational overhead, deduplication efficiency, and energy usage are the main challenges for cloud-based data deduplication. Table 2.4 presents a taxonomy of text deduplication techniques based on Granularity, Locality, Indexing, Security, Cloud, their description, and important findings.

Table 2.4: Taxonomy of Text Deduplication Based on Granularity, Locality, Indexing, Security, Cloud

Author(s)	**Based-on**	**Technique**	**Description**	**Findings**
Lin L et al., [54], 2023	Granularity	Greedy algorithm	InDe, inline deduplication that examines the valid container utilization	Significant improvement in restore performance

Continued on next page

Table 2.4 – *(Continued)*

Authors	Based-on	Technique	Description	Findings
Venkatesh Babu S et al., [55], 2022	Granularity	Content Deduplication with granularity tweak (CDGT),Hadoop	Dynamic Content Adjustment Policy performs deduplication at chunk level	Improves the storage utilization
Ye X et al., [56], 2021	Granularity	Variable-grained chunk size, Rabin fingerprint, Delta encoding	Fast variable-grained resemblance data deduplication for cloud storage	Reduces the metadata size and high deduplication ratio
Hirsch M et al., [57], 2020	Granularity	Variable-size chunks	Determine chunk boundaries in a deduplication system	Dynamic method to set chunk boundaries
Cao Z et al., [58], 2019	Granularity	CDC(Content Defined Chunking),Adler32	Tier-aware Data Deduplication File System improve deduplication performance	Better performance and space-saving
Zhou Y et al., [59], 2018	Granularity	Fixed-size chunks, Rabin fingerprint, Delta Encoding	Low latency in-line data deduplication decouples unique data block and fingerprints	Enhances the read-and-write performance
Park D et al., [60], 2017	Granularity	Look ahead and read cache	Novel dedupe storage exploits future data chunk access patterns	Fast read performance
Venish A et al., [61], 2016	Granularity	Variable Chunking	Comparison of Different chunking models and algorithms	Performance Comparison of different chunking models
Xia W et al., [62],2015	Granularity	Deduplication-Aware Resemblance Detection and Elimination Scheme (DARE)	Exploits duplicate-adjacency (DupAdj), for efficient resemblance detection in backups	Less computational and indexing overheads, high throughput
Shen GL et al., [63], 2022	Locality	Temporal and Spatial locality	Fast page merging and low CPU overhead memory deduplication	FLOMD outperforms in terms of sharing efficiency

Continued on next page

Table 2.4 – *(Continued)*

Authors	Based-on	Technique	Description	Findings
Zhang Z et al., [64], 2021	Locality	Exploits the similarity and locality, history-aware approach	SLIMSTORE, cloud-based deduplication separates computing and storage to support elastic scaling	Efficient restoration and effective space reduction
Tan Y et al., [65], 2020	Locality	Reference-Count based Eviction Bitmap based Hotness Identification techniques	Content-driven Deduplication-Aware Cache exploits block content redundancy	Significantly improves cache hit ratios
Xu G et al., [66], 2019	Locality	TTTD chunking algorithm,SHA-1,CDS algorithm	Learning-based Indexing and Prefetching Approach(LIPA) exploits both temporal and spatial locality	Better deduplication ratio with little memory overhead
Wu H et al., [67], 2018	Locality	Cache Replacement Algorithm	A novel fingerprint caching mechanism calculates duplicates' temporal proximity	Improves deduplication ratio and overhead reduction.
Fu Y et al., [68], 2017	Locality	Handprinting Based Stateful Data Routing, Application-Aware Routing Algorithm	An application-aware deduplication framework exploits data similarity and locality	High global deduplication efficiency
Wu S, et al., [69], 2016	Locality	Spatial locality	AA-Plus method groups the hash index of the same application together	Improves the write and read throughput
Ahmed ST et al., [70],2022	Indexing	Commonly Repeated Patterns (CRP) chunking algorithm	Lightweight triple-leveled hashing (LT-LH) algorithms reduces the probability of hash collision	Better hashing throughput
Saeed AS et al., [71],2021	Indexing	Multi-level lookup and matching technique	Novel multi-hash function for similarity lookup	Reduce the hashing index size and lookup time.
Hema S et al., [72],2020	Indexing	Distributed Storage Hash Algorithm(DSHA)	Efficient DSHA to lessen the memory space	Improves data read/write performance.

Continued on next page

Table 2.4 – *(Continued)*

Authors	Based-on	Technique	Description	Findings
Xia W et al., [73],2019	Indexing	Fixed-Size and Content-Defined Chunking	P-Dedupe, an efficient scalable deduplication system exploits parallelism in storage systems	Effectively parallelizes the CDC with an increase in throughput
Jasim HA et al., [74],2018	Indexing	Variable size and Two Threshold Two Divisor (TTTD) chunking	Content-based TTTD with Multi-Level Hashing to speed up the deduplication operation	Reduces the deduplication processing time
Zhang P et al., [75],2017	Indexing	Dynamic Bloom filter array(DBA) and Resemblance algorithm	Resemblance and Mergence based deduplication(RMD) speed up the fingerprint index performance	Significantly improves fingerprint query performance
Widodo RN et al., [76],2017	Indexing	AE algorithm	Novel Rapid Asymmetric Maximum chunking algorithm uses bytes to set the cut points	Low computational overhead and high throughput
Kumar N et al., [77],2016	Indexing	Fixed chunking and MD5 hashing algorithm	Bucket-based and MapReduce technique to store hashes in distinct buckets	High deduplication ratio,low chunk lookup time
Song M et al., [78], 2023	Security	Ciphertext-Policy Attribute Encryption (CP-ABE)	Blockchain-based integrity auditing with secure deduplication for cloud data security	Secure and lightweight public auditing and deduplication
Liu S et al., [79], 2023	Security	Block-level deduplication	Novel authentication data structure and blockchain-based compact audit-enabled deduplication scheme	Security and performance analysis prove the scheme's viability
Xie Q et al., [80], 2022	Security	Security-aware data encryption	Security-aware and efficient edge-assisted cloud storage data deduplication	Achieves deduplication efficiency
He Y et al., [81], 2021	Security	Ciphertext policy attribute-based encryption, Secure key delivery algorithm	Secure encrypted data deduplication scheme in the cloud storage system	Secure and efficient.

Continued on next page

Table 2.4 – *(Continued)*

Authors	Based-on	Technique	Description	Findings
Almrezeq N et al., [82], 2021	Security	Asymmetric Encryption Algorithm AES-CBC algorithm	Secure approach that uses traditional secure encryption and supports the deduplication features	Removes redundant data while being secure against attacks
Tian G et al., [83], 2020	Security	Dynamic Key-Encrypting Key tree	Randomized client-side deduplication for stronger ownership authentication	Meets security needs while conserving system resources
Wang L et al., [84], 2019	Security	Convergent encryption algorithm	Proof-of-ownership key-sharing for secure deduplication	Efficient and secure in the proposed security model.
Xiong J et al., [85], 2018	Security	Symmetric encryption, proof of ownership, and bloom filter	RSE-PoW strategy for secure multimedia deduplication	Secure and efficient approach
Sun W et al., [86], 2018	Security	Content-aware deduplication technique	A randomized oblivious key generation mechanism to reduce information leakage	Better deduplication performance and security
Vishalakshi NS et al., [87], 2017	Security	Convergent encryption, Block level deduplication	Meta data manager (MM) takes care of actual deduplication and key management operations	Secure and efficient cloud storage service
Wang Y et al., [88], 2023	Cloud	Ciphertext-policy attribute encryption (CP-ABE)	Secure deduplication using blockchain-based integrity auditing in cloud storage	Secure and efficient storage efficiency
Yu X et al., [89], 2022	Cloud	TDICP and UD-DCP protocol	Novel VeriDedup scheme that guarantees the correctness of duplication check and supports the integrity check	Secure and Efficient
Ma X et al., [90], 2022	Cloud	TDICP and UD-DCP protocol	Novel server-side deduplication scheme for encrypted data in a hybrid cloud architecture	Better performance in terms of security, effectiveness

Continued on next page

Table 2.4 – *(Continued)*

Authors	Based-on	Technique	Description	Findings
Kan G et al., [91], 2021	Cloud	Identity-based Proxy Re-encryption (IB-PRE) and Identity-based Proof of ownership (IB-PoW) Scheme	IB-PRE data deduplication schemes manage the encrypted data storage with deduplication	Security, Efficient and Effective
Zhang Z et al., [64], 2021	Cloud	History-aware Approach	Hybrid deduplication solution deduplicates and restores massive multi-version backups online.	Scalable deduplication and restoration
Zheng X et al., [92], 2020	Cloud	Certificateless Proxy Re-encryption (CL-PRE) and Proof of Ownership (PoW-CLS)	Cloud data deduplication scheme based on certificate-less proxy re-encryption	Improves the efficiency of the proof of ownership (PoW)
PG S et al., [93], 2020	Cloud	Modified Elliptic Curve Cryptography (MECC), Convergent Encryption (CE)	Recognize data redundancy at block level to construct secure deduplication systems	Outperforms in terms of computational efficiency and security levels
Yuan H et al., [94], 2019	Cloud	Re-encryption Deduplication and Rekeying-Aware Encrypted deduplication	Bloom filter-based location selection and secure data deduplication	Secure re-encryption
Hovhannisyan H et al., [45], 2018	Cloud	Message decoding and Synchronization algorithm	A nique deduplication-based covert channel with a new synchronization technique	Highlights more serious security threats in cloud storage providers.
Yuan H et al., [95], 2018	Cloud	Randomized convergent encryption algorithm	A secure and scalable data deduplication scheme with dynamic user management	Reduces the communication overhead
Zheng X et al., [96],2017	Cloud	Secure data deduplication for cloud storage	Survey on existing secure deduplication techniques based on cryptographic and security protocol solutions	Identify data security threats to improve efficienc.

Continued on next page

Table 2.4 – *(Continued)*

Authors	Based-on	Technique	Description	Findings
Poale MS et al., [97], 2017	Cloud	DelayDedupe	Load balancing technique for file server and cloud storage server	Effectively decreases response time and balances storage node load
Neelaveni P et al., [98], 2016	Cloud	File classifier based Linear Indexing Deduplication (FC-LID), Linear Hashing with Representative Group (LHRG)	File classifier based Linear Indexing Deduplication to overcome disk bottleneck problem	Efficient and Less computational overhead

In addition to the classification and discussion of text-based deduplication strategies, this chapter illustrates the evolution of text-based deduplication techniques. It helps researchers identify techniques in their sub-area. Figure 2.4 shows the evolution of five main categories of text-based deduplication approaches.

Also, Figure 2.5 illustrates the percentage of research articles that fall into these five categories. Since granularity has the potential to improve performance, it has been the primary research focus of these studies. The growth of data in the cloud has been a second major focus of researchers followed by security, indexing, and locality.

Extensive research has been done to define the parameters of deduplication techniques according to the taxonomies defined in section 2.4 and Table 2.5. Table 2.5 presents the parameters of deduplication techniques. The parameters of selected techniques are file-level chunking, variable-level chunking, spatial locality, temporal locality, full indexing, partial indexing, sparse indexing, inline method, and offline method based on timing. This table will facilitate researchers' rapid comparison of deduplication techniques using these parameters.

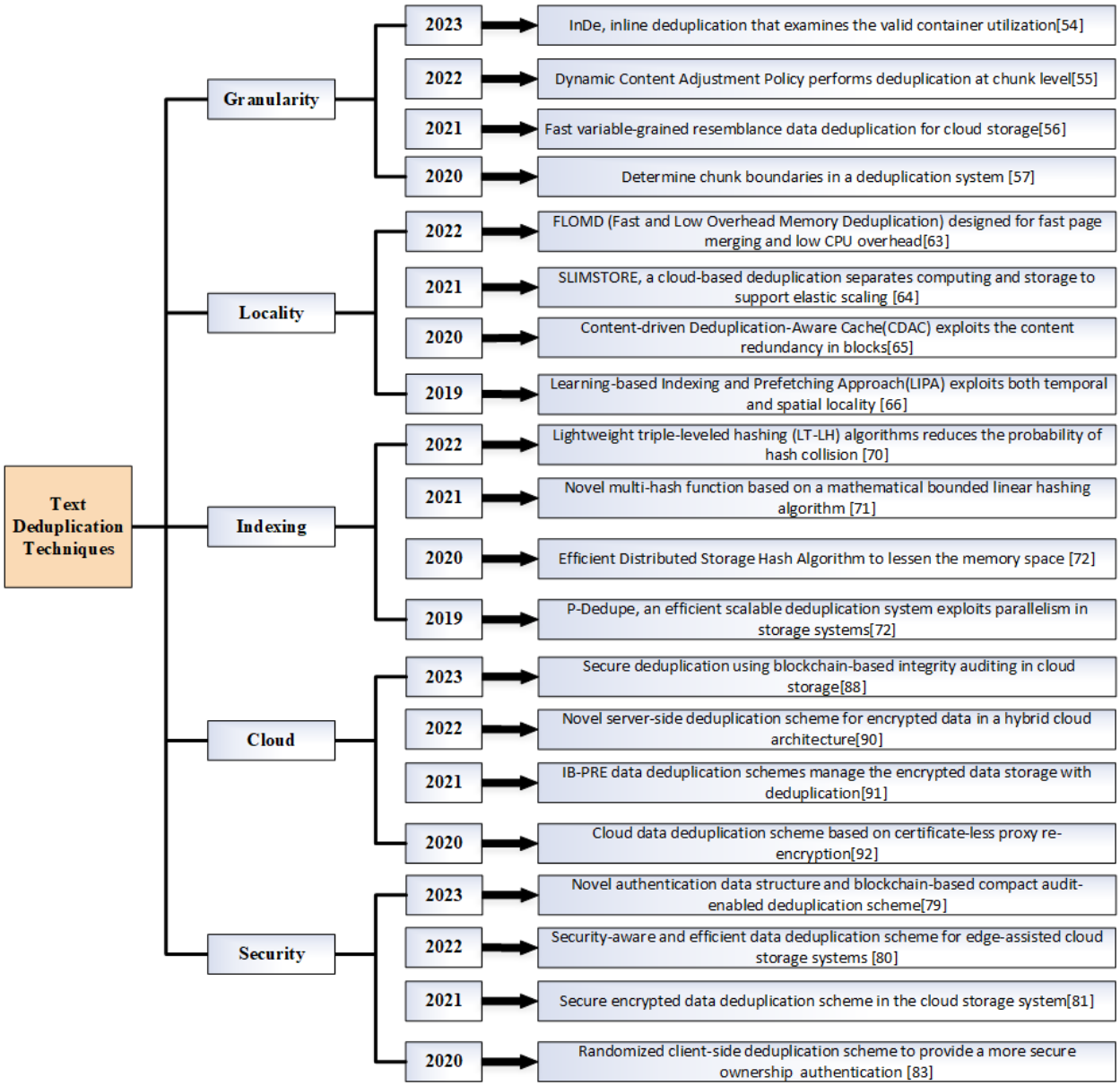


Figure 2.4: Evolution of Text Based Deduplication Techniques

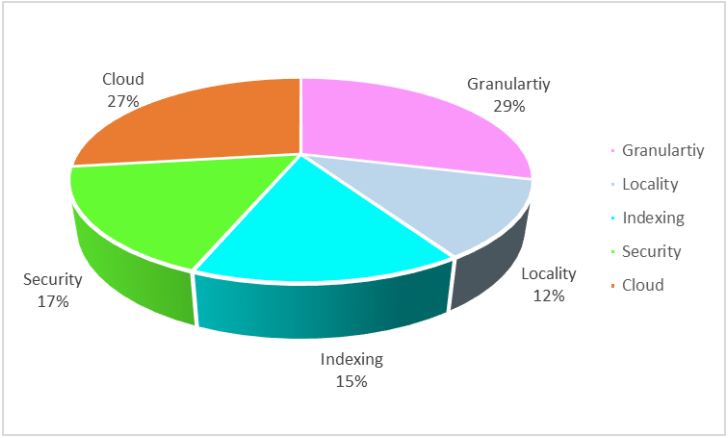


Figure 2.5: Research Contribution of Five Broad Categories of Text-Based Deduplication

Table 2.5: Parameters of Deduplication Techniques

Parameters/Techniques	Fixed Level Chunking	Variable Level Chunking	Spatial Locality	Temporal Locality	Full Indexing	Partial Indexing	Sparse Indexing	Inline Method	Offline Method
[9]	Yes	No	No	No	No	No	Yes	Yes	No
[99]	No	Yes	Yes	No	Yes	No	No	Yes	No
[100]	No	Yes	Yes	No	Yes	No	No	Yes	No
[101]	No	Yes	Yes	No	No	No	Yes	Yes	No
[102]	Yes	No	Yes	No	No	Yes	No	Yes	No
[103]	Yes	Yes	No	Yes	No	No	Yes	No	No
[104]	No	Yes	Yes	No	Yes	No	No	Yes	No
[105]	No	Yes	Yes	No	Yes	No	No	Yes	No
[106]	No	Yes	No	No	Yes	No	No	No	Yes
[107]	Yes	No	Yes	Yes	Yes	No	No	Yes	No
[108]	Yes	No	Yes	Yes	Yes	No	No	Yes	No

2.5 Multimedia Based Deduplication: Structure and Components

Many techniques to detect or extract information (features) from images or videos have been used due to the development and advancement in image and video retrieval systems. Moreover, with the development of the internet, smartphones, and social networking sites, users worldwide share many images and videos [109]. Facebook adds 0.25 billion images daily, whereas Flickr has 6 billion images [110].

Most of the images uploaded are either modified [111], forwarded, or copied [110], resulting in many duplicates or near-duplicate images on the web. This leads to a massive database of duplicate or near-duplicate images on the storage system. These large number of duplicate images are impacting the performance of the image storage system and escalating its cost. In addition, it isn't easy to index or retrieve images efficiently from an extensive image cloud storage system. These duplicate images consume valuable storage space in the storage system [47]. There must be some efficient way to remove these duplicates from the storage system.

Image deduplication is one such technique that helps remove duplicate images from the storage system. A deduplication technique was applied to images to detect similar images and on videos to detect similar frames (or images) from videos using different parameters like feature extraction algorithms, hashing algorithms, and distance measures. Hashing algorithms are used to generate hashes based on features extracted from images. These hashes are then compared to determine the similarity between two images, using a predefined threshold.

Figure 2.6 defines the technique employed to detect exact or near-exact image duplicates. 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. Feature Extraction Techniques, Hashing Techniques, and Matching Distance Measures are the main stages involved in Image-based Deduplication, as shown in Figure 2.6

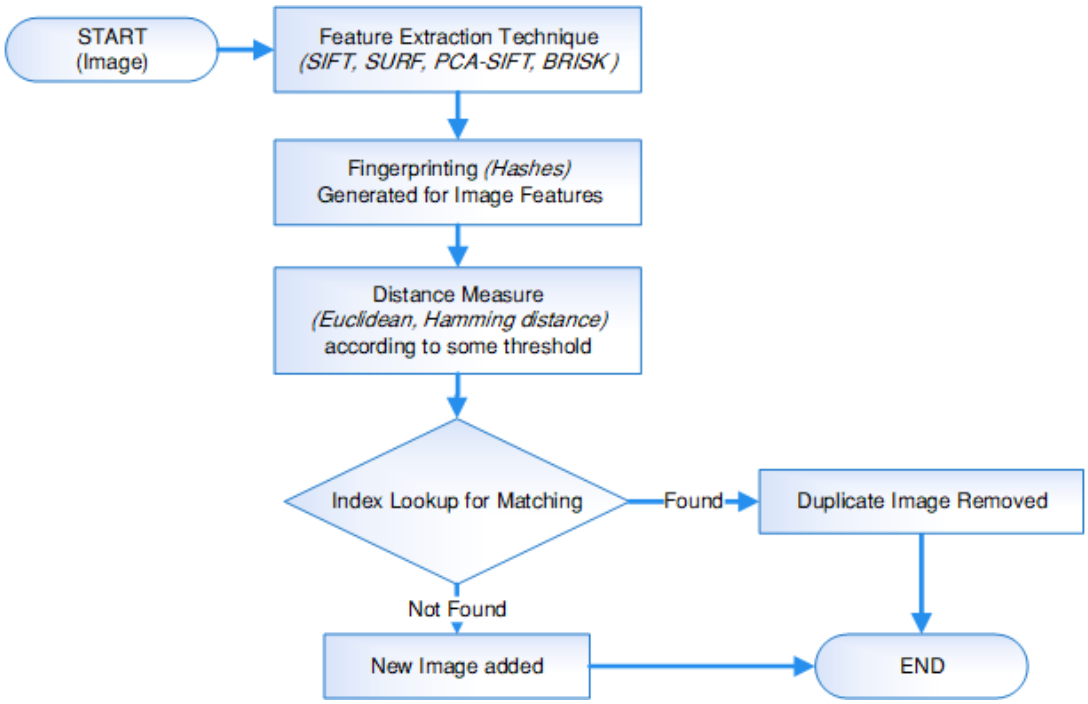


Figure 2.6: Image Deduplication Process

Different image deduplication techniques have been discussed based on their characteristics, like image feature extraction, image hashing algorithms for indexing, and distance measures to detect similarity between images or videos in the next section.

2.5.1 Image Feature Extraction Techniques

Techniques for image deduplication are based on image detection methods, which are further categorized into exact and near-exact image detection. The most fundamental and significant step in image deduplication is image feature extraction. Image Deduplication techniques were applied to images to detect similar images using feature extraction algorithms like SIFT, SURF, PCA-SIFT, and BRISK, etc., hashing algorithms, and distance measures to check the similarity between two images using some threshold. Time and accuracy are the two biggest challenges to detect exact and near-exact images in a cloud storage system.

Basic feature extraction techniques include Speeded Up Robust Features(SURF) [112,

113], Scale-Invariant Feature Transform(SIFT) [114, 115, 116], Binary Robust Invariant Scalable Keypoints(BRISK) [117], Features from Accelerated Segment Test (FAST) [118], Haar[119, 120], Harris [121], Discrete wavelet transform(DWT) [122] or hybrid techniques such as binarized SIFT [123], Local Binary Patterns(LBP) [124], Local-based Binary Representation (LBR) [125], Probabilistic Center-symmetric Local Binary Pattern (PCSLBP) [126], and Dense Scale-Invariant Feature Transform (DSIFT) [127]. Composite feature vectors like Discrete cosine transform (DCT) with SURF, DoG with PCA-SIFT[128, 129] etc. are also used in various research papers to improve performance and computational speed. Also, it helps in substantial space savings.
These can also be replaced with state-of-the-art CNN-based feature extraction techniques for improving accuracy. Convolutional Neural Networks(CNN)-based image detection techniques [130, 131] are evolving and have been reported for various tasks such as object detection [132], face detection [133], identification of plant diseases like tomato [134, 135], and potato [136, 137] leaf diseases. Further, CNN is also used in medical image processing for the detection of various diseases like breast cancer detection [138, 139, 140], lung diseases [141, 142, 143] and Diabetic Retinopathy [144, 145, 146] etc. and their patterns. CNNs have been used as a feature extractor and a classifier for all these cases.
Some of the basic and hybrid feature extraction techniques to detect duplicate images are listed below:

- HAAR: Haar wavelets are rescaled square-shaped functions that comprise a wavelet family [147]. It creates sparse image matrices by averaging and differencing values.
- SIFT: SIFT iS scale and rotation invariant and uses the Difference of Gaussian (DoG) to o identify features of interest across various scales. The DoG is employed to improve the computational speed during the extraction process in an image [114].
- Harris Corner Detector: Harris corner detector is an efficient feature detection that uses the Harris-Stephens algorithm to detect the corners of an image [148, 149] and is robust against rotation, scale, jpeg compression, noise, and blurring.
- SURF: Hessian matrix approximation makes SURF resilient to noise, detection displacements, and geometric and photometric deformations [112]. SURF is efficient in image feature detection and description matching.
- DWT: DWT transforms a discrete-time signal into a discrete wavelet using a time-scale representation. It divides an image into LL, LH, HL, and HH for dimension reduction. SURF features are extracted from the DWT's LL section, assuming the image's LL region has the most information. Extracted key features are utilized to compare descriptor vectors [150].

- DCT: DCT depicts infinite data points as the sum of cosine functions vibrating at different frequencies. DCT helps separate the image into parts of variable relevance. DCT converts a signal or image from spatial to frequency domain [151].
- PCA-SIFT: PCA-SIFT [115] describes each keypoint, and the combination is used to search high dimensional data efficiently [128]. The retrieval procedure finds each descriptor in the input image's nearest neighbor in the database. Due to high computation overhead, scalability is one of the limitations of this technique.
- ASV: Affinity-aware summary vectors use Locality-Sensitive Hashing (LSH) [152] to identify and aggregate correlated images. Interest points are represented by the improved Difference of the Gaussian detector from the original DoG and improved PCA-SIFT. ASV, DoG, and PCA-SIFT [153] reduce backup on-disk index lookups and simplify near-duplicate image identification.
- Binarized SIFT: SIFT features are converted into binary strings to improve SIFT matching speed. Hashing techniques will store binarized features into hash tables to retrieve similar images[154]. The technique used speeds up the image retrieval process and improves matching accuracy.
- PCSLBP: PCSLBP uses variable-length signatures for feature matching using a probabilistic approach to encode pixel intensity difference. Earth Mover's Distance checks image similarity. The technique is flexible even in image distortions [155].
- LBR: A sliding window extracts texture features information from the image and converts them to block-based local binary patterns [125]. The local features can be used to calculate a statistical histogram, which is then encoded as a binary vector to increase efficiency. The technique reduces memory costs and speeds up online computation.
- LBP: LBP compares all neighborhoods with the central pixel to produce an 8-bit binary pattern and is integrated with a basic contrast measure by computing for each neighborhood the difference of the average gray level of pixels with values 1 and 0. LBP is effective due to its monotonic gray scale invariance and minimal processing complexity[156].
- FAST: The corner detection method FAST [157] finds image interest points. It compares pixels solely on a circle with a fixed radius around the point using a 16-pixel circle of radius 3.
- BRISK: BRISK is scale- and rotation-invariant with efficient keypoint detection, description, and matching approach. The grayscale relationship of neighborhood ran-

dom point pairs creates the local image's binary feature descriptor. This descriptor consists of a 512-(64-byte) bit-string and is represented as a 64-dimensional feature vector [158]. BRISK has a faster matching speed and smaller storage memory.

- DSIFT: The DSIFT algorithm derives from SIFT algorithm is a significant keypoint-based method [159]. DSIFT is capable of acquiring more features in a shorter time than SIFT. DSIFT algorithm also explored local averaging and local contrast normalization techniques.
- CNN-based Deep Learning Features: CNN is a feedforward multilayered hierarchical architecture based on convolutional filters stacked across layers. Each layer uses convolutional kernels to perform multiple transformations [160, 161, 162]. Deep CNN's multilayered, hierarchical structure enables it to extract low, mid, and high-level features. CNN applications include image and video recognition [163], recommender systems [164], image classification and segmentation [165], object detection [166], video processing [165], natural language processing [167], medical image processing [168, 169, 146] etc.

2.5.2 Image Hashing Techniques

In computer vision [170], the Hash algorithm can determine the degree of similarity between two images by comparing their fingerprints. The more similar the fingerprint comparison results, the closer the image content is. Many hashing-based image deduplication techniques [171, 172, 173, 174] have been proposed for removing duplicates from image datasets, and these image-hash-based techniques compress images by comparing each pixel value. Such hash values are further used to detect exact and near-exact images. Methods for detecting Hash similarity include average Hash (aHash) [175], difference Hash (dHash) [175], perceptual Hash (pHash) [176] and wHash. Average Hashing (aHash) [177] is a method that creates an image hash by calculating the mean value of the pixels in the image and then comparing each pixel to that mean value. The difference hash (dHash) [177] is a method that creates an image hash by comparing the intensity of adjacent pixels in the image. It converts the image to grayscale and then subtracts each pixel's value from the next pixel's value in a horizontal direction. Perceptual Hash(pHash) [174] is a method that creates a hash of an image based on the visual content of images. It uses the Discrete Cosine Transform (DCT) [178] to extract features from the image and create a robust hash to image transformations such as cropping, scaling, and rotation. Like pHash, the Wavelet Hashing (wHash) [173] uses the wavelet transform [179, 180] to extract features from the image and creates a hash based on these features.
These techniques generate an image's unique digital signature or hash, which can be used

to detect duplicate images. However, crucial differences between these algorithms can impact their performance. The efficacy of hash-based approaches for detecting duplicate images depends on exact or near-exact image features. These techniques are discussed in detail below:

aHash (Average hash): Images are two-dimensional signals [181] with distinct frequency components. High-frequency components describe edges and texture, and low-frequency components provide structural information. aHash is a simple and effective perceptual hashing technique that generates fingerprints by comparing each pixel value to the image's average pixel value. This approach uses low-frequency visual information through the following procedure:

Algorithm 2.1 aHash

Input: Image name and Image path

Output: 64 bit binary code

1. Resize the input image to 64×64
2. Grayscale the resized image and calculate its mean value M

$$h(i) = \begin{cases} 0 \\ 1 \end{cases} \text{ if } A(i) < M \text{ otherwise } (1 \leq i \leq 64) \tag{2.1}$$

3. Compare each pixel value $A(i)$ to M to convert it to binary
4. Convert the output into a 64-bit binary code

pHash(Perceptual Hash): pHash produces an image's hash from its visual content. Perceptive hash (pHash) algorithm [181] improves the average approach by extracting image features using a discrete cosine transform (DCT) [182] and providing a hash that is robust to image transformations, including cropping, scaling, and rotation [176]. DCT translates the image from the pixel value domain to the frequency domain, then calculates the average of the generator matrix's upper left corner's low-frequency information area. Then, create image fingerprints by comparing pixel values to the average. DCT is extensively used in signal and image processing, particularly for lossy data reduction to reduce duplication and correlation in general images, due to its considerable "energy compaction" feature [183, 184].

The implementation of pHash contains the stages listed below:

Algorithm 2.2 pHash

Input: Image name and Image path

Output: Output: pHash Code or the fingerprint of the image

1. Resize the input image to 64×64
2. Grayscale the resized image
3. Perform a 64×64 DCT to the grayscale image to generate a 64×64 DCT coefficient matrix
4. Hash extraction uses 64 low-frequency DCT coefficients
5. Calculate the mean value of the DCT coefficients:

$$m = \frac{1}{64} \sum_{r=1}^{64} \sum_{c=1}^{64} d_{r,c} \tag{2.2}$$

   where $d_{r,c}$ indicates the DCT coefficients and m is the mean of DCT coefficients

6. Compute the pHash code of an input image by comparing the threshold with respect to their mean value.

$$h_i = \begin{cases} 0 & C_i < m \\ 1 & C_i \geq m \end{cases} \quad , \quad i = 0, 1, 2, \ldots, 63 \tag{2.3}$$

   where h_i is the bit of the perceptual hash at position i

wHash(Wavelet Hash): wHash [185] is very similar to pHash, with the primary difference being the use of the discrete wavelet transform (DWT) instead of the discrete cosine transform (DCT) to calculate the coefficients mentioned in Algorithm 2.2. Moreover, the DWT coefficient is separated into two distinct coefficients: high pass coefficients and low pass coefficients. The high pass coefficients provide details about the intensity of each pixel, while the low pass coefficients offer an approximation of pixel intensities. Apart from this difference, the remaining steps involved in the computation of the hash are analogous to those used in pHash.

dHash (Difference Hash): dHash [186] creates image hashes by comparing the intensities of nearby pixels. It turns the image to grayscale and then subtracts each pixel's value from the value of the next pixel in the horizontal direction. This generates a hash that is sensitive to small image changes but may not be as robust to image modifications

as pHash, aHash, or wHash. It tracks relative gradient directions more accurately as aHash. The dHash algorithm works as follows:

Algorithm 2.3 dHash

Input: Image name and Image path
Output: dHash Code or the fingerprint of the image

1. Resize the input image to 64×64
2. Grayscale the resized image
3. Compute the difference between adjacent pixels

$$dpx_{r,c} = px_{r,c+1} - px_{r,c} \tag{2.4}$$

where $px_{r,c}$ is the gray value of the resized image at row r and column c, and difference value is represented by d$px_{r,c}$

4. Each row has 64 variations between adjacent pixels and 64 rows. All rows create a 64-bit 1D array. where dr,c are DCT coefficients and m is their mean.
5. Calculate the hash code and assign bits. Whether the right pixel is brighter than the left pixel sets each bit to 0 or 1.

$$m = \frac{1}{64} \sum_{r=1}^{64} \sum_{c=1}^{64} d_{r,c} \tag{2.5}$$

where $d_{r,c}$ indicates the DCT coefficients and m is the mean of DCT coefficients

6. Compute the pHash code of the input image by comparing the threshold with respect to their mean value.

$$h_i = \begin{cases} 0 & C_i < m \\ 1 & C_i \geq m \end{cases} \quad , \quad i = 0, 1, 2, \ldots, 63 \tag{2.6}$$

where h_i is the bit of the perceptual hash at position i

Following is the comparison of these image hashing techniques as discussed above based on certain parameters:

- Robustness: pHash is considered to be the most robust to image transformations, followed by wHash and aHash and dHash is the least robust
- Speed: dHash is faster than pHash, aHash, and wHash, as it uses simple subtraction operations to create a hash, while pHash, aHash, and wHash use more complex operations.
- Space: dHash requires less space than pHash, aHash and wHash, as it uses simple subtraction operations to create a hash.
- Similarity: pHash is more similar to the actual image, as it uses DCT to extract features from the image, aHash, and wHash are also similar to the actual image but

less than pHash, while dHash uses simple subtraction operations to create a hash, which is less similar to the actual image.

2.5.3 Matching Distance Measures

Features are extracted from the image and compressed to hash values to detect exact or near-exact images in a large-scale storage system. A 1-D feature vector is extracted from a 2-D image and this feature vector represents the image's perceptual properties. Thus, two images that appear similar to the human visual system must have feature vectors that are close to each other in terms of some distance metric [176]. Similarly, two visually distinct images must have significantly dissimilar feature vectors. To assess image similarity, distance-matching algorithms such as Hamming and Euclidian distance algorithms are employed. These algorithms use matching to extract strong pattern geometric and statistical information from grayscale images and have been previously used for iris detection [187, 188] are now used for surface characterization [189]. These are computed between the test image and the reference image database.

2.5.3.1 Euclidean Distance

Euclidean distance (ED) is the straight-line distance between two pixels in a Euclidean space. The Euclidean distance for n-dimensional space is given by

$$D_E(a,b) = \sqrt{\sum_{i=1}^{n} (a_i - b_i)^2} \tag{2.7}$$

where n is the feature vector dimension and a_i is the i^{th} component of the feature vector of reference image, and b_i is the i^{th} component of the feature vector test image.

2.5.3.2 Hamming Distance

The smallest number of bit flips required to transform one hash into another is known as the Hamming Distance (HD). Thus, the Hamming Distance between two hash strings $\mathrm{H}_x = \{h_x(0), h_x(1), \ldots, h_x(n-1)\}$ and $\mathrm{H}_y = \{h_y(0), h_y(1), \ldots, h_y(n-1)\}$ is computed as:

$$\mathrm{H\ D}(H_x, H_y) = \sum_{i=0}^{n-1} h_x(i) XOR h_y(i) \tag{2.8}$$

Hamming distance calculates the number of positions at which the symbols diverge, similar to Euclidean distance.

2.6 Multimedia Based Deduplication Techniques

Table 2.6 discussed some key findings of the existing image and video deduplication techniques, categorized into image or video deduplication, exact or near-exact image or video duplicate detection, and security-based image or video deduplication. Deduplication algorithms were applied on images to identify similar images and on videos to identify similar frames from videos using various parameters. Different feature extraction techniques, such as SIFT, SURF, PCA-SIFT, BRISK, etc., or CNN-based feature extraction techniques are employed to extract relevant features from the images or videos. Hashing algorithms are then applied to these extracted features to generate hashes and distance measures are used to assess the similarity between two images based on a predefined threshold.

Video data typically consumes a substantial amount of storage space, especially as video quality and resolution increase. High-definition and 4K videos, for example, demand significantly more storage capacity than lower-resolution videos. Videos are essentially a sequence of images or frames displayed in rapid succession. There is a strong correlation between consecutive frames, as they share many similarities. In video compression, this correlation is leveraged to reduce storage space. Instead of storing every frame independently, video compression techniques, like MPEG and H.264, store keyframes (intra-frames) that contain complete image information and interframes (predicted frames) that only store the differences from the previous frame.

While interframe video storage optimization is already taken care by the different video storage formats like MPEG etc, applying deduplication between two videos becomes a slightly more complicated process. The problem becomes simpler in the case of exact video deduplication, where the same algorithm developed for the images can be applied by comparing the frames of the two videos one by one. However, in the case of near-exact videos, where the decision is to have an optimized storage procedure instead of having both copies, the storage procedure needs to include information to reconstruct the video using the exact copy pointer as well as extra frame information. The process becomes more complicated when the differences lie in various interleaved positions rather than at any end of the video. Businesses and cloud service providers can exploit video deduplication to optimize storage, improve data management, and enhance video services, such as video

So, Multimedia-based deduplication techniques face challenges like scalability, high accuracy, and performance. The feature extraction techniques like SIFT, SURF, PCA-SIFT, and BRISK need further enhancements to achieve scalability, accuracy, and performance. So there is a need for an autonomic, scalable, and efficient multimedia deduplication technique.

Table 2.6: Multimedia Based Deduplication Techniques

Author(s)	**Technique**	**Description**	**Findings**
Gharib M et al., [190], 2023	N-anonymity algorithm, ID-based key management algorithm	Secure deduplication using the hashing technique to detect similar videos	Effective technique
Mageshkumar N et al., [191], 2023	Neighbourhood Correlation Sequence (NCS) and Seagull Optimization(SGO) algorithm	Deep transfer learning enabled the classification model to detect duplicate images	Better Performance
Kumar S et al., [192], 2023	Block-based DCT, Stationary Wavelet Transform	Copy-move forgery detection and localization in digital images using hybrid features to match identical regions	Exhibit the efficiency, robustness, and accuracy
Qin A et al., [193], 2022	CNN-based and ORB features	Approximate nearest neighboring search (ANNS) indexes and searches the high-dimensional feature vectors in video deduplication	High throughput and Compression ratio
Deng C et al., [194], 2022	Similarity-based deduplication and fast sampling method (Feature Map)	Lossless image deduplication reduces fine-grained image redundancy	Provides higher throughput and compression ratio
Sujatha G et al., [195], 2022	Hash-based algorithms	Examine cryptographic hashing algorithms to identify the image uniquely.	It helps improve image identification
Matatov H et al., [196], 2022	ORB Descriptors, VGG-16	Evaluate visual near-duplicates retrieval methods using a large-scale image dataset from social media	Obtain High recall values and successful in retreiving images
Thyagharajan KK et al., [197], 2021	Various Feature Extraction Techniques	Reviewed various feature extraction methods and how efficiently these detect near duplicate images	Discussed various feature extraction techniques and its challenges
Qin A et al., [198], 2021	Heuristic and generic algorithm, Hashing algorithm	Video retrieval framework to recognize similar videos and images using coarse-grained deduplication	Scalable and high performance

Continued on next page

Table 2.6 – *(Continued)*

Authors	Technique	Description	Findings
Xie H et al., [199], 2021	Quality assessment and selection algorithm	Eliminate the visually identical image chunks in data backups	Achieves a good reduction ratio
Jia W et al., [200], 2020	Binary tree embedded algorithm, Scalable Hash Codes	Novel deep learning-based scheme for video deduplication framework	Strong and stable network system
Zhou Z et al., [201], 2020	SURF region detector, AlexNet	Local CNN feature-based image copy detection	Achieves a superior performance
Yan S et al., [202], 2019	DoG Detector, SIFT features, Approximate k-means (AKM)	Lightweight network used to extract local compact binary features for fast reliable near-duplicate patches matching	Better performance with higher query speed
Li Y et al., [203], 2019	MapReduce strategy, LSH)	Hash code-based distributed video deduplication	Efficient large-scale video deduplication
Alam F et al., [204], 2019	Perceptual hashing techniques, VGG-16	Presented a social media image processing pipeline that includes noise-filtering to extract useful information	Efficient and timely extraction of useful information
Yan H, et al., [205], 2018	File and block-level video deduplication	Secure video deduplication with Centralized privacy-preserving To remove the encrypted duplicated data	Increase security and deduplication efficiency
Hu W et al., [206], 2018	LSH	Deep constrained siamese hash coding neural network with deep feature learning	Reduce query time, a practical algorithm
Bini SP et al., [207], 2017	Set Partitioning in Hierarchical Trees (SPIHT) algorithm	Compress the image using SPIHT algorithm, perform partial encryption on the compressed image	Achieve a secure deduplication of images in data storage
Zhou Z et al., [208], 2017	PageRank algorithm, Coarse-to-Fine Clustering method	Fast and accurate near-duplicate elimination approach for visual sensor networks	Better performances in terms of both efficiency and accuracy
Li X et al., [209], 2016	Perceptual hash algorithm	Secure perceptual similarity deduplication generates signatures to detect duplicate images	Achieves high deduplication, storage, and bandwidth savings
Deshmukh AS et al., [210], 2016	MapReduce and Pearson correlation technique	Fast duplicate image identification systems	Improves efficiency and reliability of the system

Continued on next page

Table 2.6 – *(Continued)*

Authors	Technique	Description	Findings
Rashid F et al., [211],2016	Set Partitioning In Hierarchical Trees (SPIHT)	SPIHT compresses and partially encrypts images to prevent CSP access	No extra computational overhead for image encryption
Nian F et al., [125], 2016	Local-based Binary Representation (LBR), Binary pattern, Histogram	Online near-duplicate image detection using LBR	Robust to image variations, Low computational speed, and better performance
Huang F et al., [212], 2016	Image relational graph (IRG), PageRank graph model	Uses local and global features to detect near-duplicate images	Effective method to detect near-duplicate images
Chen CC et al., [154], 2015	SIFT, Hamming distance	Binarized SIFT features and hashes for fast image retrieval	Low complexity and fast retrieval time
Zheng Y et al., [213], 2015	Scalable Video Coding	Encrypted cloud media center for safe video deduplication	Scalability issue
Zargar AJ et al., [214], 2015	Content-based Image Retrieval, Block Truncation Coding	Photo-based deduplication to discover duplicate electricity bills in large databases	Better space utilization
Li X et al., [215], 2015	Fixed-size blocks	Video deduplication for image privacy	Saves storage
Hua Y et al., [152], 2015	PCA, SIFT, DoG	SmartEye and in-network coarse-grained deduplication to identify similar images	Energy Savings and bandwidth
Yao J et al., [155], 2014	Contextual Descriptors	Contextual descriptor to measure image contextual similarity	Better performance and efficient method
Hua Y et. al, [216],2014	FAST DoG, PCA-SIFT, LSH, Bloom filter	A novel near-real-time method based on DoG and PCA-SIFT to detect image features	Reduce the processing latency of parallel queries.
Li L et al., [217], 2014	SURF and DAISY	Image matching uses SURF and Daisy descriptor	Improves matching accuracy
Lei Y et al., [218], 2014	Kd-trees	Cluster of Uniform Randomized tree to detect near-duplicate images	Improves the detection efficiency and search space.
Thomee B et al., [219], 2013	Content-based image detection	A comparative study using content-based duplicate image detection	Analysed descriptor size, description time, and matching time

Continued on next page

Table 2.6 – *(Continued)*

Authors	Technique	Description	Findings
Li Z et al., [220], 2013	SIFT, k-means clustering and LSH	SIFT features are extracted and clustered using the k-means algorithm	Detect near-duplicates effectively
Wang XJ et al., [221], 2013	MapReduce	Large-scale duplicate detection using local and global image descriptors	Effective method in terms of high accuracy and recall
Chen M et. al, [50], 2013	Haar wavelet, B+ Tree	Gray block features create a B+ tree index, and Haar Wavelet extracts image edges	Higher deduplication rate, but scalability is a challenge
Leutenegger S et al., [222], 2012	BRISK, Hamming Distance	Novel key-point detection in continuous scale space to match similar images	Quality matches at less time
Dong W et al., [223], 2012	Scale-invariant feature transform(SIFT)	Entropy-based filtering for near duplicates	Scalable using Hadoop commodity server
Velmurugan K et al., [224], 2011	Speeded Up Robust Features(SURF) and K-dimensional(Kd)-tree	Kd-tree with Best Bin First algorithm indexes and compares image features extracted by the SURF algorithm	Improves average image-matching precision
Ramaiah NP et al., [110], 2011	Content-based Image Retreival(CBIR), Histogram Refinement, K-means-based clustering	Histogram Refinement removes family photo-based duplicate ration cards	Deduplication requires human intervention
Katiyar A et al., [225], 2011	ViDedup (2011)	Application-aware video compression and visual redundancy detection	Scalability issue
Zhao J et al., [226], 2010	Harris and SIFT	Improved Harris corner detector and SIFT using the nearest neighbor algorithm for feature matching	Accurate and fast matching
Nikolaidis N et al., [227], 2009	R-trees, Linear Discriminant Analysis(LDA)	Color-based R-tree and LDA descriptors for image and video fingerprinting	Efficient for digital rights management of images and video
Yang X et al., [228], 2009	Local-Difference-Pattern(LDP), LSH	Local-feature-based framework indexes near-exact images and videos	Less processing and storage
Yu X et al., [229], 2008	SIFT technique	SIFT-based technique and geometry isomorphic relationship to identify image homology	Robust to image embedding

Continued on next page

Table 2.6 – *(Continued)*

Authors	Technique	Description	Findings
Chum O et al., [230],2008	SIFT and k-means clustering	Similarity-based min hash technique to find near-duplicate images	Improves search efficiency without computational cost
Srinivasan SH et al., [231], 2008	Fourier-Mellin Transform(FMT)	Fourier-Mellin transform image fingerprinting to detect near-duplicate images	Fast, accurate, scalable method
Shen HT et al., [232], 2007	Near Duplicate Video Clip (NDVC), K-nearest neighbor algorithm	UQLIPS, a visual content-based NDVC detection system, detects similar images efficiently	Fast and accurate for real-time search
Foo JJ et al., [233], 2007	SICO (Similar Image Collator), PCA-SIFT	SICO, uses PCA-SIFT to extract features and Hash-based probabilistic counting, detect near-duplicate images	Effective and efficient method
Gavrielides MA et al., [234], 2006	Histogram technique, different quantization methods	image fingerprinting generates robust, unique image descriptors	Better performance of the system
Naturel X et al., [235], 2005	Fast shot-based method, Discrete Cosine Transform (DCT)	Detect duplicate TV video shots	Detect duplicate shots quickly
Lu CS et al., [236], 2005	Discrete Cosine Transform(DCT), Harris Corner Detector	DCT-based mesh-based robust image hashing	Improves image hashing resistance to geometric distortions
Ke Y et al., [128], 2004	DoG, PCA-SIFT, and LSH	DoG-PCA-SIFT system to detect near-duplicates.	High computational overhead as the database grows
Seo JS et al., [237], 2004	Radon Transform, Hamming distance	Radon transform and perceptual hashing for multimedia image fingerprinting	Highly robust against affine transformation

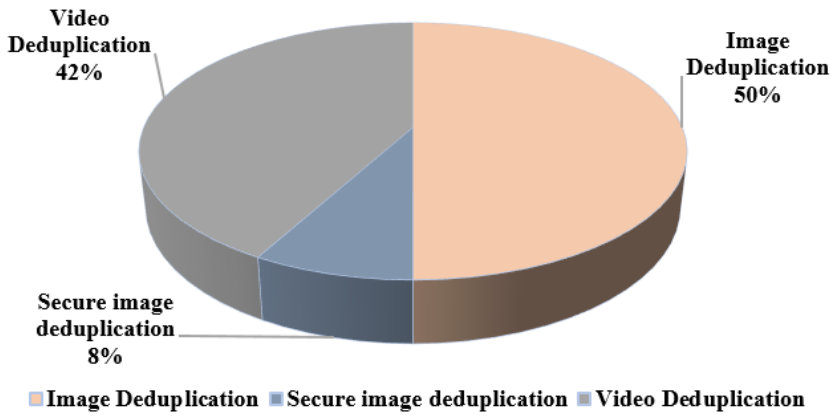


Figure 2.7: Research Contribution of Image, Video and Secure Image Deduplication Techniques

Deduplication techniques in the multimedia domain encompass image, video, and secure image deduplication. Figure 2.7 exhibits the percentage contribution of research papers presented in these three categories. Among multimedia-based deduplication techniques, exact image deduplication has been the primary focus, followed by video duplicate detection and secure image deduplication.

Additionally, this chapter not only classifies multimedia-based deduplication techniques and discusses their specific aspects but also presents their evolution, aiding researchers in identifying relevant methods within their sub-area. Figure 2.8 represents the evolution of primary multimedia-based deduplication techniques classified into image deduplication and video deduplication techniques. The image deduplication techniques focus on exact image deduplication and near-exact image duplicate detection. Notably, video deduplication techniques are applied to video frames, which are then treated as images for further application of image deduplication techniques on the video data. Figure 2.8 shows significant contributions made in different years by various researchers and the methods they employed.

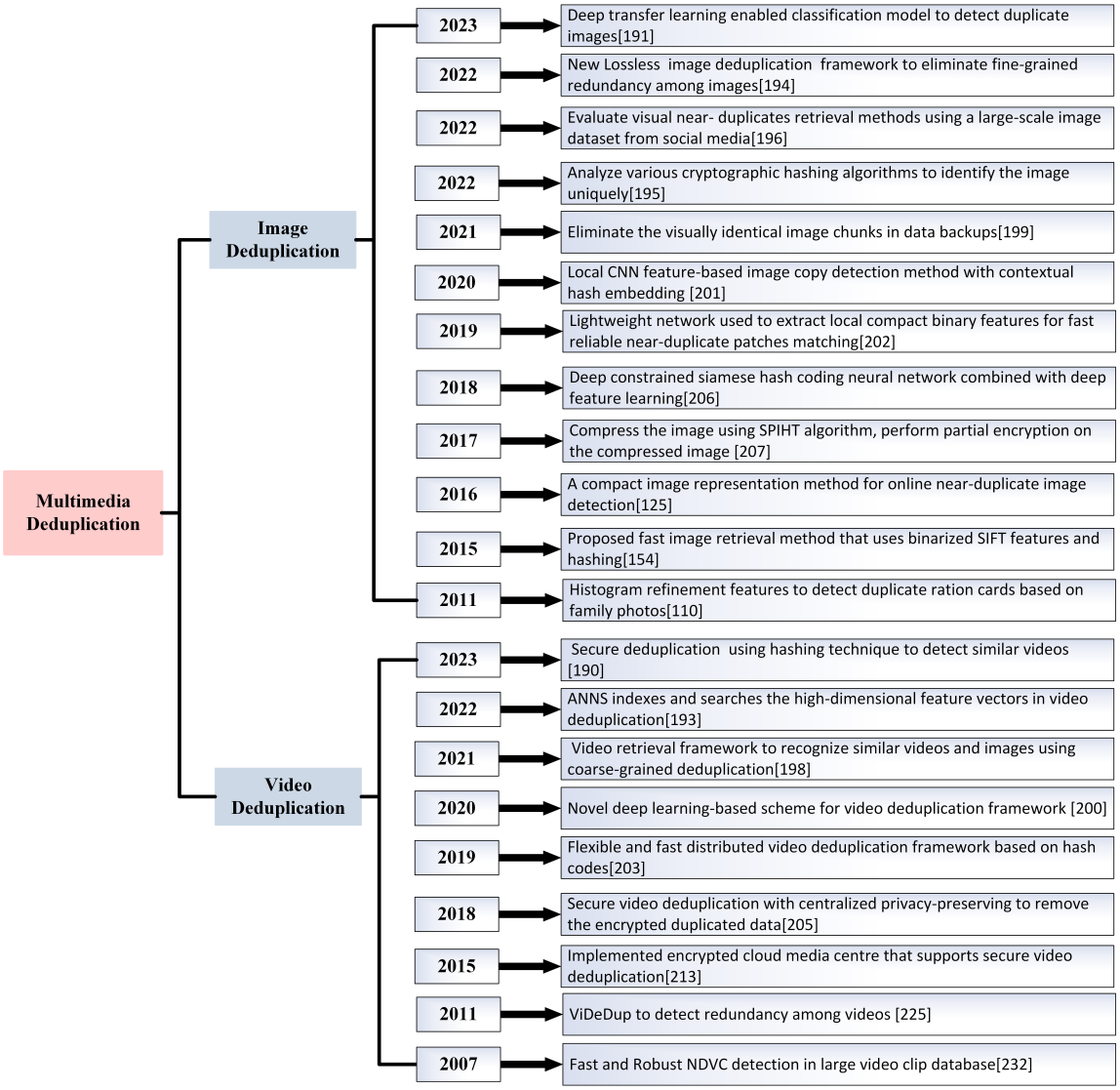


Figure 2.8: Evolution of Image Deduplication Techniques

2.7 Energy Saving Techniques in Cloud Storage System

Energy-efficient and cloud storage computing has emerged as the latest buzz in the ICT industry today [238, 239]. The increasing volume of data has brought about challenges in storage and retrieval processes. As a result of the proliferation of complex data-intensive applications in cloud storage systems, large data centers have emerged, leading to higher energy consumption. These data centers not only incur high operational costs but also contribute significantly to carbon footprints due to their substantial energy usage. As the cloud computing industry continues to grow, energy consumption and carbon emissions also rise, prompting extensive research in energy-efficient cloud computing and its impact

on performance in large-scale data centers.

To ensure high reliability and availability, cloud data centers utilize overprovisioning and redundant power distribution, processing, cooling resources, and storage, including discs and controllers, consume significant power in data centers [240]. Data centers consume 1.5% of global energy consumption[241], with storage systems accounting for 40% of total energy usage[242]. According to [242], cooling and power distribution systems consume 45% and 15% of total energy, respectively, while IT equipment (computer servers and networking equipment) accounts for approximately 40%. The growing volume of digital data necessitates more storage space, leading to increased costs and performance requirements for backups. As a result, there is a substantial demand for data centers to accommodate storage needs [243].

Cloud computing technology offers storage as an essential service, with cloud service providers maintaining at least one redundant copy of data in the cloud storage server to ensure high availability at all times and locations. Data deduplication technology has attracted many cloud vendors as it reduces storage costs for users by eliminating duplicate copies of data from the cloud storage server and optimizing storage space [244].

Energy consumption remains a significant issue in data centers, necessitating sophisticated and costly cooling arrangements to dissipate heat from server components and ensure reliability. The estimated cost of electricity and cooling equipment for a single rack in a standard data center is $52,800.1 [245]. The power costs vary based on the energy consumed by each data center. Before developing new techniques for energy efficiency, it is crucial to assess existing energy-saving approaches and understand their impact on energy management. Data deduplication techniques are widely used in backup and storage systems [246], with previous studies aiming to enhance deduplication efficacy, reduce time and cost, and improve system throughput. The common optimization goal of these approaches is energy efficiency. Table 2.7 discussed some key findings of the existing energy-saving techniques in cloud storage systems.

Table 2.7: Energy-Saving Techniques in Cloud Storage System

Author(s)	**Technique**	**Description**	**Findings**
Talwani S, et al., [247], 2022	Machine Learning (ML) based Artificial Bee Colony (ABC)	Energy-aware VM allocation and migration for cloud data centers' rising needs	Energy Savings
Li B, et al., [248], 2021	CNN and Variable-length Integer used	CBIR retrieval for encrypted JPEG images between the image owner and the cloud server	Good performance and retrieval accuracy

Continued on next page

Table 2.7 – *(Continued)*

Authors	Technique	Description	Findings
Xu M et al., [249], 2020	Workload-Shifting Algorithms	Workload-shifting algorithm reduces cloud data center carbon emissions	Reduce emissions and response time.
Yan F et al., [250], 2019	Disc scheduling algorithm (LDP) and variable-length chunking	CGDL-based disc scheduling and selective deduplication algorithm for energy-saving	Energy-saving disc read/write operations
Ali SA et al., [251], 2019	Categorise energy-efficient methods	Discusses cloud data center energy use and energy-saving strategies	Increase resource utilization and lower power usage
Mishra SK, et al., [252], 2018	CloudSim simulator	Task-based VM-placement approach (ETVMC) for heterogeneous tasks, VMs, and hosts in the cloud	Energy-efficient approach
Mishra SK et al., [253], 2018	Adaptive Task Allocation Algorithm (ATAA) changes task execution times.	AATA, cloud-based adaptive task allocation using a host, virtual machine, and task model	Energy-efficient method
Zahedi Fard SYet al., [254], 2017	Dynamic threshold method, CloudSim simulator.	DthMf, a virtual machine consolidation approach that saves energy.	Energy-efficient technique
Hieu NT et al., [255], 2017	VM consolidation algorithm	VMCUP-M, a VM consolidation algorithm to boost cloud data center energy efficiency	Energy reduction and SLA compliance,
Zhou Z et al., [256], 2016	Adaptive three-threshold energy-aware algorithm	ATEA and a novel virtual machine (VM) placement technique	Reduce energy consumption
Uchechukwu A et al., [257], 2014	Energy consumption optimization policies	New energy consumption models with static and dynamic cloud components	Findings show that 20% of the energy can be saved
Lee YC et al., [258], 2012	ECTC and MaxUtil Task Consolidation Algorithms	Two energy-conscious task consolidation approaches to maximize resource usage	Experimental results show an energy-saving capability

This chapter provides an in-depth discussion of several taxonomies of deduplication techniques. Additionally, it identifies the deduplication tools offered by various storage firms. The following Table 2.8 presents a list of these deduplication tools along with a brief description of each.

Table 2.8: Deduplication Tools and Technologies

Tool	Description
StorReduce of Amazon	Deduplication tool for big data from Amazon
StorSimple of Microsoft Azure	Block-level deduplication for cloud storage system from Microsoft
Avamar of EMC	Variable length Deduplication from EMC
Quantum	Inline dedupe for backup
SSET of NetApp	Space Savings Estimation Tool (SSET) from NetApp
Data Domain of EMC	Inline deduplication tool from EMC
WinPure	Standalone software to clean up databases
Sepaton	Sepaton DeltaStor deduplication software
Netrics	Netrics helps to clean up projects and duplicate records
Revinetix	Revinetix works at the file level on backup
Exagrid	Deduplication tool for backup
CommVault	Deduplication tool for backup for heterogeneous storage infrastructure

2.8 Conclusion

This chapter provides a comprehensive and systematic review of data deduplication techniques, thoroughly analyzing and evaluating various methodologies employed in this field. It specifically investigates text and multimedia-based deduplication techniques, shedding light on the challenges encountered in these domains, with a particular focus on image-based deduplication methods. The need for effective solutions to tackle these challenges is emphasized.

The next chapter presents a Deep CNN-based online data deduplication technique for cloud storage systems. The technique finds duplicate and near-duplicate data in the cloud storage system and improves the overall performance in terms of system storage efficiency.

Chapter 3

Proposed Online Image Deduplication Technique

The previous chapter presented a comprehensive review of the subject under study, i.e., Data Deduplication. The literature review emphasized the importance of data deduplication and its various techniques in effectively identifying significant duplicates, with a specific focus on text and multimedia-based deduplication methods. Moreover, the review provided insights into the challenges encountered in these domains, particularly in image-based deduplication methods.

Online data deduplication in cloud storage systems has not been addressed to a large extent, although extensive research has been undertaken in this field. The real-time duplicate image detection and subsequent removal of the exact or near-exact duplicate images is a major challenge in cloud-based storage systems as it requires excessive computation and memory. To provide a solution to the cloud-based storage system, a Deep CNN-based online image deduplication technique has been proposed and designed in this chapter to detect exact or near-exact images. This chapter highlights the implementation of the proposed work in two phases. The first phase determines handcrafted, and CNN features to detect exact or near-exact images. The second phase is a classification based on image clusters. The performance of the proposed method has been evaluated based on image-matching accuracy and the time required to match the images.

The chapter is organized as follows: Section 3.1 presents the brief introduction of online image deduplication followed by the architecture of the proposed technique in Section 3.2. The CNN-based image deduplication technique is presented in Section 3.3, while Section 3.4 presents patch generation using a Hot decomposition vector. Section 3.5 presents the experimental setup, i.e., the dataset taken and evaluation metrics, followed by experimental results in Section 3.6. The Performance Comparison of Image Classifiers based on different image variations is evaluated in Section 3.7, followed by the performance evaluation of cross-domain net in Section 3.8. Section 3.9 concludes the chapter.

3.1 Online Image Deduplication: An Overview

Removing duplicate images from a cloud storage system is known as image deduplication. It is a method that eliminates duplicate images, boosts storage efficiency, and lowers storage costs. The process of online image deduplication looks for images on the cloud that are identical or nearly identical. The effectiveness of recognizing similar or nearly identical images directly influences how well online images may be deduplicated. Image deduplication approaches are subdivided into exact and near-exact image detection based on image detection techniques. Exact image deduplication approaches detect duplicate similar images without considering image modification or transformation. Near-exact image detection contemplates image transformation and its alteration by cropping [259, 260], scaling, rotation, etc. For exact or near-exact image deduplication, the method remains the same. The only difference lies in the storage of image transformation for near-exact images.

The image deduplication technique identifies duplicate images, eliminates all but one copy, and creates local references to the data that users can easily access. The central idea behind image deduplication is to delete duplicate images through five stages [50] that include feature extraction [261], high-dimensional indexing, accuracy optimization by extracting image edge information, centroid selection using cluster formation, centroid image fixation, and deduplication evaluation and comparison as discussed in Section 2.5 in the above chapter.

Figure 3.1: Exact or Near-Exact Images

When a new image is received, the query image features are extracted using various

image feature extraction techniques, as discussed in Section 2.5.1 in Chapter 2 in the image deduplication approach. Their fingerprints or signatures are formed. These image fingerprints are the main focus of deduplication algorithms for further processing. The signature database matches newly generated fingerprints with previously stored fingerprints. If it already exists, a pointer is created to match the image with a previously stored image in the storage database. Otherwise, the fingerprint is considered new, and both the fingerprint and the image are stored in the storage database.

The chapter 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 proposed technique is a novel online technique to detect near-exact images for a large distributed cloud storage system. Figure 3.1 demonstrates the existence of near-exact images in the image database.

3.2 Architecture of Proposed Online Image Deduplication Technique

Online data deduplication in cloud storage systems has not been addressed to a large extent, although extensive research has been undertaken in this field. The real-time duplicate image detection and subsequent removal of the exact or near-exact duplicate images is a major challenge in cloud-based storage systems as it requires excessive computation and memory. To provide a solution to the cloud-based storage system, a Deep CNN-based online image deduplication technique is required to detect exact or near-exact images.

The proposed technique consists of three components: client-side preprocessing of the query image; an intermediary computation node for signature matching of images with image signature databases of existing hash tables; and distributed image storage. The proposed technique extracts query image features on the client side and compares their signatures to an image database maintained on an intermediate node. The original image database is stored on the server. The following are the architectural components of the proposed technique:

- Client-Side Pre-Processing: The client is the source of an image or new query image that must be analyzed to determine an exact or near-exact image match. This image is queried against the intermediate node's signature database for a match. Image feature extraction techniques are utilized to extract features from a query image. The proposed approach uses CNN-based feature extraction to detect exact

or near-exact image detection for online deduplication. The query image's extracted features are sent to intermediate computation nodes for further processing to check whether these are unique features of an image or have previously been stored in the database. The client side is also responsible for deciding whether or not to accept the nearly identical version of a query image. The exact match between a query image and the image database results in the elimination of duplicate images. For a near-exact image match, the user is prompted to decide whether the near-exact image should be stored or discarded. The image patches are generated to store a nearly identical image version in the database. The image patches contain color synchronizations, affine transformations, and crop data.

- Intermediate Computation: The intermediate node is responsible for maintaining the image signatures or image hash values to process the image matching. An image signature database looks up features extracted from query images on intermediate computation nodes. If there is no match between the query image features and the signature database, it indicates that the query image is a new image. The signature of this new image is added to the signature database, and the image itself is stored directly on the distributed storage on the server side. The exact match of the query image leads to duplicate image detection, and a pointer to an existing duplicate image is provided. The signature database is initially empty, and every time a new image enters the system, its features are extracted using our trained CNN. These extracted features are then provided to the LSH (Locality-Sensitive Hashing) hash table for efficient indexing, allowing for quick retrieval and matching of similar images in the cloud storage system.

- Distributed Image Storage: Distributed Image Storage utilizes distributed nodes on the storage system to store entire images. The original image is distributed among multiple storage nodes. Any match of the near-exact image is presented to the client-side end-user for approval to maintain the original image or its near-exact match.

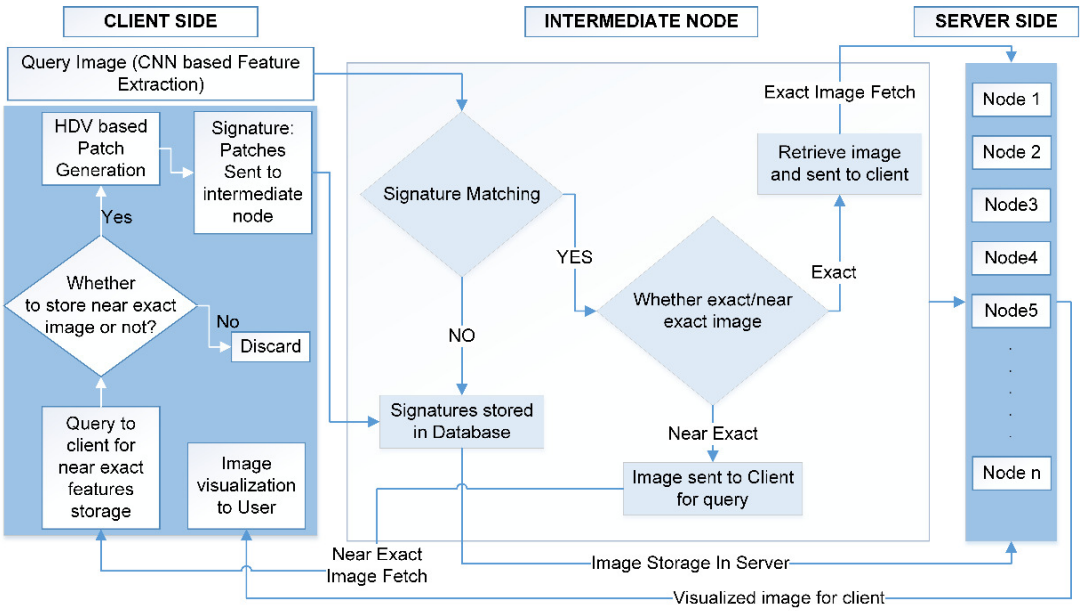


Figure 3.2: Proposed Online Image Deduplication Technique

Figure 3.2 illustrates the architecture of the proposed Deep CNN-based exact or near-exact image detection for online deduplication. Deep Convolutional Neural Network (CNN)-based image feature extraction is employed in proposed online deduplication because of its capacity to handle complex patterns, and variability, adapt to various applications, maintain accuracy, and enable efficient processing of large image datasets. The image patches are generated using the proposed HDV technique for a near-exact version of an image and kept in the database. It includes details on affine transformations, cropping, and color synchronization.

The proposed CNN-based techniques incorporate distributed processing components. Image pre-processing occurs on the client side, while the intermediate computation nodes handle image signature or hash matching through efficient indexing. This architectural design harnesses distributed computation and image storage techniques, ensuring high scalability. The proposed HDV image patch generation method takes advantage of distributed image storage and is readily scalable for handling extensive datasets.

3.3 Convolutional Neural Network

Convolution Neural Network [262, 263, 264] has attained high success in large-scale image and video recognition. The CNN architecture comprises feature extraction and classification layers. Feature extraction layers of a CNN consist of convolution, activation, and pooling layers, whereas the classification layers use Fully Connected or Activation layers. Deep learning neural networks use feature extraction layers to learn a deep hierarchy

of features, making them highly effective for image recognition. Deep CNN (DCNN) features are invariant to translations, rotations, and illuminations and have succeeded greatly in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). There are several models for deep network techniques, such as the AlexNet [263], VGGNet [264], and GoogleNet [265]. The details of the most significant layers used in the proposed technique are discussed below:

- Convolutional layers: Convolutional Layers consist of weighted matrices called filters or kernels. The following convolution formula can be used to calculate the output feature map values:

$$G[r, c] = (F * f)[r, c] = \sum_i \sum_k h[j, k] f[r - j, c - k] \tag{3.1}$$

  where F stands for the input image, f stands for the filter, and the output matrix represents the row and column indexes r and c. The *operator is matrix multiplication.

- Activation layers: Activation layers are an element-wise function used on the input feature maps to strengthen the network. The most popular activation function is the Rectified Linear Unit (ReLU). The ReLU [266] function is applied in the neural network's hidden layers to improve the nonlinearity of the images. It is a simple maximum-value threshold function. The mathematical equation for ReLU is:
  ReLU(i) = max(0,i) or f(i) = max(0,i)
  or alternatively

$$\mathrm{Re}\, LU'(i) = \{0, i < 0 \mid 1, i >= 0\} \tag{3.2}$$

  which specifies that negative nodes will output 0, making them irrelevant for prediction. These nodes are inactive. Nodes with values larger than 0 will remain intact and activated when deemed important for prediction. ReLU activation layers do not alter the dimensionality of the feature stack. ReLU outputs 0 for negative inputs and i for positive values.

- Pooling layers: Pooling layers use the average or maximum operation to minimize input data spatial size, further reducing network parameters and computations. MaxPooling, the most common pooling technique, takes the maximum value from each window or cluster of neurons in the preceding layer. It extracts an image's essential features and minimizes the feature map.

- Fully-connected layers: The last layer of a Convolutional Neural Network (CNN), referred to as the Fully Connected (FC) layer, is commonly recognized as the classifier layer.The Fully Connected (FC) which is the last layer of a CNN is also known as the classifier layer. The primary function of this layer is to connect neurons from the preceding layer so that every neuron in one layer is connected to every neuron in the next layer. The FC/ACTV [267] activation function transformed the summed weighted input from the node into the output of that input.

3.3.1 Deep CNN for Exact or Near-Exact Image Detection

Figure 3.3 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 256X256 image size convolved by a series of convolutional layers. The experimental work in this chapter evaluated AlexNet and VGGNet architecture trained on the ILSVRC-2012 dataset for feature extraction. These features are extracted from the second-last layer of both the nets giving a vector of 4096 for AlexNet and VGGNet, respectively. It further used pre-trained AlexNet and fine-tuned it using Pascal VOC 2007. Here, the fc7 layer is modified, and two more fc layers of sizes 1024 and 128 are added, as shown in Figure 3.3. The 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 and is denoted by Net-0 henceforth. This facilitates a low memory and time requirement for feature matching.

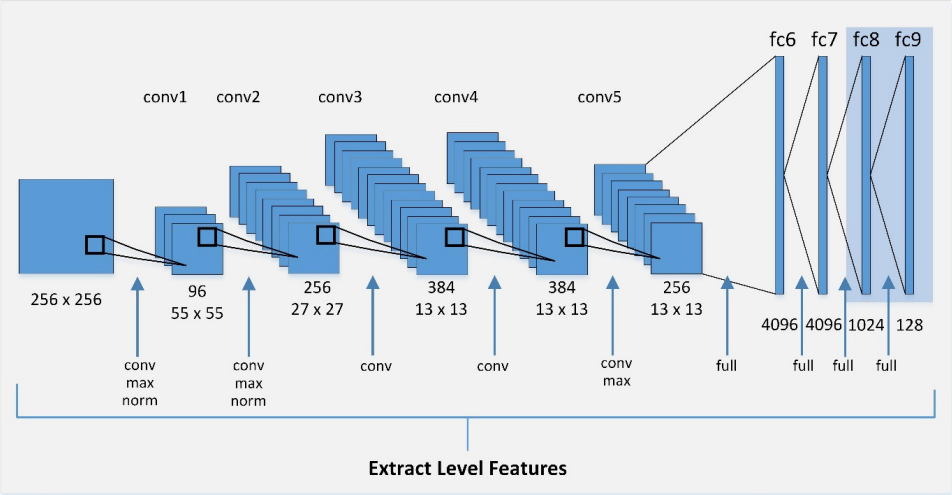


Figure 3.3: Architecture of Fine-Tuned AlexNet based on Exact or Near-Exact Image Detection for Online Deduplication

This research work later demonstrates the time variations in indexing depending on the

network chosen and the length of the feature vector extracted. These networks are referred to as Net-1, Net-2, etc., and are further described in the experimental results section of this chapter.

In this work, pre-trained features are obtained from AlexNet, VGGNet, and ResNet which were originally trained using the ImageNet dataset. The next step involves transfer learning or fine-tuning which involves two important considerations: (a) the choice of network and (b) the layer from which to extract pre-trained features. All experiments are further performed with AlexNet keeping in mind the time constraint for an online search. As the system moves to deeper networks, such as VGG or Resnet, its memory and computational requirements will increase. Hence, 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 well-known that deeper layers generate high-level features, whereas lower layers provide low-level features. The task at hand involves data deduplication without the data domain knowledge. For example, the data can belong to satellite images (PatternNet), handwritten characters (OMNI), or 3D model image data (ModelNet40), all substantially different from natural scenes in ImageNet or CIFAR100. This work hence aims to propose a network that deals not only with cross-domain images but also with perturbations in each of them.

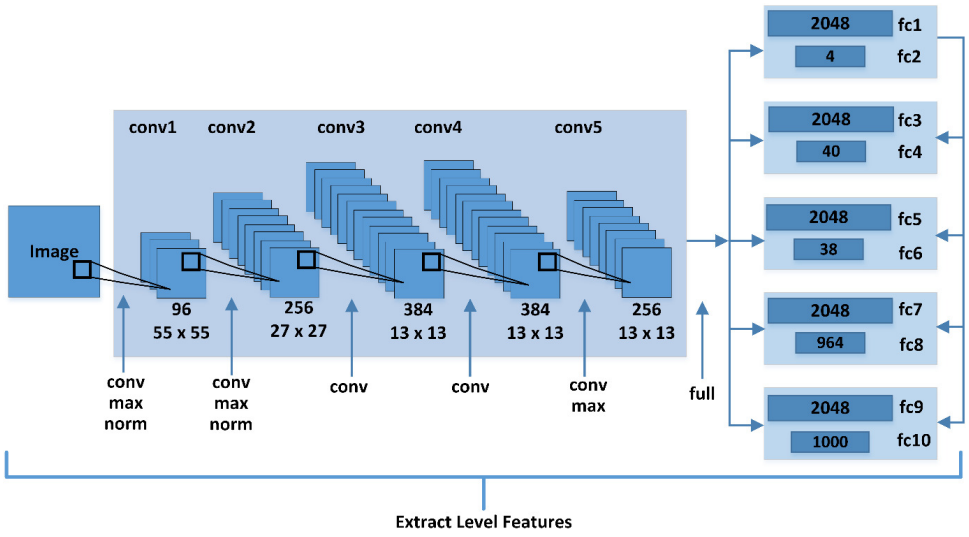


Figure 3.4: Architecture of Cross-Domain Images with Perturbations

As shown in Figure 3.4, a network architecture of cross-domain images with perturbations has been proposed. Output from the fifth convolution layer after max pooling is obtained using pre-trained AlexNet. This is then shared with five different modules, which learn weights individually to generate a 2048-sized feature vector. The resulting feature vector is utilized to generate hash tables using LSH (Locality-Sensitive Hashing). 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, four types of images are considered. As mentioned before, satellite images (PatternNet), handwritten characters (OMNI), 3D model image data (ModelNet40), and natural images with indoor and outdoor scenes are captured. Extensive experiments using these image types are later provided in the results section of this chapter.

3.4 Patch Generation for Near-Exact Images using Hot Decomposition Vector

An important consideration in obtaining a near-exact match is the decision to either discard or store the image. This chapter proposes a mechanism that allows the storage of near-exact images not in their entirety but as patches that differ from the original. Image patch generation, which keeps dissimilar parts of near-exact images, can further reduce storage requirements. Instead of keeping two images, where one is 45 degrees rotated with respect to the first, it stores only a single image and the necessary information required to reconstruct it. The proposed HDV technique compares these images and generates 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 equation 1.

$$D= (bw_i{}^1..., bw_i{}^2..... bw_i{}^{128}) (i= 1,..,n) \tag{3.3}$$

The above equation is referred to as the 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 3^{rd} order tensor, and a hot decomposition is carried out using a 2nd order mode matrix W_r and a third-order mode matrix W_c. W_r is the DCT of the mode 2 flattened matrix of A defined as

$$A = D \times_1 W_1^T \times_2 W_2^T \times_3 W_3^T \tag{3.4}$$

W_r and W_c are optimized using

$$\| W_r^{k+1\top} W_r^k \| > (1 - \sigma)\, d_r \tag{3.5}$$

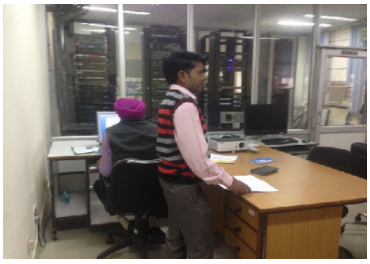
$$\| W_c^{k+1\top} W_c^k \| > (1 - \sigma)\, d_c \tag{3.6}$$

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 $[A_{\times 2} \ W_r^{\top} \times_3 \ W_c^{\top}, \ D]$.

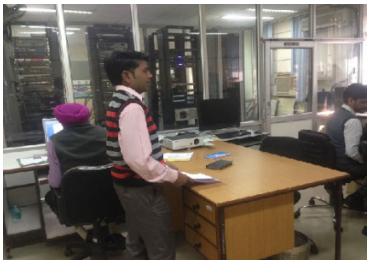
3.4.1 Near-Exact Image Patch Generation

For a near-exact image match, the user is queried for making 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 between the 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 the reconstruction of the image. HDV features of base and query images are used to obtain image alignment (affine) information. HDV features are used to generate image correlation and patches with poor correlation factors 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 transformations is considered currently. The hash table stores the key of each image which refers to the B+ tree. The leaf stores data ${}_{B \rightarrow Q}$, $\mathrm{dp}^c(\mathrm{x}_i, \mathrm{y}_i)$ (for i = 1 to 4) required for near-exact reconstruction. ${}_{B \rightarrow Q}$ gives the transformation between the base and the query image. $\mathrm{dp}^c(\mathrm{x}_i, \mathrm{y}_i)$ denotes the corner points of the base image matched with the query image. $\mathrm{dp}^c(\mathrm{x}_i, \mathrm{y}_i)$ is computed from $\mathrm{Q}_{B \rightarrow Q}$. Image patch p^{dp} obtained from $\mathrm{dp}^c(\mathrm{x}_i, \mathrm{y}_i)$ is transformed as and mapped with Q to obtain p^Q. p^Q is represented as Q having the general form 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 ${}^{pdp}{}_{B \rightarrow Q}$.

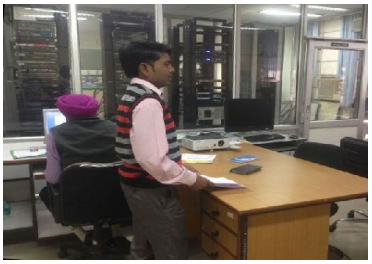
Figure 3.5 shows the generation of image patches. Figure 3.5(a) is taken as the original image, and Figure 3.5(b) is a near-exact image of Figure 3.5(a). Figure 3.5(b) is the query image and has been registered as shown in Figure 3.5(c). Figure 3.5(d) demonstrates the correlation between Figure 3.5(a) original image and Figure 3.5(c) registered image. Figure 3.5(e) represents the correlation of the image in Figure 3.5(b) with Figure 3.5(c). Finally, Figure 3.5(f) shows the patches generated from Figure 3.5(d) and Figure 3.5(e) that reflect the dissimilar part of the image.



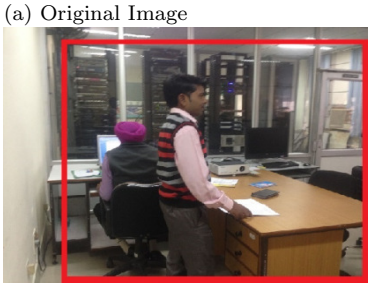
(a) Original Image



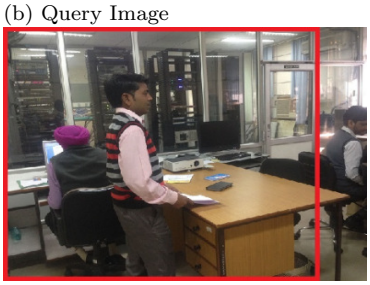
(b) Query Image



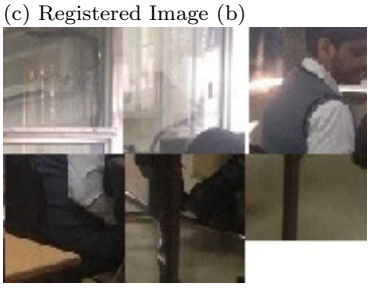
(c) Registered Image (b)



(d) Correlation of (a) with (c)



(e) Correlation of (b) with (d)



(f) Image Patches Generated

Figure 3.5: Near-Exact Image Detection for Image Deduplication and Generation of Patches for Dissimilar Parts

(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)

Figure 3.6: Near-Exact Image Detection for Image Deduplication and Generation of Patches for Dissimilar Part

Figure 3.6 shows another example of an original image, its near-exact images, image patches are generated for dissimilar parts, and the transformation matrix used for recon-

struction is stored as follows: $\tau_a{-}b$, (0.97870.02470, -0.02470.97870, -39.379313.57521.0000) $\tau_a{-}c$, (1.0032 -0.00830, 0.00831.00320, -83.66525.61711.0000) $2dp^c\,(53_1, 1_1)\,(359_2, 1_2)\,(53_3, 478_3)\,(359_4, 478_4)\,, 3dp^c\,(88_1, 1_1)\,(359_2, 1_2)\,(88_3, 478_3)\,(359_4, 478_4)$

3.5 Experimental Setup

This section provides comprehensive details regarding the datasets utilized, evaluation metrics, and other relevant information related to the proposed technique.

3.5.1 Datasets Used

The experimental work in this chapter is implemented in Matlab, Torch, and PyTorch. The different datasets used are PASCAL VOC, VGG-Flower, Aircraft, Omni, and CIFAR100, along with the PatternNet, ModelNet40, and self-collected datasets of 70,000 images (named DS-70K) to evaluate the behavior of different image detection algorithms for deduplication. Pascal VOC 2007 [268] has 20 classes. This dataset has 9963 images with 24640 annotated objects from 20 classes. VGG-Flower has 1362 images [269], Aircraft has 10200 images [270], Cifar100 has 60000 images [271], Omniglot has 17853 images [272], PatternNet has 38,400 images [273], and ModelNet40 has 48,000 images [274]. Additionally, the DS-70K dataset consists of images captured indoors and outdoors on the university campus. These images include various scenes like classrooms, laboratories, car parking, stairs, cafe, one or more people, etc. Moreover, these were captured on different days and times to accommodate the same scene with scale and illumination changes. The dataset images have various augmentations, such as scale, rotation, and illumination.
The DS-70K dataset is further divided into ten subgroups (DS1 to DS10) that represent random groups of these images with varied scale, illumination, and capture angles. The subsets contain various image variants and their combinations for comparison. The experiments are modeled and executed on the same machine and dataset to reduce bias.
The experiments are conducted on a Linux-based workstation with an Intel i5 2.53 GHz CPU and 12.0 GB of RAM. The experimental results of image-matching techniques are evaluated and compared in terms of time and accuracy. The experimental input image sizes are 256X256, 240X240, and 220X220, and the image files are in JPG format. The image sizes vary based on the network used. In the experiments, the various datasets are split into training, testing, and validation in the ratio of 70% for training, 15% for testing, and 15% for validation. The experimental results of techniques are repeated thrice, and averages of all parameters are used for various comparisons.

3.6 Experimental Results

The experimental results of the proposed online image deduplication technique have been categorized into the following sections:

- Concurrent Handcrafted Feature Extraction Techniques
- Hot Decomposition Vector Performance
- CNN based fine-tuned AlexNet and VGGNet Techniques
- Performance Comparison of Image Classifiers
- Performance Evaluation of Cross-Domain Net

The following list presents a summary of the experiments conducted:

1. Concurrent handcrafted feature extraction techniques like HAAR, DSIFT, LBR, and PCA-SIFT are implemented with the DS-70K dataset. The results in terms of matching accuracy and time are presented and compared in Figure 3.7 and Figure 3.8.
2. Performance comparison of individual key points feature descriptors like SURF, FAST, MSER, and BRISK are mentioned in Figure 3.9 and in Figure 3.10 using DS-70K dataset. Also given is an HDV image-matching technique that combines SURF key-point features with DWT and DCT-block. The choice of SURF key-point descriptors is validated. The performance of HDV is provided and compared with individual key point feature descriptors (FAST, MSER, and BRISK) with DWT combinations using the DS-70K dataset and are shown in Figure 3.11 and Figure 3.12.
3. To detect exact or near-exact images, fine-tuned AlexNet is implemented, and the outcomes of this proposed network are compared with the original AlexNet and VGGNet. Figure 3.13 and Figure 3.14 depict the performance of CNN-based image detection using the DS-70K dataset of scaled images, illumination images, and rotated images discussed in section 3.5.1 and are compared. The experimental work first used pre-trained AlexNet(trained on the ILSVRC-2012 dataset) and fine-tuned it using Pascal VOC 2007 dataset.
4. Image classifiers are evaluated based on different image variations to detect exact or near-exact images, as discussed in Table 3.1, Table 3.2, Table 3.3. To evaluate the performance of CNN, ten different image subsets DS1-DS10 were taken from the DS-70K dataset (as discussed in section 3.5.1).

5. The proposed network trained with cross-domain images and perturbations is tested in terms of hash computation time, image query time, and hits obtained. The results are presented in Tables 3.4 - 3.11. The different Datasets used are VGG-Flower, Aircraft, Omni, and CIFAR100 along with the PatternNet, and ModelNet40 datasets.

The experiments were conducted meticulously, utilizing diverse datasets and evaluation metrics to ensure robust and comprehensive results for the image deduplication techniques.

3.6.1 Comparison of Concurrent Handcrafted Feature Extraction Techniques

Using the collected dataset of DS-70K images, Figure 3.7 and Figure 3.8 report the performance of concurrent exact or near-exact techniques. The test set consists of 70,000 images, while the matching set contains 50,000 images. The analytical comparison of these techniques is based on image-matching accuracy and time in seconds. It was found that LBR and Haar have better image-matching accuracy than PCA-SIFT and DSIFT for a mixture of scale, illumination, and pose variant images. LBR and Haar also take significantly less time than other image techniques. Although DSIFT and PCA-SIFT almost achieve the same matching accuracy, DSIFT is faster than PCA-SIFT.

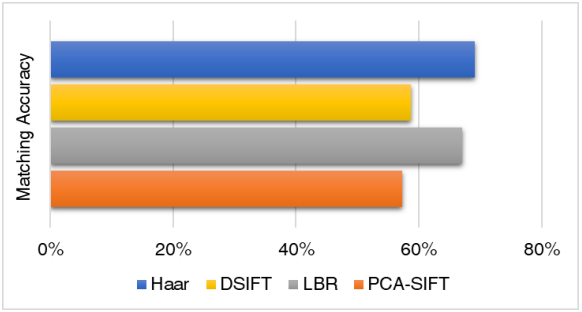


Figure 3.7: Average Accuracy of Concurrent Exact or Near-Exact Techniques

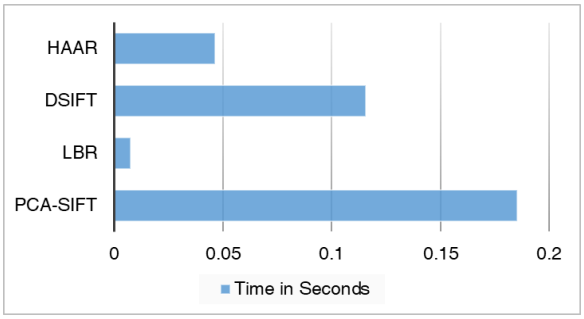


Figure 3.8: Time in Seconds of Some Concurrent Exact or Near-Exact Techniques

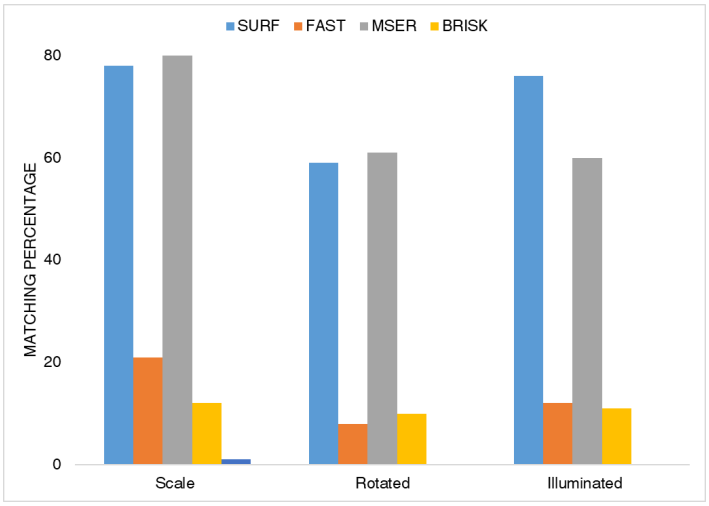


Figure 3.9: Image Matching Accuracy of Individual Key-Point Feature Descriptors on Image Deformation

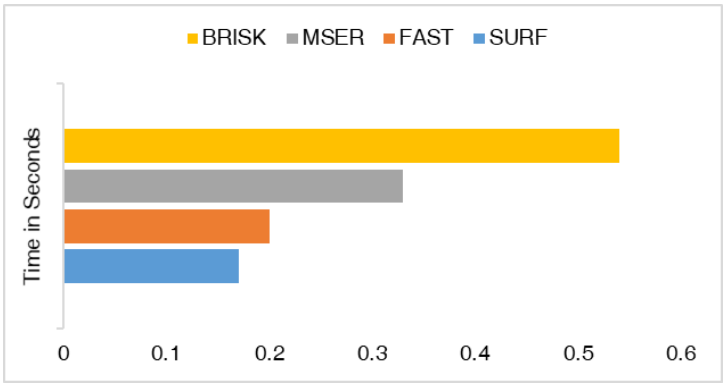


Figure 3.10: Time in Seconds of Individual Key-Point Feature Descriptors on Image Deformation

3.6.2 Hot Decomposition Vector Performance

Figure 3.9 and Figure 3.10 illustrate 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 Figure 3.9, SURF and MSER performed well on all three image deformations scaled, rotated, and illuminated images applied to 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. On the other hand, MSER performed slightly better in scaled and rotated images in terms of image-matching accuracy but took a relatively higher computation time than SURF. BRISK took maximum computation time with the least image-matching accuracy among all the key-point feature extractors. Both FAST and BRISK yield a low number of keypoints. For example, a subset of 20,000 images showed less than 10 points in 80% images for FAST features and 70% images for BRISK features. Therefore, overall SURF outperforms all other feature extraction techniques based on computation time and image-matching accuracy in illuminated, scaled,

and rotated images for individual keypoint feature extraction.

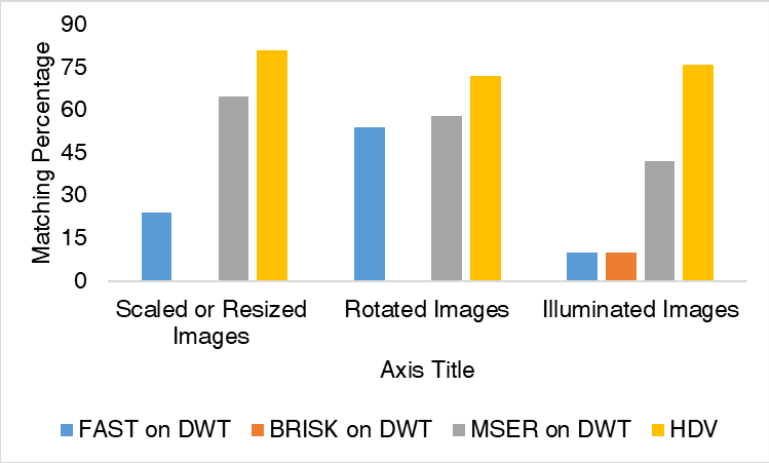


Figure 3.11: Matching Accuracy of Feature Extraction Algorithms with DWT on Image Deformation

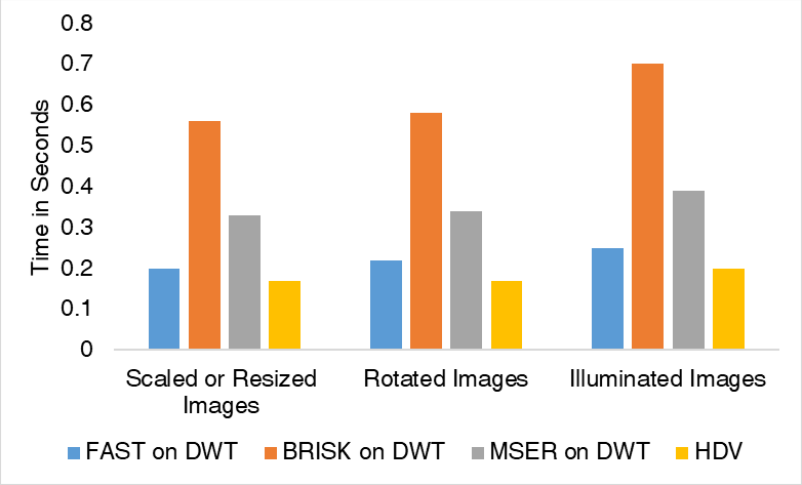


Figure 3.12: Time in Seconds based on Feature Extraction Algorithms with DWT on Image Deformation

Based on the results shown in Figure 3.9 and Figure 3.10, the proposed HDV image matching technique utilizes SURF keypoints and DCT-block. HDV is based on an application of orthogonality on DCT features and SURF on wavelet coefficients. The proposed HDV has performed better in terms of image-matching accuracy in all three image deformations and takes relatively less computation time.

To validate the effectiveness of HDV, an empirical comparison with existing feature extraction methods on DWT is conducted, and the results are shown in Figure 3.11 and Figure 3.12. The results demonstrate that HDV exhibits higher and more stable image-matching accuracy with relatively little computation time for all three types of image deformations. For FAST, Harris, MSER, and BRISK on DWT with a single approximate

coefficient are applied to all three deformations on the same image database. In particular, HDV surpasses other feature extraction algorithms, including FAST, Harris, MSER, and BRISK on DWT, in terms of image-matching accuracy. MSER on DWT comes closest to HDV in image matching accuracy but takes comparatively longer computation time. Although MSER on DWT performs well in scaled and rotated image-matching accuracy, it shows reduced accuracy in illuminated image deformation. On the other hand, BRISK on DWT exhibits very low matching accuracy in illuminated images and no matching accuracy in scaled and rotated images. The computation time of BRISK on DWT is relatively much higher in all three image deformations. FAST on DWT exhibits significantly weaker matching performance than HDV and MSER on DWT. In addition, FAST on DWT shows inconsistent image matching accuracy on scaled, rotated, and illuminated image deformations.

Overall, HDV outperforms the other four feature extraction algorithms depicted in Figure 3.11 and Figure 3.12 for all three types of image deformations, with higher matching accuracy and shorter computation time.

3.6.3 Comparison of Deep CNN-based Feature Extraction

As discussed in section 3.3, Deep-CNN for exact or near-exact image detection has been proposed which is a fine-tuned AlexNet (Net-0). The proposed fine-tuned AlexNet is compared with existing AlexNet and VGGNet in terms of performance. The original features extracted in the fine-tuned AlexNet were 4096, 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 requires more network space and time for online image matching than AlexNet, which is simple and scalable.
Figure 3.13 illustrates the matching accuracy of the CNN feature extractors, while Figure 3.14 presents their performance in terms of computation time in seconds. These techniques were tested on the same dataset, which comprised all three types of deformations.

The empirical comparison is based on image-matching accuracy and computation time in seconds. The results of these techniques are depicted in Figure 3.13 and Figure 3.14. These techniques are applied to the same dataset, which has all three types of deformations: scaled, rotated, and illuminated images as discussed in section 3.5.1. As shown in Figure 3.13 and Figure 3.14, the proposed fine-tuned technique outperformed the existing VGGNet and AlexNet in terms of image matching accuracy. VGGNet has shown

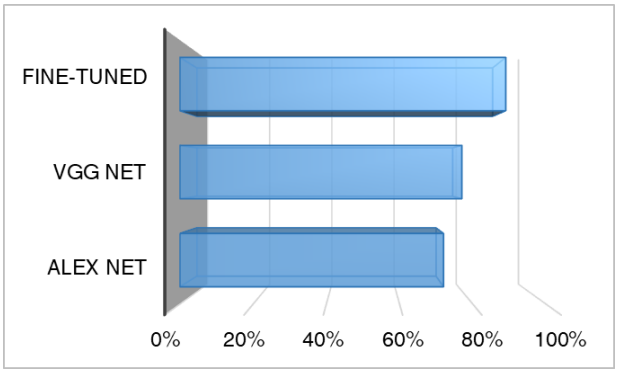


Figure 3.13: Matching Accuracy of CNN Features Extractors

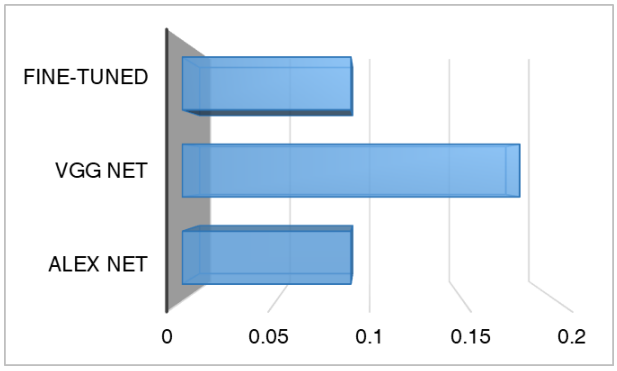


Figure 3.14: Performance of CNN Features Extractors in Seconds

better matching accuracy than AlexNet, with higher computation time than fine-tuned AlexNet.

3.7 Performance Comparison of Image Classifiers

Image classifiers are evaluated based on different image variations to detect exact or near-exact images. The outcomes are presented in Table 3.1, Table 3.2, and Table 3.3. Ten different image subsets with different image sizes, variants, and their combinations are used to evaluate the performance of CNN. The parameters to measure the performance of CNN are based on image size, image variations, and true negative (TN) rate of Top 1 and Top 3 recognition.

Table 3.1: 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%

The image dataset was compared with different subsets DS1-DS10 with varying variants of the image. Table 3.1 demonstrates the performance of CNN features extracted from the fine-tuned net, particularly for scaled, illumination, and rotated image variants. The client accepts or rejects the identified images as depicted in Figure 2.2. Euclidean distance feature classifiers are used to extract the matches of Top-1 and Top-3 image recognition. The True-Negative (TN) in Top-1 and Top-3 recognition provides 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 3.2, three different types of datasets, namely DS1, DS5, and DS10, are chosen randomly with a sample of 850, 1000, and 1400 images with various image variations. The purpose is to compare the image detection performance of distance-based classifiers. Table 3.2 depicts the image detection performance of Class Mean Euclidean, Euclidean, and Bayesian image classifiers. Notably, the Bayesian image classifier outperforms the other classifiers when it comes to handling illumination, pose image variations, and exact images. Hence, the Bayesian image classifier exhibits superior performance in identifying near-exact images.

Table 3.2: Image Detection Performance of Distance Classifiers

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

There is another important issue to be considered for the selection of classifiers. Class Mean Euclidean and Bayesian classifiers require the mean and standard deviation to be stored along with the signatures. On the other hand, the Euclidean distance classifier has no such condition. Table 3.3 depicts True-Negative (TN) variations when the mean and standard deviation are computed with two different sample sizes, 30 and 60, for each class.

In cases where there is no match, and the image signature is to be stored for the first time, the client provides the mean and standard deviation of the query images using different pose augmentations of the image. This means that the user will provide the mean and standard deviation using a query image's illumination and scaled and rotated image variants.

As demonstrated in Table 3.3, DS2 and DS8 show identical results for true negatives with sample sizes of 30 and 60. However, in other circumstances, such as DS6 and DS7, the error rate is lower for a 60-sample size than for a 30-sample size. In Table 3.1, Table 3.2 and Table 3.3, ten different subsets are taken. Clustering is done on a 70,000-image dataset. The primary objective is to map the archive images into the cluster so that images of the same class are close to one another, irrespective of irrelevant characteristics and variations. The pixels in the feature space are clustered together to measure similarity. Here, the similarity is measured using the class mean Euclidean distance classifier, the Euclidean Distance classifier, and the Bayesian Classifier.

Table 3.3: Number of True-Negatives for Bayesian Classifier with a Sample Size of 30 and 60

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%

3.8 Performance Evaluation of Cross-Domain Net

In this section, the datasets used are VGG-flower, Aircraft, OMNI, and CIFAR100, along with the PatternNet, and ModelNet40 datasets, and are discussed in detail. These datasets are used for hash computations and queries.

This section begins by demonstrating the feature computation time, 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 Table 3.4, Table 3.5 and Table 3.6. This chapter involves the use of the VGG-Flower Dataset for this purpose. In this chapter, these features are denoted by Net-1 and Net-2 as shown in Table 3.4.

Table 3.4: 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

The minimum hash computation time in milliseconds using different bits and hash-tables is shown in Table 3.5 for the VGG-Flower dataset. Each entry in the table corresponds to values for (Net-1, Net-2). For instance, for a 16-bits hash length, the minimum hash computation time with Net-1 is 125.4 ms for hash-table 4, while for Net-2, it is 150 ms for hash-table 2. Similarly, Table 3.5 further presents the results for varied hash bits lengths and different numbers of hash-tables. Here, hash-table 2 represents two numbers of hash-tables used in the computation.

Table 3.5: 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)

The query time in milliseconds for Net-1 and Net-2 in the VGG-Flower dataset is shown in Table 3.6. The table presents the query time for different hash bits, with Net-1 and Net-2 features. For 16-bits hash length, the 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 3.6.

Table 3.6: 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)

Two inferences can be made from the results: a) Alex-Net will reduce the time complexity to a certain extent and increases the image retrieval speed. b) The feature dimensions must be reduced further to achieve a significantly lower hash computation time. Notably, the experimental work takes features from the 5^{th} convolutional layer in this study, as discussed in the previous chapter, and performs cross-domain training, including perturbed data. Instead of retraining, the experimental work selects features from fully connected layers to decrease the feature vector size. Due to the requirement of a cross-domain feature extraction network, features from lower levels are chosen in this work.
This experimental work uses 18k original images each from datasets OMNI, ModelNet40, PatternNet, and ImageNet to train the cross-domain net. Table 3.7 shows eight different perturbations used in the cross-domain network, as shown below.

Table 3.7: Eight Different Perturbations for Cross-Domain Dataset

Perturbations	**Cross-Domain Images**
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

A learning rate of .001 is configured in this work to learn the non-shared parameters in each case using a cross-entropy loss with an Adam optimizer for 30 epochs. Table 3.8 shows classification performance metrics using the Omni dataset. This chapter 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 given dataset.

Table 3.8: Classification Performance Metrics using Omni Dataset

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

Table 3.9 and Table 3.10 show the hash computation time and image query time using different combinations. The experiments are conducted on a cross-domain net dataset with varying numbers of hash bits and hash tables. The datasets VGG-flower, OMNI, CIFAR100, PatternNet, and Aircraft are used for this purpose. Table 3.9 displays the average and minimum hash computation times in milliseconds using a different number of bits and different hash-tables when employing a cross-domain net. For a 16-bit hash length, the lowest average hash computation time is 15.4 milliseconds for hash-table 4, and the lowest minimum hash computation time is 14.7 milliseconds for hash-table 2 and hash-table 4.

Table 3.9: 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 Min= 14.7	Avg = 19.1 Min= 14.7	Avg= 15.4 Min=15.1	Avg= 21.2 Min= 15.9	Avg= 22.8 Min= 16.7
24	Avg= 19.8 Min= 16.3	Avg= 19.8 Min=17	Avg= 22.4 Min=18	Avg= 23.1 Min=19	Avg= 24.6 Min= 23.7
32	Avg= 22 Min= 17.5	Avg= 22.6 Min= 19.8	Avg= 25.6 Min= 19.8	Avg= 26.9 Min=19.8	Avg= 29.5 Min=27.8
64	Avg=37.8 Min= 31	Avg=39.4 Min= 32	Avg= 45.1 Min= 34.1	Avg= 48.3 Min= 34.9	Avg= 53.8 Min= 49.2
128	Avg=38.5 Min=31.3	Avg= 39.4 Min= 31	Avg= 47.6 Min= 35.7	Avg= 50.1 Min= 38.9	Avg= 60.6 Min= 56.4

Table 3.10 shows the average and minimum query time for cross-domain net. Specifically,

for 16-bit hash length, hash-table 2 exhibits the lowest average query time of 216.8 milliseconds and the lowest minimum query time of 40 milliseconds. Table 3.10 also includes the results of other hash bits with a different number of hash tables.

Table 3.10: 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 Min= 40	Avg= 451.7 Min= 90	Avg= 776.1 Min= 170	Avg= 2133.4 Min=420	Avg= 2897.2 Min= 1040
24	Avg= 19.8 Min= 10	Avg= 20.6 Min= 20	Avg=59.3 Min= 20	Avg= 87 Min= 30	Avg= 200.9 Min= 80
32	Avg= 5.5 Min= 10	Avg=7.9 Min=4	Avg=11.1 Min=4	Avg= 20.6 Min= 20	Avg=30.9 Min= 15
64	Avg=18.2 Min=9.3	Avg= 26.1 Min=18.7	Avg= 28.5 Min=30	Avg= 34.8 Min=24.8	Avg= 49.1 Min=30
128	Avg= 22.2 Min=14.5	Avg= 21.4 Min= 14.5	Avg= 30.9 Min= 20	Avg=45.9 Min=35	Avg= 77.5 Min= 68

Table 3.11: 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

Table 3.11, presents the number of hits obtained while querying different perturbed images, showcasing three distinct cases. In the first case, a signature database with perturbed images is created and then queried. In cases 2 and 3, signatures are generated using the original and queried perturbed images. However, in case 2, the w/o_P version of the network is used, while in case 3, the w_P cross-domain net is utilized, yielding the best results.

Many interesting results are reported in [275] using a ResNet architecture for feature

extraction, indicating that the number of misses increases with varying perturbations. The experimental work in this chapter focuses primarily on the fact that cross-domain learning with a perturbed dataset can handle this impact to a significant extent.
Observations from Table 3.11 indicate that a lower number of hits are obtained when indexed for solarization and sharpening, as observed in case 3. Numerous experiments conducted with varying parameters consistently yield the same results. It's important to note that blur, compression, and Gaussian noise can be varied in the image. A lower impact of these perturbations leads to higher hit rates, while a higher impact results in fewer hits. For instance, with a variance of 0.005 for Gaussian noise, the hit is 0.96. While training the cross-domain net, these parameters are fixed. A blur window of 15 pixels is taken, along with a compression ratio of 50% and a Gaussian noise variance of 0.1. The hits shown here use the same extremes. If these are lowered, then the hits increase. Although good classification accuracy with blurred images is obtained in this work, 57% hits are returned when indexed. This, indeed, is a very interesting observation and requires more attention.
To address this situation, one approach is to perform filtering operations before feature computation. Additionally, denoising noisy Gaussian images can further increase the hit rates. However, since it is unknown beforehand whether the query image is blurry or noisy, two approaches can be considered. One involves building a classifier to identify blur or noise and applying an appropriate deblur or denoising filter. The choice of the deblur or denoising filter will depend on the specific data, as filters effective on normal images may not work well on medical images [276, 277, 278]. A more efficient approach would be to pass all images through a network, preferably a cross-domain one trained to clean noisy or blurry images while preserving other images unchanged. This approach allows the network to intelligently handle various perturbations without prior knowledge of the specific perturbation present in the query image. Consequently, this approach offers a more versatile and robust solution.

3.9 Conclusion

The chapter introduces a novel deep CNN-based image detection technique for online deduplication, aimed at detecting exact or near-exact images. Further, the chapter presents Hot Vector Decomposition (HDV) as a means of generating image patches for near-exact images, which helps to reduce storage requirements further. Experimental details and testing results of the proposed approach are presented in this chapter, with various performance metrics used to validate its effectiveness in detecting duplicate images. The chapter begins by validating the results of concurrent handcrafted features

and HDV feature extraction techniques through evaluated performance metrics. Subsequently, a CNN-based fine-tuned AlexNet is evaluated and compared with existing Alexnet and VGGNet techniques. Moreover, various image classifiers are evaluated based on different image variations to detect exact or near-exact images.
The next chapter introduces EsDeDUP, a novel mage deduplication technique for energy-saving. The technique detects duplicate images and determines energy usage in terms of CPU computation and performance analysis of scalable image deduplication techniques.

Chapter 4

EsDeDUP: Proposed Image Deduplication Technique for Energy-Saving

The previous chapter details the proposed online image data deduplication technique designed for exploring exact or near-exact duplicate images in a cloud storage system using CNN and traditional feature extraction techniques. The performance of the proposed method has been evaluated based on image matching accuracy and the time required to match the images.

In this chapter, a novel EsDeDUP image deduplication technique for energy-saving has been proposed to compute the energy consumption for deduplication. The proposed technique analyzes the impact of image deduplication techniques on energy-saving, accuracy, and storage reduction using real datasets for empirical analysis. The research investigated the performance of hash-based, i.e., pHash, wHash, aHash, and dHash-based and CNN-based fine-tuned AlexNet, fine-tuned VGG-NET-16, and fine-tuned VGG-NET-19 image deduplication techniques to detect exact or near-exact duplicate images. Extensive experiments on real datasets evaluate the accuracy, storage reduction, and energy-saving of hash-based and CNN-based techniques using the EsDeDUP technique.

The chapter is organized as follows: Section 4.1 provides a brief overview of exact and near-exact image deduplication. Section 4.2 presents the proposed EsDeDUP technique, which analyzes the energy-saving and performance of CNN-based and image hash-based deduplication techniques for detecting exact or near-exact images. This section also describes the proposed image deduplication technique for energy-saving, encompassing three energy consumption levels before, during, and after deduplication for a given workload. The experimental setup and performance evaluation of Hash-based and fine-tuned CNN-based deduplication techniques are discussed in section 4.3. Two real image datasets were used to assess the effectiveness of these techniques in detecting exact or near-exact images and their impact on energy saving followed by a conclusion in section 4.4.

4.1 Exact or Near-Exact Image Deduplication

With the rapid advancement of storage technology, millions of images are uploaded and saved on the Internet each second [279]. As mentioned earlier in chapter 3 of section 3.1, deduplication is an optimization technique that reduces the amount of energy and cost needed to store and process a large scale of images. However, deduplication for such a massive dataset is challenging and complex. Thus, it is important to analyze the energy consumption, savings, and performance achieved by such a scalable deduplication technique for exact or near-exact images [280].
The traditional Hash-based techniques are still employed to get the duplicate images. This hashing-based image deduplication technique utilizes hashing for chunking, hash calculation, fingerprint matching, index lookup, and chunk storage. The primary objective in designing hash functions is ensuring that similar images are assigned binary codes, as explained in Section 2.5.2 of Chapter 2. This process involves converting high-dimensional visual data into a lower-dimensional binary space.
The RGB images are converted to downscaled greyscale images to reduce resolution and simplify the matching process. Each pixel of the down-sampled image is converted to a single binary value, either 0 or 1. The resulting binary matrix has flattened image values, reflecting the original image features. The calculated hash values are stored within a hash table to facilitate efficient query lookups. Near neighbor techniques, such as hamming distance, are used in pairs to determine the similarity between two image hashes by comparing the closeness of matched bits [281, 282]. A predetermined threshold determines the maximum number of dissimilar bits allowed in the hash code of near-duplicate images. Based on this, duplicate images are eliminated, and only the unique image is saved. In these, the cost related to hashing, storage, and indexing has been factored in hash-based image deduplication technique experiments. Figure 4.1 presents a hash-based exact and near-exact image deduplication technique.

In several fields of computer vision, cutting-edge deep neural network architectures have been deployed. These techniques are used to classify images according to their content features. Deep neural networks have achieved significant success in various areas including computer vision, natural language processing, natural language processing, and recommendation system, etc. Popular Deep Learning architectures include Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Deep Belief Networks (DBNs). Training deep neural networks often requires substantial data and computational resources. The emergence of cloud computing and specialized hardware, such as GPUs, has facilitated the training of deep neural networks.
Recent developments illustrate that Deep convolutional neural networks (CNN) can

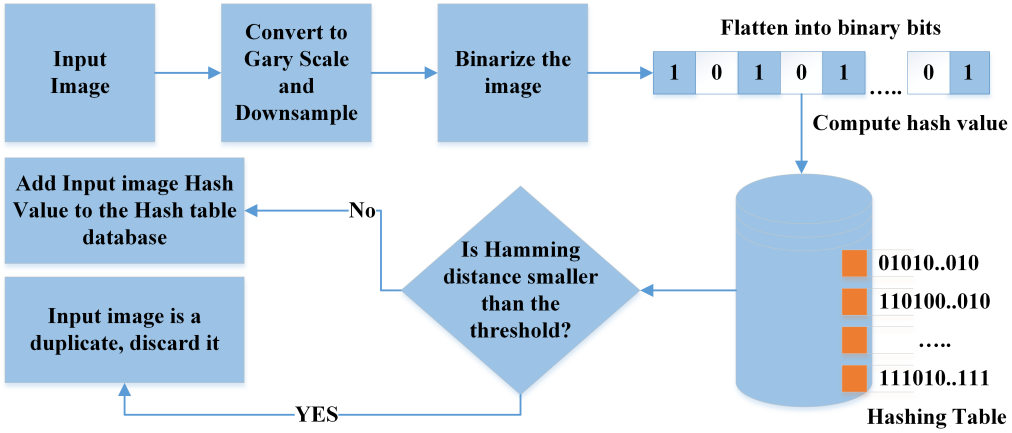


Figure 4.1: Hash-based Exact and Near-Exact Image Deduplication Technique

learn rich mid-level representations [283] that are valuable for image identification, object recognition, machine vision, and feature extraction in the field of computer vision [284, 264, 265, 285, 286, 287]. These strategies scale effectively for massive data sets containing multiple image categories. Moreover, CNN models trained on a large dataset can perform a new task in a different domain and extract more relevant and meaningful information from images [132, 288]. The supervised domain, which preserves deep hashing for learning binary codes from labeled images, was utilized to obtain duplicate images. A collection of latent properties or binary codes represents image labels. The hash functions act as a hidden layer between image representations and classification outputs, and binary codes are learned by minimizing an objective function over classification error [283]. Mobile devices typically have limited computation and memory resources, which pose challenges for storage optimization tasks like deduplication. To address this, public mobile images can leverage cloud-based deduplication applications and storage. Image features are detected on the mobile device and sent to a CNN-based deduplication model as a query image to identify exact or near-exact duplicates, which are then managed accordingly. For private images stored in the cloud, the process remains similar, as it checks the private image space for duplicates using the described procedure. To detect duplicates stored directly on the mobile device, lightweight CNN models such as MobileNet are used. Several applications, including Google Files Go, Gallery Doctor - Photo Cleaner, and Clean Master, offer such functionality.

This chapter proposes EsDeDUP, a novel image deduplication technique for energy-saving, to compute the effect of energy-saving for deduplication. The energy consumption with and without duplicates and energy-saving and performance analysis of scalable image deduplication techniques have been proposed. The significant contributions of this chapter are listed below:

- Proposed EsDeDUP, a novel image deduplication technique for energy saving. These deduplication techniques have different approaches to identify duplicate images and have varied requirements of computation, storage, memory, and energy for processing. These techniques exhibit different levels of accuracy. So a common architecture is needed to evaluate the energy-saving and performance. To address this open issue, this chapter proposed EsDeDUP technique that characterized the energy-consuming units of deduplication techniques. It also proposes an energy-saving computation model for deduplication techniques on common datasets and workload.
- This chapter evaluated the performance of four hash-based pHash, wHash, aHash, and dHash-based deduplication techniques discussed in Section 2.5.2 in Chapter 2 and their effect on energy-saving. The energy savings achieved by these techniques are presented using the proposed EsDeDUP technique to extract the exact and near-exact duplicate image on a common dataset and workload.
- The chapter proposed fine-tuned CNN-based deduplication techniques. The neural structures of fine-tuned AlexNet, fine-tuned VGG-NET-16, and fine-tuned VGG-NET-19 are proposed to extract the exact and near-exact duplicate images. The performance and energy-savings of fine-tuned CNN-based deduplication techniques are presented using the proposed EsDeDUP technique to extract duplicate images.
- The extensive experiments of hash-based and fine-tuned CNN-Based deduplication techniques for performance evaluations and energy-saving were achieved using two real image datasets. The result shows that the fine-tuned CNN-based deduplication techniques have shown better accuracy with higher energy consumption than hash-based deduplication techniques to detect exact and near-exact duplicate images.

4.2 Energy Efficiency using Data Deduplication in Cloud Storage System

Deduplication techniques make the storage system more cost and resource-effective by eliminating or reducing data duplication. This optimization technique also reduces the energy and cost needed to store and process large-scale data. The deduplication process for such a massive dataset is challenging and complex. Thus, analyzing the energy consumption, savings, and performance achieved by such scalable deduplication techniques is important. There is a gap in such an analysis, which impacts the energy and cost of a storage system. So, researchers need to devise a common framework to analyze the impact of the deduplication technique on energy saving.

This section discusses the proposed EsDeDUP technique to analyze the energy-saving and performance of CNN-based and hash-based techniques employed in image deduplication. This section discusses the related work on these techniques. This section also discusses the proposed technique for Deep CNN-based duplicate Image detection and presents the fine-tuned AlexNet, VGGNNet16, and VGGNet19 for exact or near-exact image detection.

4.2.1 Deep CNN-based Duplicate Image Detection

Deep CNN-Based Duplicate Image detection introduced the latent layer for binary code between the fully connected layers. The binary code will reduce the high-dimensional vector comparison during exact and near-exact duplicate image detection. CNN classification network is a pre-trained network on the same domain dataset to extract mid-level image representations. In this chapter, a latent layer is proposed that has neurons to learn binary code representations. The binary code will reduce the high dimensional vector comparison during exact and near-exact duplicate image detection. Figure 4.2 shows the technique for deep CNN-based duplicate image detection.

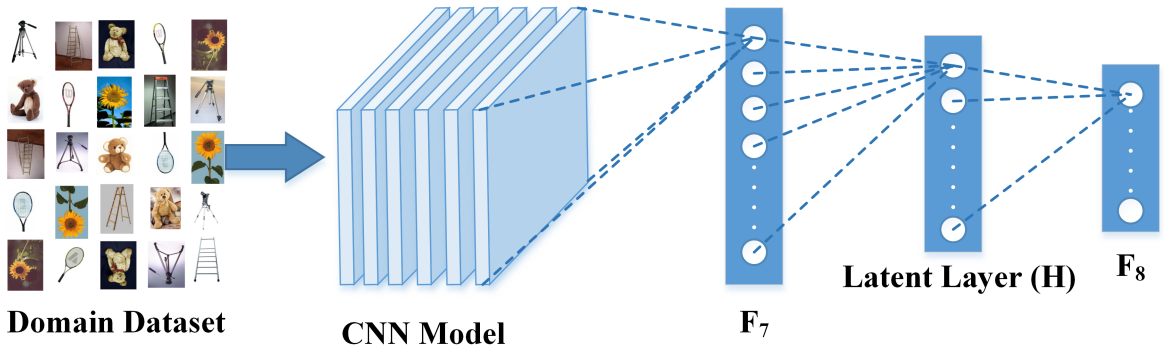


Figure 4.2: Architecture of Fine-tuned CNN Models with Latent Layer to Learn Binary Codes and to Detect Exact or Near-Exact Images.

The fine-tuned architectures of Alex Net, VGGNet16, and VGGNet19 are presented in Figure 4.3, Figure 4.4, and Figure 4.5. A latent layer H has been added between the F7 and F8 layers of the deep neural network structure mentioned above. The fine-tuned structures are mentioned. The F7 layer consists of 4096 features, the latent layer H has 1024 features, and the F8 layers contain 1024 features. The latent layer is introduced to exploit the learned binary codes, and F7 features to detect exact or near-exact images. The exact and near-exact duplicate image to the query image via coarse-rained and fine-grained search is mentioned in Fig 4.6. This experimental work uses a pertained Alex

Net, VGGNet16, and VGGNet19 CNN models with ImageNet Dataset. The dataset consists of 1.2 million images categorized into 1000 classes. The VGG network and AlexNet models are used, which are pre-trained models trained originally from AlexNet and VggNet architectures. The image with a similar binary code will have the same image label in a dataset. However, these high-dimensional image representations require high operational computation. The binary code can be extracted from the hidden layer based on the extracted features of the F7 layer. The sigmoid activation is used to obtain the activation of neurons in hidden layer H. The dataset-specific binary codes are learned for accurate and efficient exact and near-exact duplicate image detection.

The duplicate images are extracted at two levels for accurate and efficient duplicate image detection, represented here as coarse-grained and fine-grained levels. At a coarse-grained level, the hidden layer features of query image Q are extracted and represented by F(H), which is denoted by Out(H). The number of bits in this code equals the number of nodes in the hidden layer here, using 1024 for uniformity. An ith bit of binary code of an image is computed and represented by a threshold value greater than or equal to 0.5.

$$H_i = \begin{cases} 1 & \text{Out }_i(H) \geq th \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

The binary codes are then generated from all the input images in the dataset. The binary code obtained from the query image Q is compared with the existing dataset of binary codes of images based on Hamming Distance (HD). HD less than the threshold means the query image is an exact or near-exact duplicate of an image in the dataset. Later, at a more fine-grained level, the similar image features extracted from layer F7 form a poll. The query image feature vector F_{vq} is compared with the feature vector F_{vi}^{Pool} of ith pool images using Euclidean distance (ED), represented by Sim_i. Sim_i is indexed, and a smaller value of ED means more accurate duplicate detection of images.

$$Sim_i = \left\| F_{vq} - F_{vi}^{Pool} \right\| \tag{4.2}$$

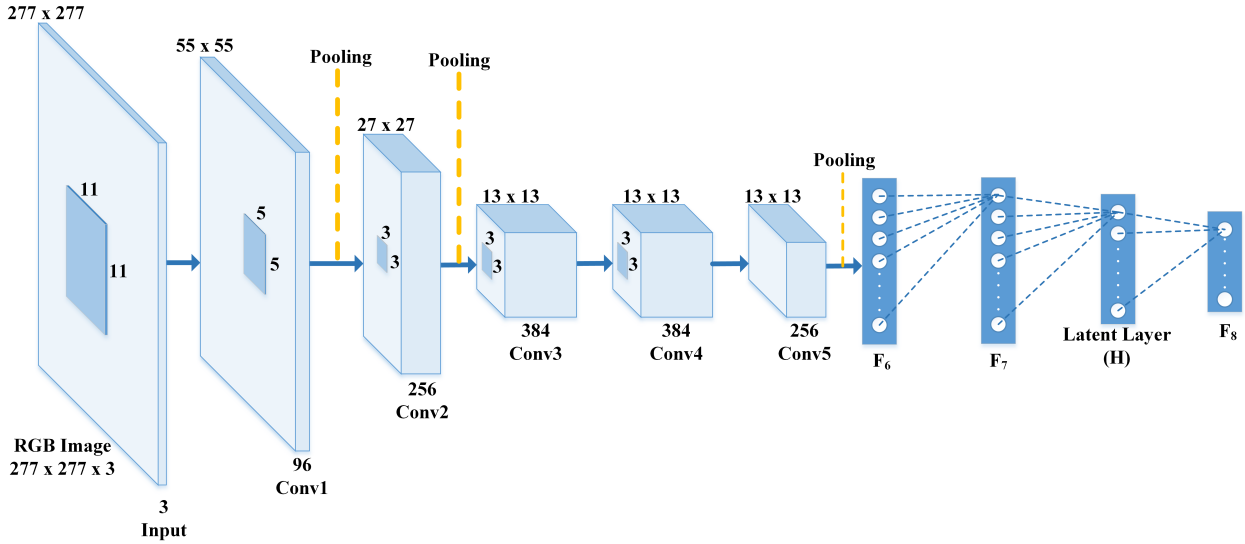


Figure 4.3: Fine-tuned AlexNet to Learn the Binary Codes and F7 Features to Detect Exact or Near-Exact Images

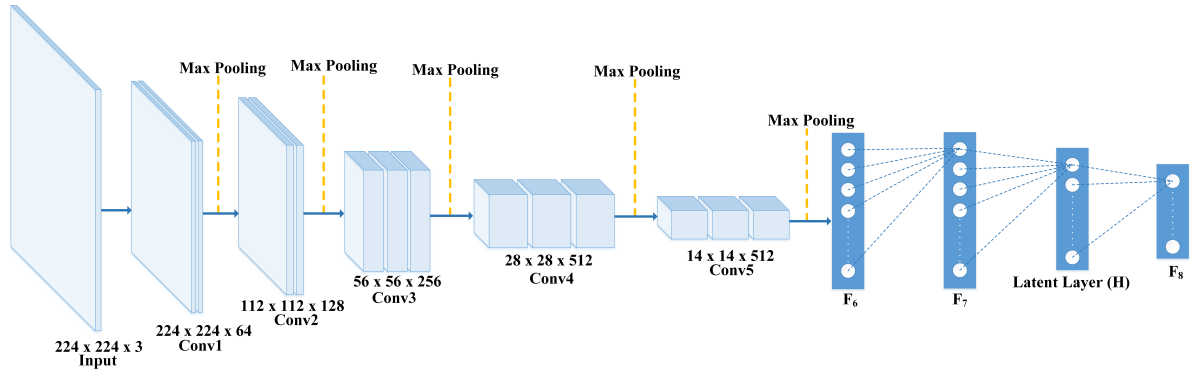


Figure 4.4: Fine-tuned VGGNet16 to Learn the Binary Codes and F7 Features to Detect Exact or Near-Exact Images

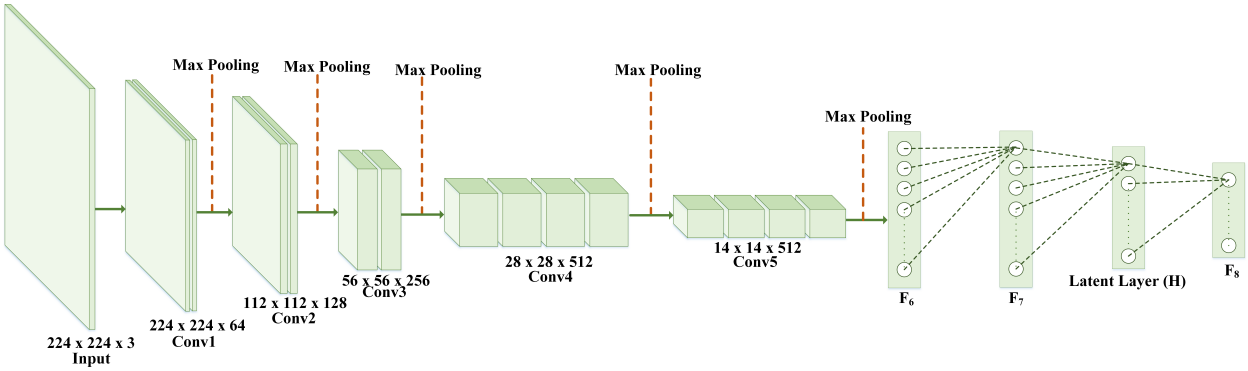


Figure 4.5: Fine-tuned VGGNet19 to Learn the Binary Codes and F7 Features to Detect Exact or Near-Exact Images

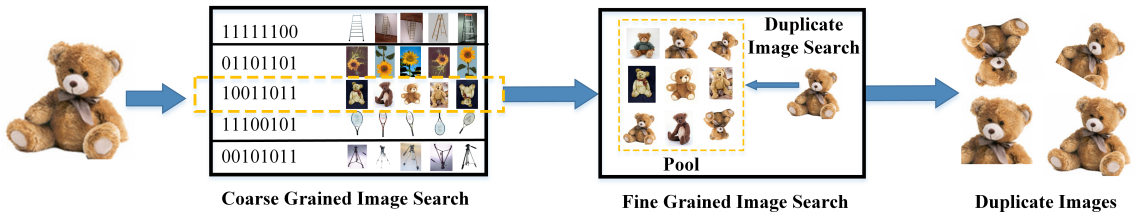


Figure 4.6: Coarse-grained and Fine-grained Search of Exact and Near-Exact Duplicate Image Detection.

4.2.2 Image Hash for duplicate image detection

This chapter utilized pHash, aHash, dHash, and wHash image deduplication techniques to examine energy savings and performance, as described in section 2.5.2 of chapter 2 in the thesis. These hash-based techniques have their own unique characteristics for extracting the image features needed to compute the hash code of images. For uniformity, this chapter uses the 32x32 image size, which translates to 1024-bit hash codes or 1024 features in the hash code of the images. Figure 4.7 represents a sample image from the Caltec-256 dataset and its grayscale 32x32 and 32x32 bit hash representations of the image's pHash, aHash, dHash, and wHash.

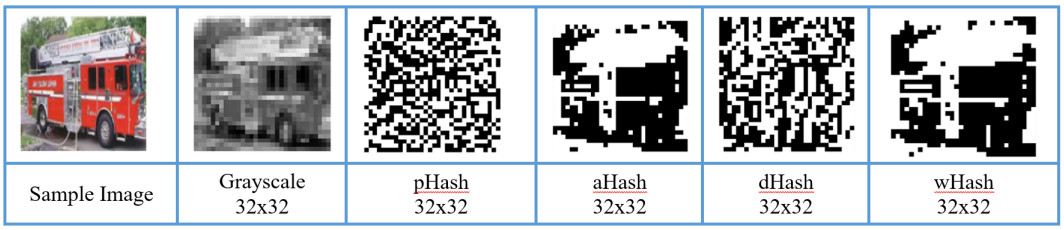


Figure 4.7: Image from the Caltech-256 dataset, grayscaled and resized to 32x32 and 32x32 image representation using pHash, aHash, dHash and wHash techniques

4.2.3 Proposed Image Deduplication Technique for Energy-Saving

To measure the effectiveness of image deduplication techniques on energy saving and storage cost, the chapter proposed EsDeDUP, a novel image deduplication technique for energy-saving. The proposed EsDeDUP technique comprises three levels of energy computation before, with, and after deduplication. Energy is mainly measured from the CPU package, DRAM, I/O interrupts, CPU core, GPU, or any other processes having a fork, exec, or exit state. Table 4.1 represents energy consumed by different processes and their description.

Fig 4.8 depicts an EsDeDUP technique for energy-saving, cost, and performance analysis of scalable exact or near-exact image duplicate detection in cloud storage system. The

Table 4.1: Energy Consumed by Different Processes and their Description

Notation	Description
E_{CPUi}	Energy consumed by CPU packages and CPU cores at i^{th} time interval.
E_{DRAMi}	Energy consumed by DRAM at i^{th} time interval.
E_{IOi}	Energy consumed by I/O interrupts at i^{th} time interval.
E_{GPUi}	Energy consumed by GPU at i^{th} time interval.
E_{BD}	Energy consumed to read images from a fixed workload before applying deduplication techniques.
E_{BDi}	Energy consumed to read images from a fixed workload before applying deduplication techniques at an ith time interval.
E_{D}	Energy consumed to execute the deduplication technique.
E_{Di}	Energy consumed to execute the deduplication technique at an ith time interval.
E_{AD}	Energy consumed to read images from a fixed workload after applying deduplication techniques.
E_{AD}	Energy consumed to read images from a fixed workload after applying deduplication techniques at an ith time interval.

EsDeDUP technique components exploit parallel and distributed techniques and can be employed on distributed data. Due to these distributed components, the EsDeDUP is scalable and can be employed on large data and dynamic storage systems to extract duplicate data. EsDeDUP analyses the impact of image deduplication techniques on energy savings and storage reduction. Specifically, the study examines the performance of CNN-based fine-tuned AlexNet, fine-tuned VGGNet16, and fine-tuned VGGNet19, alongside pHash, wHash, aHash, and dHash based deduplication techniques, in extracting exact and near-exact duplicate images.

The proposed technique in this chapter has been divided into three levels. The stages are discussed below:

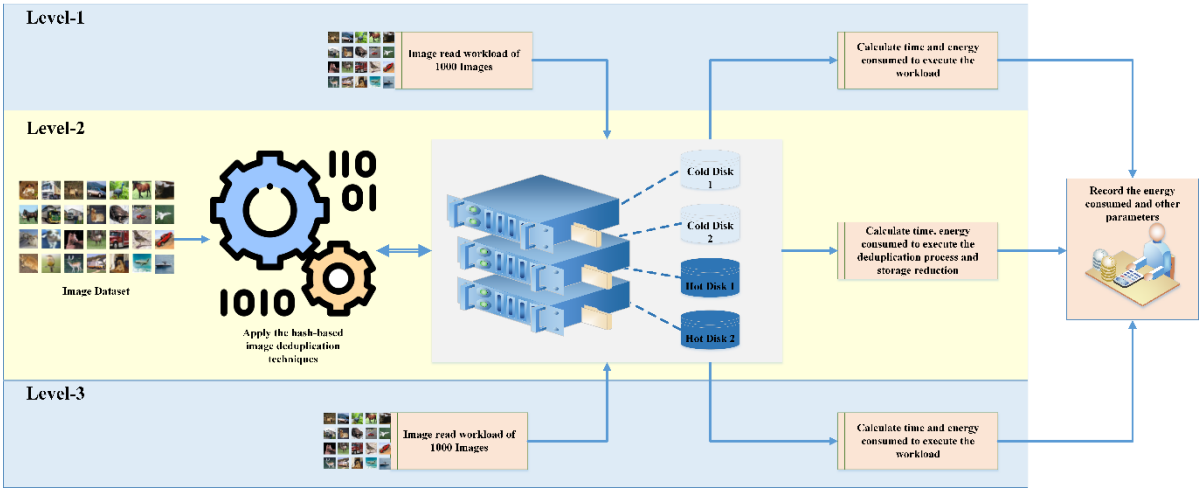


Figure 4.8: Proposed Energy-Saving Image Deduplication Technique for Scalable Exact or Near-Exact Image Duplicate Detection in Cloud Storage System

In the proposed technique, Level 1 involves measuring the energy consumption on a fixed workload before applying deduplication. At this level, a fixed workload of 1000 read images is applied, and measure the energy consumed to execute the read images workload. This energy consumed is donated by E_{BD}.

$$E_{BD} = \sum_{i=1}^{t} E_{BDi} \tag{4.3}$$

where $E_{BD} = E_{CPUi} + E_{DRAMi} + E_{IOi} + E_{GPUi}$

Here i represent the time taken to read image workload in seconds such that $i \in \{1, 2, 3, \ldots t\}$.

Storage Reduction: Deduplication is a storage optimization technique that removes exact or near-exact images from the storage system. The images are replicated on different servers for scalability and availability with a replication factor of at least three. Such a system has large duplicate images. So any reduction in image data has an enormous impact on storage reduction. This chapter analyses image storage reduction by applying contemporary CNN-based and hash-based image deduplication techniques.

In level 2 of the proposed EsDeDUP technique, CNN-based and hash-based image deduplication techniques are applied to the public dataset to get exact and near-exact duplicate image matches and remove duplicate images stored on the server. The energy consumed by these techniques on various datasets is analyzed in this stage of EsDeDUP.

The contemporary deep CNN-based and hash-based deduplication techniques are applied to publicly available real datasets to detect duplicate images. The image hash-based techniques generate hash codes for images based on image features. These techniques generate

and store the image hash code in a database. Later the pair-wise hash code matches using Euclidean or Hamming distance to get the exact and near-exact duplicates.
The contemporary feature extraction and duplicate image detection techniques exploit deep CNN structure to execute the image feature comparison and get exact and near-exact duplicates. Such deep CNN-based image deduplication techniques use complex operations for image feature matching and consume additional time and energy from servers for computation. So, CNN-based deduplication techniques require high computation, memory resources, and energy consumption. The energy consumed by these techniques on various datasets is analyzed at this level of EsDeDUP. The total energy consumed by deduplication on a dataset is represented by E_D and is mentioned in the equation below:

where $E_D = E_{\text{CPUi}} + E_{DRAMi} + E_{\text{IOi}} + E_{GPUi}$

Here i represents the time taken to execute the deduplication technique in seconds such that $i \in \{1, 2, 3, \ldots t\}$. E_{CPUi}, E_{DRAMi}, E_{IOi} and E_{GPUi} are measured from level 2 of the proposed EsDeDUP technique and other parameters like time, accuracy, etc. are stored for further analysis.
Level 2 of the proposed EsDeDUP technique evaluates the energy usage of CPU and I/Os by utilizing the RAPL module to determine the energy consumption of the dataset obtained when applying the deduplication approach as duplicate images are identified and removed in level 2 of the proposed technique. The effect of deduplication on storage reduction, energy consumption, and energy saving is analyzed at this level. The dataset without duplicate images reduces the data's size, storage requirements, and energy consumption.
Similar to level 1 of the proposed technique, the energy consumption on a fixed workload after the deduplication technique is investigated in level 3. The same workload of 1000 read images is executed, and measure the energy consumed to execute the read images workload. This energy consumed is donated by E_{AD}.

4.3 Experimental Setup and Performance Evaluation

This section discusses the experimental results of EsDeDUP employed on hash-based and deep CNN-based image deduplication techniques. These deduplication techniques differ in terms of extraction strategies, accuracy, and energy consumption. These techniques have been employed in EsDeDUP to compare accuracy, storage reduction, and energy

consumption. The experiments have been focused on these parameters. The experiments are executed on an Ubuntu 20.04 server having 12 Core Intel Xeon Silver 4116 2.1 GHz, 128 GB DDR4 RAM, 8 GB GDDR5 graphic card, 8*1.2 TB 10K RPM SAS drives. Python, its allied libraries, and TensorFlow are used for implementation and evaluations.

4.3.1 Datasets Used

In this experimental work, fine-tuned CNN-based Alexnet, VGGNet16, and VGGNET19 are employed for performance comparison of their energy usage, time, and accuracy. The hash-based deduplication techniques dHash, aHash, pHash, and wHash are also applied for performance comparison of their energy usage, time, and accuracy. The experimental work in this chapter has used two datasets for the performance evaluation, and its characteristics are mentioned in Table 4.2. The Caltech-256 dataset consists of 30,607 images categorized into 256 different object categories. The Caltech-256 dataset is derived from the original Caltech-101. Similarly, The ImageNet-Mini is a derivative of the original ImageNet, having 1000 classes and nearly 38.7K images. For uniformity in experiments, all dataset training and test samples are resized to 256 x 256 These resized images are further resized according to the requirement of the fine-tuned CNN models image inputs dimensions as shown in Table 4.3.

Table 4.2: Data Sources used in Experiments

Image Dataset Source	**Number of Categories**	**Number of Images**	**Size in GB**
Caltech-256	256	30.6K	1.2
ImageNet-Mini	1000	38.7K	4.24

Table 4.3: Hyper-parameters of CNN-Based Models

Model Name	**AlexNet**	**VGGNET16**	**VGGNET19**
Input Image size	227 x 227	224 x 224	224 x 224
Channels	3	3	3
Activation Function	ReLU	ReLU	ReLU
Batch size	128	256	256
Dropout	0.5	don't use	don't use
Layers	8+1	16+1	19+1
Number of nodes	60 millions	138 millions	140 million

For uniformity and simplicity, the experimental work augmented the dataset images with variation to test the accuracy of CNN-based and hash-based deduplication techniques,

and each augmented variation and the number of images produced are shown in Table 4.4. This augmented dataset increases the base dataset with more near-duplicate images having horizontal flip, vertical flip, slight rotation, and zoom. This work augmented 0.5 million images with these augmented operations and added them to the same dataset as shown in Table 4.5. The augmented datasets are closer to real-world datasets used for experiments to obtain the deduplication accuracy, energy consumption and storage reduction of CNN-based and hash-based deduplication techniques. These augmented datasets are uniformly used for all EsDeDUP experiments and results evaluation for CNN-based and hash-based deduplication techniques. All experiments employ 80% training data and 20% test data. The proposed CNN-based deduplication techniques and their hyperparameters are also employed.

Table 4.4: Augmenter Operations on Caltech-256 and ImageNet-Mini

Sr. No.	**Augmenter operations**	**Description**
1.	Random rotation	Left and right rotation randomly between 0 to 10%
2.	Horizontal flip	Random left/Right
3.	Vertical flip	Top to bottom
4.	Rotate 90°	90°Rotation
5.	Zoom	Random zoom between 110% - 160%

Table 4.5: Augmented Datasets for Experiments

Dataset Augmented	**Number of Categories**	**Number of Images**	**Size in GB**
Caltech-256	256	50000	1.2
ImageNet-Mini	1000	50000	4.24

4.3.2 Energy Consumption Evaluation of Deduplication Techniques

The experiments conducted on the server consume continuous energy to run its normal internal operation. This work measured the internal server energy consumption per second over a period of 14 days and presented the average energy consumption per second on each day as shown in Figure 4.9.

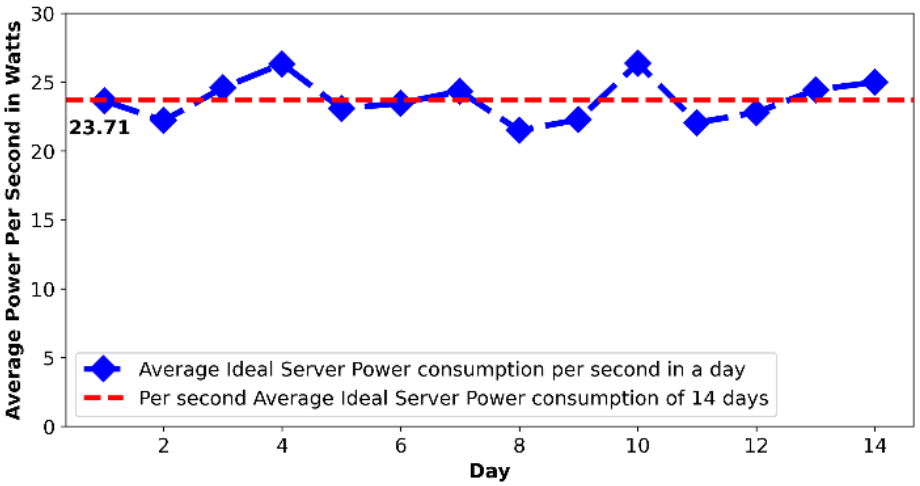


Figure 4.9: Average Ideal Internal Server Energy Consumption Per Second of Continuous 14 days.

In E_sDeDUP is energy E_I is measured in watts per second and is represented as ideal internal operations server energy consumption at i^{th} second. The average of all 14 days is E_I=23.71 watts per second to energy its internal operations. Furthermore, the server continuously consumes average energy of 23.71 watts per second or nearly 85.4 kilowatts per hour for its internal components. This energy depends on the server hardware specifications and operating system kernel operations.

4.3.3 Performance Evaluation of CNN-based and Hash-based Deduplication Techniques

The performance is measured through the accuracy of fine-tuned AlexNet, VGGNet-16, VGGNet-19 and hash-based deduplication techniques using two augmented Caltech-256 and augmented ImageNet-Mini datasets. In this chapter, the experimental work has employed contemporary deduplication techniques and obtained accuracy, energy usage, and storage reduction using two datasets. The energy usage of CNN-based and hash-based techniques is evaluated and analyzed by applying different numbers of images from augmented datasets. The experimental results on the accuracy, energy usage, and storage reduction are presented and discussed in this section.

4.3.3.1 Accuracy of CNN-based and Hash-based Techniques

The Top-1 and Top-5 accuracy of trained CNN-based models are presented in Table 4.6 using a dataset of the Caltech-256 and ImageNet-Mini on test datasets. The fine-tuned VGGNET16 and VGGNET19 perform better than AlexNet in terms and accuracy of

Top-1 and Top-5 Images on both datasets.

Table 4.6: Accuracy of Fine-tuned CNN-based Models using Augmented Caltech-256 and ImageNet-Mini Datasets

	CNN-Based Model	**Fine-tuned AlexNet**	**Fine-tuned VGGNet-16**	**Fine-tuned VGGNet-19**
Augmented Caltech-256 dataset	**Top-1 Accuracy**	52.1	72.2	71.7
	Top-1 Error	47.9	27.8	28.3
	Top-5 Accuracy	82.4	92.8	92.3
	Top-5 Error	17.6	7.2	7.7
Augmented ImageNet-Mini dataset	**Top-1 Accuracy**	60.6	76.7	76.1
	Top-1 Error	39.4	23.3	23.9
	Top-5 Accuracy	83.1	93.2	92.8
	Top-5 Error	16.9	6.8	7.2

The Top-1 accuracy and Top-5 accuracy of Hash-based models are presented in Table 4.7 using a test dataset of augmented Caltech-256 and augmented ImageNet-Mini. The dHash and aHash techniques perform poorly on exact or near-exact image detection as compared to the pHash and wHash due to their DCT-based hash image feature capturing techniques. For consistency, 1024 bits of image features are used for all hash-based techniques. The DCT-based pHash technique exhibits slightly high accuracy in detecting exact or near-exact images as compared to wHash and is more robust to image transformations such as cropping, scaling, and rotation.

Table 4.7: Accuracy of Hash-based Techniques to Detect Exact and Near-Exact Duplicate Images using Augmented Caltech-256 and ImageNet-Mini Datasets

	Hash-Based Techniques	**aHash**	**dHash**	**pHash**	**dHash**
Augmented Caltech-256-dataset	**Top-1 Accuracy**	13.9	10.1	23.9	21.8
	Top-1 Error	86.1	89.9	76.1	78.2
	Top-5 Accuracy	21.7	19.6	36.2	34.9
	Top-5 Error	78.3	80.4	63.8	65.1
Augmented ImageNet-Mini-dataset	**Top-1 Accuracy**	14.6	11.8	25.4	23.3
	Top-1 Error	85.4	88.2	74.6	76.7
	Top-5 Accuracy	23.4	21.8	38.3	37.1
	Top-5 Error	76.6	78.2	61.7	62.9

4.3.3.2 Storage Reduction of CNN-Based and Hash-Based Techniques

Identical or nearly identical duplicate images are detected using a fine-tuned CNN-based model and applying coarse and fine-grained searches using the augmented Caltech-256 dataset and the augmented ImageNet-Mini dataset. The number of duplicate images detected and storage reduction in megabytes on both datasets are examined and listed in Table 4.8. The augmented Caltech-256 dataset contains more augmented images, resulting in a higher number of duplicate images. VGGNET models achieve a greater storage reduction than AlexNet. VGGNET19 has a slightly greater image detection performance than VGGNET16. The same has been translated here in storage reduction due to its capacity to learn and represent more complex features than VGGnet16.

Table 4.8: CNN-based Exact and Near-Exact Duplicate Detection using Augmented Caltech-256 and ImageNet-Mini Datasets

	CNN-Based Model	**Fine-tuned AlexNet**	**Fine-tuned VGGNet-16**	**Fine-tuned VGGNet-19**
Augmented Caltech-256 dataset	# of Duplicate Images Detected	20204	22819	22987
	Storage reduced in MB	792	894	901
Augmented ImageNet-Mini dataset	# of Duplicate Images Detected	15091	16853	16924
	Storage reduced in MB	1617	1826	1844

The number of duplicate images detected and storage reduction using aHash, dHash, pHash, and wHash techniques are investigated. The wHash technique has detected more exact or near-exact images and reduced storage than aHash and dHash. Due to DCT-based hash generation and detection of images with changes such as cropping, scaling, and rotation, the pHash outperformed wHash in duplicate image detection and storage reductions. Table 4.9 shows the empirical results of the duplicate image detection and storage reduction on two datasets.

Table 4.9: Hash-based Exact and Near-Exact Duplicate Image Detection and Storage Reduction using Augmented Caltech-256 and ImageNet-Mini Datasets

	Hash-Based Model	**aHash**	**dHash**	**pHash**	**wHash**
Augmented Caltech-256 dataset	# of Duplicate Images Detected	4985	4498	8876	8470
	Storage reduced in MB	197	181	356	340
Augmented ImageNet-Mini dataset	# of Duplicate Images Detected	4249	3959	6955	6737
	Storage reduced in MB	477	444	780	756

The CNN-based techniques detect complex image features and have exhibited much better performance in terms of duplicate images detected and storage reduction than hash-based techniques. This storage reduction cuts the demand for additional storage disks and reduced the energy requirement of the storage system. The reduction of storage has a long-term effect on the energy saving of the storage system. In this chapter, the work analyzed the energy consumption of duplication techniques and its impact on the storage system.

4.3.3.3 Energy Consumption of CNN-Based and Hash-based Deduplication Techniques

The experimental work in this chapter evaluated the energy consumption during the execution training and detection of exact or near-exact duplicate images for both fine-tuned CNN models and hash-based techniques. These image duplicate detection techniques use additional CPUs, GPUs, memory, and disk reads. To measure the energy consumption, the total energy consumed by the deduplication process on a dataset is represented E_D as proposed in E_sDeDUP technique. Here $E_{D1_{\text{VggNet 16}}}, E_{D1_{\text{VggNet 19}}}, E_{D1_{\text{AlexNet}}}$ and $E_{D2_{\text{VggNet 16}}}, E_{D2_{\text{VggNet 19}}}, E_{D2_{\text{AlexNet}}}$ represents the average energy consumed by CNN-based deduplication techniques on Augmented Caltech-256 dataset and ImageNet-Mini dataset and are depicted in Figure 4.10. The energy consumption of deduplication techniques is evaluated and is measured in terms of kilowatts on various image scales. The scale of images varies from 10K images to 50K images in all the experiments. The energy consumed by VggNet19 is more on both datasets than VggNet16 and AlexNet, as VggNet19 has a deep neural network model and more memory units to store and process. The AlextNet uses the least energy on both datasets due to its relatively small structure.

The Caltech-256 has less number of categories, so the energy consumed and execution time is relatively low than the ImageNet-Mini dataset in all the CNN-based deduplication techniques.

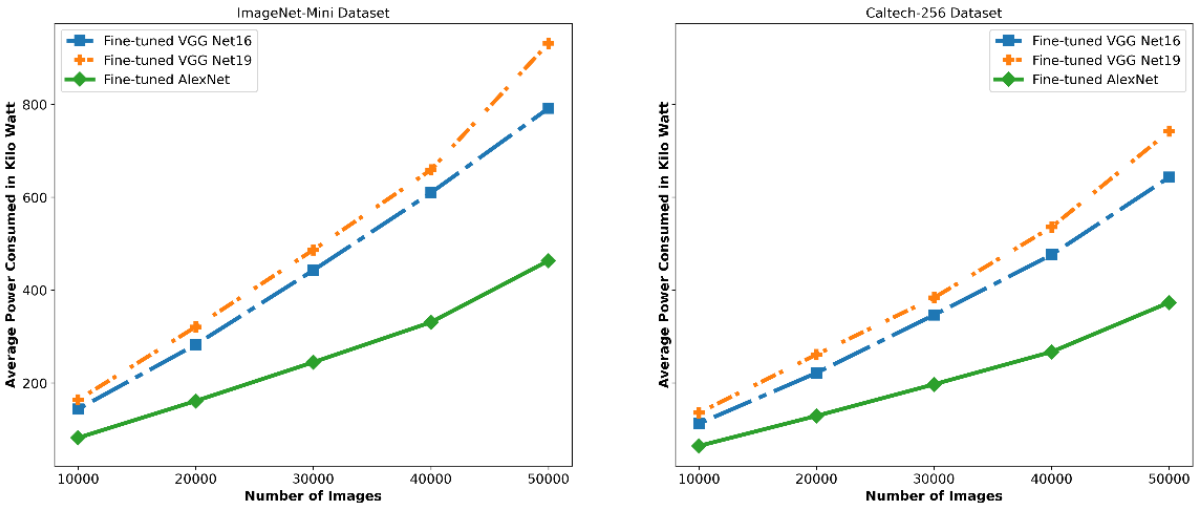


Figure 4.10: Average Energy Consumed in Kilo Watts by CNN-based Techniques to Detect Exact or Near-Exact Duplicate Images using ImageNet-Mini and Caltech-256 Datasets

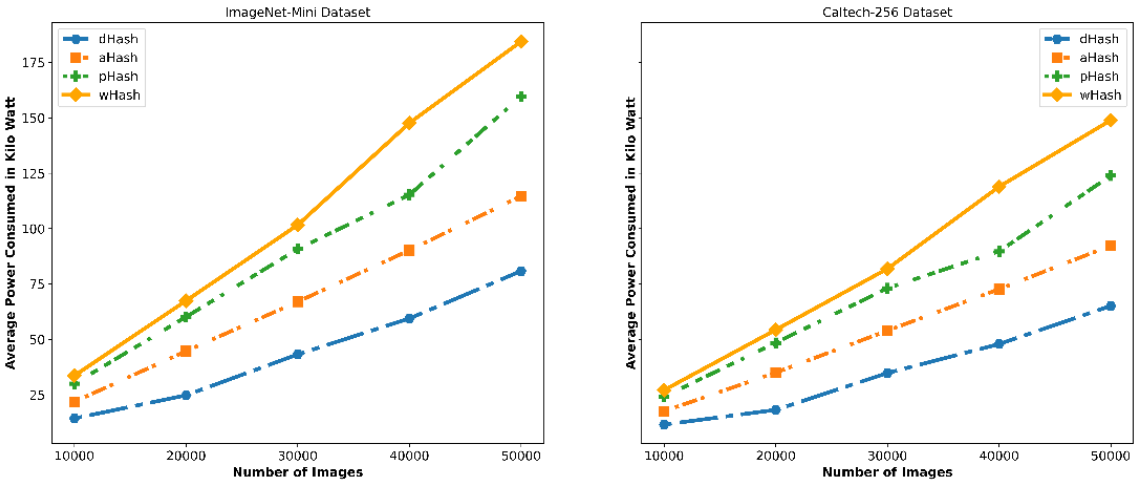


Figure 4.11: Average energy Consumed in Kilo Watts by Hash-based Techniques to Detect Exact or Near-Exact Duplicate Images using ImageNet-Mini and Caltech-256 Datasets

On the same terms, the average energy consumed by hash-based techniques represented by E_{D1_dHash}, E_{D1_aHash}, E_{D1_pHash}, E_{D1_pHash}, E_{D1_wHash} and E_{D2_dHash}, E_{D2_aHash}, E_{D2_pHash}, E_{D1_wHash} at different scale of input images are presented in Figure 4.11. The dHash exhibits the lowest energy consumption due to the low time and space complexity of dHash computation and matching on both datasets. The wavelet hash wHash consumed higher energy than pHash and aHash.

Each technique has its own time and space complexity, and its execution time varies which affects the energy consumption of the technique. To evaluate this, the research work in this chapter presented the execution time of CNN-based and hash-based deduplication techniques on different image scales on both the datasets in Figure 4.12.and Figure 4.13, respectively. The chapter also examined the average energy usage in watts per second during the execution of these techniques. The average energy consumed per second by CNN-based and hash-based deduplication techniques on various scales is studied using both datasets and presented in Figure 4.14. and Figure 4.15. respectively.

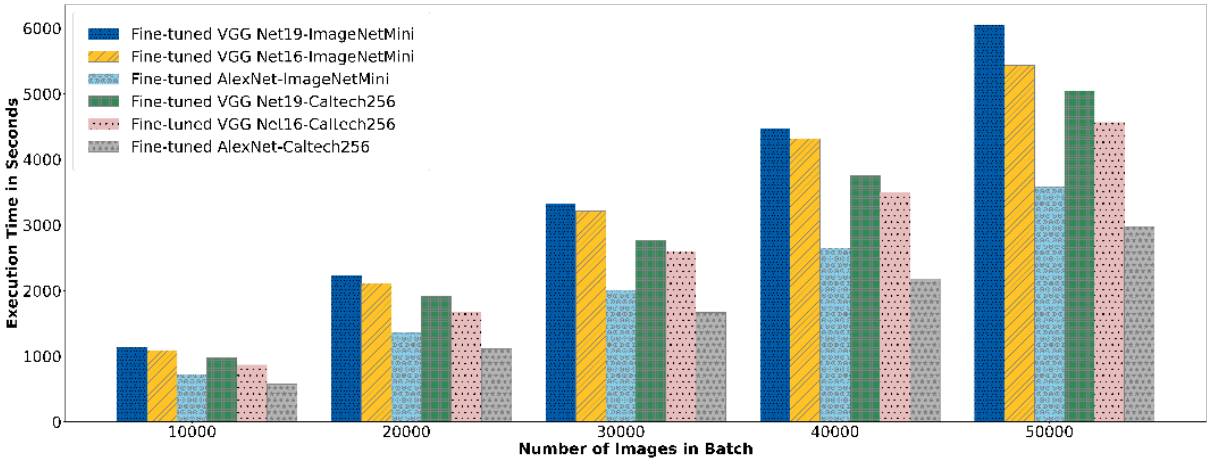


Figure 4.12: Execution Time of CNN-based Techniques to Detect Exact or Near-Exact Duplicate Images using ImageNet-Mini and Caltech-256 Datasets

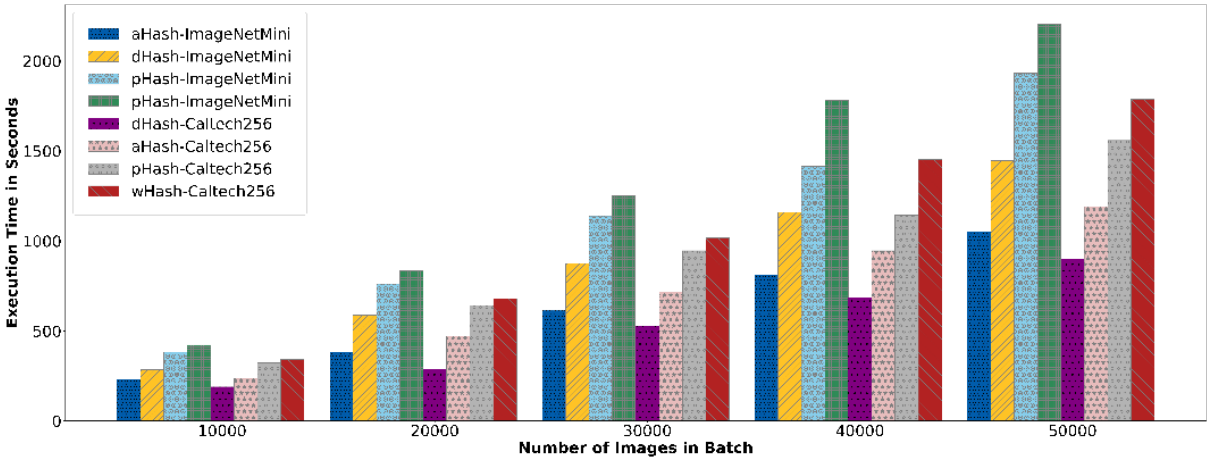


Figure 4.13: Execution Time of Hash-based Techniques to Detect Exact or Near-Exact Duplicate Images using ImageNet-Mini and Caltech-256 Datasets

Energy is mainly measured in CPU and I/Os because CPU and memory account for most energy consumption. Several parameters are used to calculate energy, such as CPU, I/Os, disc, network bandwidth, and so on, but in the proposed method, this study just looked at CPU computation as the CPU consumes maximum energy when read-write operations

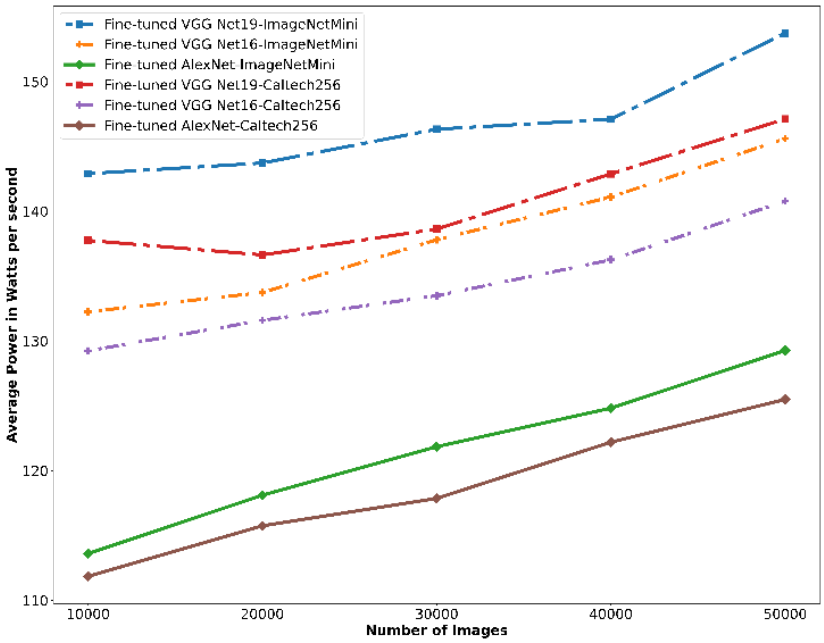


Figure 4.14: Average Energy in Watts per Second during Execution of CNN-based Deduplication Techniques at Different Scale of Images using ImageNet-Mini and Caltech-256 Datasets

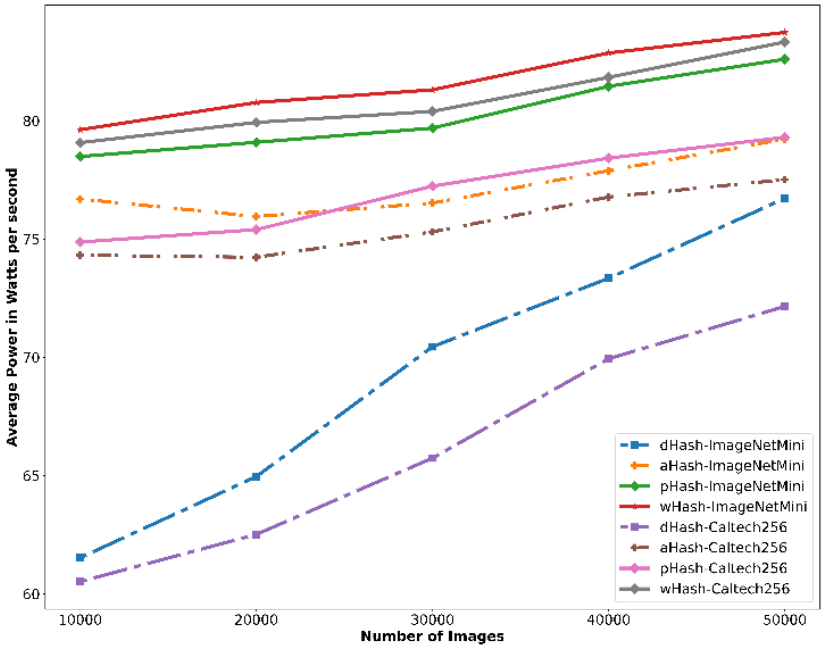


Figure 4.15: Average energy in Watts per Second during Execution of Hash-based Deduplication Techniques at Different Scale of Images using ImageNet-Mini and Caltech-256 Datasets

are executed.

4.4 Conclusion

This chapter proposes EsDeDUP a novel image deduplication technique for energy-saving for scalable exact or near-exact duplicate detection in cloud storage systems. The chapter proposes a model to analyze the energy consumption, energy savings, and performance analysis of scalable image deduplication techniques. The proposed technique characterizes the energy-consuming units and processing units of deduplication techniques. Further, the work evaluated the performance of phash, whash, ahash, and dhash-based duplication techniques and their effect on energy consumption and energy savings to detect an exact and near-exact duplicate image. Furthermore, this work utilizes fine-tuned CNN-based techniques, specifically employing fine-tuned models such as AlexNet, VGG-NET-16, and VGG-NET-19 to detect exact and near-exact duplicate images. The performance and energy savings of the CNN-based and hash-based deduplication techniques are evaluated using extensive experiments using two real image datasets. The result shows that the fine-tuned CNN-based deduplication techniques have shown better accuracy with higher energy consumption than hash-based deduplication techniques to detect exact and near-exact duplicate images in cloud storage systems.

The next chapter concludes the work, provides a summary, and highlights the research presented in this thesis, emphasizing its key contributions. In addition, it highlights potential future research directions.

Chapter 5

Conclusions and Future Directions

This chapter presents the research's conclusions and prospects for the future, thereby concluding the thesis. The work provides a comprehensive analysis of the contributions made by the thesis.

This thesis efficaciously addresses the need to design a framework for a cloud-based storage system using data deduplication techniques with a strong focus on storage parameters such as duplication detection capability, resource utilization, cost, and energy parameters. It provides an extensive review of data deduplication techniques in cloud storage systems focusing on different scenarios for exploring these techniques. The research carried out the implementation of an online image deduplication technique to detect exact or near-exact images using CNN and handcrafted features of images for cloud storage systems. Furthermore, the framework for the proposed technique has been designed and validated in this thesis. An energy-saving and performance analysis of image deduplication technique for scalable exact or near-exact image duplicate detection is presented using CNN and hash-based deduplication techniques. The performance of both techniques has been thoroughly evaluated and validated in this thesis.

Section 6.1 discusses the overall conclusions drawn from this work. In Section 6.2, the significant contributions of this work have been extensively discussed. In addition, the chapter suggests potential future research directions and extensions of this work, providing valuable insights for further exploration as outlined in Section 6.3.

5.1 Conclusions

This research presents data deduplication techniques for optimizing storage utilization in cloud storage systems. The techniques rely on identifying and eliminating duplicate data, improving cloud storage systems' overall storage and energy efficiency. The following section presents the main outcomes of this work:

The thesis commences with an introduction to data deduplication in cloud storage systems in Chapter 1. It offers a comprehensive overview of data deduplication, covering its evolution, necessity, functioning, and classification. The merits and demerits of data deduplication are also explored. The chapter briefly outlines the research motivation behind data deduplication and presents the primary contributions of this work. Finally, the chapter discusses the organization of the rest of the thesis.

Chapter 2 reviews the literature on various data deduplication techniques, traditional data compression approaches, and classification of data deduplication in cloud storage systems. In addition, several existing data deduplication techniques based on data types, i.e., text, image, and video-based deduplication techniques have been identified, analyzed, and classified along with their characteristics, enhancing the existing survey. Furthermore, based on existing research challenges, the objectives of the thesis have been sketched.

Chapter 3 proposes a novel data deduplication framework following a comprehensive analysis of existing data deduplication techniques and their limitations, as determined by Chapter 2. A deep CNN-based online image deduplication technique for a cloud storage system has been proposed and the key objectives and requirements of the suggested framework have been analyzed in-depth. Initially, the system uses a CNN-based online image deduplication algorithm to identify exact or near-exact images. Later, the Hot Decomposition Vector (HDV) was proposed to enhance the accuracy and efficiency of image matching for nearly identical images. Also, HDV generates image patches to store dissimilar parts of near-exact images. A fine-tuned AlexNet is proposed for normal and cross-domain online image deduplication. Moreover, multi-classifier decision fusion is proposed for exact or near-exact image detection.
The experimental results have been implemented in this chapter by categorizing them into three sections. The first section describes the performance of HDV with that of traditional image feature extraction techniques and their combinations. The results demonstrate that HDV achieves greater and more stable image-matching accuracy for all three

forms of image deformation with relatively least computational time. Fine-tuned AlexNet is implemented, and the outcomes of this proposed network are compared with existing CNN-based AlexNet and VGGNet techniques to detect exact or near-exact images. The key performance indicators are compared to existing CNN approaches, and the findings are provided. The proposed fine-tuned CNN-based feature extraction technique for online deduplication provides better accuracy when compared to AlexNet and VGGNet. Various image classifiers are evaluated based on different image variations to detect exact or near-exact images, and various findings are discussed. The Bayesian image classifier outperforms the other image classifiers in terms of illumination, pose image variations and exact images.

In **Chapter 4**, EsDeDUP, a novel image deduplication technique for energy-saving has been proposed to compute the energy-saving effect for deduplication. The technique detects duplicate data, manages storage systems to resolve duplicate data issues, and provides a cost-effective solution. The main objective of this chapter is to detect duplicate images and determine energy usage in terms of CPU computation and performance analysis of scalable image deduplication techniques. The chapter evaluated the performance of hash-based namely, pHash,wHash, aHash, and dHash-based and CNN-based fine-tuned AlexNet, fine-tuned VGG-NET-16, and fine-tuned VGG-NET-19 image deduplication techniques and their effect on power consumption to detect exact or near-exact images. The proposed technique analyzes the impact of image deduplication techniques on energy-saving, accuracy, and storage reduction using real datasets. It proposes a power-saving computation model for deduplication techniques on a common dataset and a workload. The extensive experiments of hash-based and fine-tuned CNN-Based deduplication techniques for performance evaluations and power-saving were achieved using two real image datasets. The result shows that the fine-tuned CNN-based deduplication techniques have shown better accuracy with higher power consumption than hash-based deduplication techniques to detect exact or near-exact duplicate images.

5.2 Thesis Contributions

This work attempts to enhance the efficiency of cloud storage by incorporating data deduplication techniques into this realm. This effort aims to investigate the areas of interest examined by previous researchers and identify existing gaps and unexplored areas in data deduplication for cloud storage systems, particularly focusing on online deduplication techniques. The primary contributions of this thesis are outlined below:

- An exhaustive survey of the work carried out in data deduplication in cloud storage systems has been performed. This work aims to gain insight into various data deduplication strategies applicable to diverse data types to develop an effective model.
- Proposed an online CNN-based data deduplication technique for detecting exact or near-exact duplicate images in cloud storage systems.
- Proposed Hot Decomposition Vector (HDV) based on duplicate image detection for near-exact image patch generation.
- Proposed an energy-efficient CNN-based deduplication technique that detects exact or near-exact image duplicates and conserves energy by employing a CPU computation parameter and calculating the accuracy of the suggested technique.
- The design, development, and implementation of the proposed techniques and their applicability in cloud computing are described in the thesis.

5.3 Future Directions

The contribution to this thesis has spawned new research areas in cloud storage systems that are required to be addressed through further research. Consequently, this section presents some suggestions for potential implementation to extend this work in the future:

- The energy-efficient data deduplication approach has the potential to incorporate additional energy parameters, thereby reducing energy consumption costs and heat emissions in the cloud storage system. Consequently, this would result in a reduction of the cloud storage system's overall power consumption expenses.
- The proposed online image deduplication technique could be further developed by integrating security parameters. These additions would safeguard data against potential side-channel attacks and ensure privacy and security regulations compliance.
- There is potential for the scope of this work to broaden, encompassing additional enhancements geared towards reducing computation time. This could involve areas such as duplicate detection, hash calculations, and more, achieved through the implementation of parallel computations.
- Data deduplication techniques can be extended to other domains, such as employing advanced machine learning methods, to enhance performance and accuracy without incurring additional computational costs.

- Less attention has been paid to primary, RAM, and SSD deduplication systems, and additional contributions for enhancing deduplication throughput, decreasing I/O latency, and increasing deduplication space savings are still anticipated.
- In the context of future research, it is possible to improve various aspects, including reference management, scalability, dependability, and security, even in the domain of backup deduplication, which involves a substantial workload.
- To optimize the deduplication performance, image patches can be utilized for reconstructing images. This can be achieved by employing a data structure, such as a B+ tree, to store nearly identical images efficiently. It has the potential to decrease storage capacity and cost further.
- Another area that necessitates additional focus is utilizing B+ tree data for image reconstruction. The current experiments were limited to single-step crops or rotations. Future research will focus on scenarios involving multiple operations on the same image.
- Optimizing deduplication techniques for big data storage in real-world scenarios can be a complex task due to various other challenges and intricacies. Some of the key challenges are complex distributed storage systems, Data Diversity, Data Security, Regulatory Compliance, Duplicate Data Retention Policies, etc. These challenges are evolving research problems that need scientific solutions.

References

[1] Carlos Alvarez. Netapp deduplication for fas and v-series deployment and implementation guide. *Technical ReportTR-3505*, 2011.

[2] Min Gu, Xiangping Li, and Yaoyu Cao. Optical storage arrays: a perspective for future big data storage. *Light: Science & Applications*, 3(5):e177–e177, 2014.

[3] Federica Lucivero. Big data, big waste? a reflection on the environmental sustainability of big data initiatives. *Science and engineering ethics*, 26(2):1009–1030, 2020.

[4] David Reinsel-John Gantz-John Rydning, John Reinsel, and John Gantz. The digitization of the world from edge to core. *Framingham: International Data Corporation*, 16:1–28, 2018.

[5] David Reinsel, John Gantz, and John Rydning. Data age 2025: the evolution of data to life-critical don't focus on big data; focus on the data that's big. *International Data Corporation (IDC) White Paper*, 2017.

[6] Anil Singh and Nitin Auluck. Load balancing aware scheduling algorithms for fog networks. *Software: Practice and Experience*, 50(11):2012–2030, 2020.

[7] Waraporn Leesakul, Paul Townend, and Jie Xu. Dynamic data deduplication in cloud storage. In *2014 IEEE 8th International Symposium on Service Oriented System Engineering*, pages 320–325. IEEE, 2014.

[8] Dirk Meister, Jurgen Kaiser, Andre Brinkmann, Toni Cortes, Michael Kuhn, and Julian Kunkel. A study on data deduplication in hpc storage systems. In *SC'12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–11. IEEE, 2012.

[9] Deepavali Bhagwat, Kave Eshghi, Darrell DE Long, and Mark Lillibridge. Extreme binning: Scalable, parallel deduplication for chunk-based file backup. In *2009 IEEE International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems*, pages 1–9. IEEE, 2009.

[10] Ho Min Jung, Won Vien Park, Wan Yeon Lee, Jeong Gun Lee, and Young Woong Ko. Data deduplication system for supporting multi-mode. In *Asian Conference on Intelligent Information and Database Systems*, pages 78–87. Springer, 2011.

[11] A Lavanya and P Shanmugam. Data de-duplication using frequency based chunking on virtual storage.

[12] João Paulo and José Pereira. A survey and classification of storage deduplication systems. *ACM Computing Surveys (CSUR)*, 47(1):1–30, 2014.

[13] Bo Mao, Hong Jiang, Suzhen Wu, Yinjin Fu, and Lei Tian. Read-performance optimization for deduplication-based storage systems in the cloud. *ACM Transactions on Storage (TOS)*, 10(2):1–22, 2014.

[14] Dutch T Meyer and William J Bolosky. A study of practical deduplication. *ACM Transactions on Storage (ToS)*, 7(4):1–20, 2012.

[15] Xun Zhao, Yang Zhang, Yongwei Wu, Kang Chen, Jinlei Jiang, and Keqin Li. Liquid: A scalable deduplication file system for virtual machine images. *IEEE transactions on parallel and distributed systems*, 25(5):1257–1266, 2013.

[16] Eduardo N Borges, Moisés G de Carvalho, Renata Galante, Marcos André Gonçalves, and Alberto HF Laender. An unsupervised heuristic-based approach for bibliographic metadata deduplication. *Information Processing & Management*, 47(5):706–718, 2011.

[17] Wen Xia, Hong Jiang, Dan Feng, Fred Douglis, Philip Shilane, Yu Hua, Min Fu, Yucheng Zhang, and Yukun Zhou. A comprehensive study of the past, present, and future of data deduplication. *Proceedings of the IEEE*, 104(9):1681–1710, 2016.

[18] Anmol Jyot Maan. Analysis and comparison of algorithms for lossless data compression. *International Journal of Information and Computation Technology*, 3(3):139–146, 2013.

[19] Wen Xia, Hong Jiang, Dan Feng, Lei Tian, Min Fu, and Yukun Zhou. Ddelta: A deduplication-inspired fast delta compression approach. *Performance Evaluation*, 79:258–272, 2014.

[20] Huijun Wu, Chen Wang, Kai Lu, Yinjin Fu, and Liming Zhu. One size does not fit all: The case for chunking configuration in backup deduplication. In *2018 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, pages 213–222. IEEE, 2018.

[21] Ali Shakarami, Mostafa Ghobaei-Arani, Ali Shahidinejad, Mohammad Masdari, and Hamid Shakarami. Data replication schemes in cloud computing: a survey. *Cluster Computing*, 24(3):2545–2579, 2021.

[22] Anand Bhalerao and Ambika Pawar. A survey: On data deduplication for efficiently utilizing cloud storage for big data backups. In *2017 international conference on trends in electronics and informatics (ICEI)*, pages 933–938. IEEE, 2017.

[23] Nagapramod Mandagere, Pin Zhou, Mark A Smith, and Sandeep Uttamchandani. Demystifying data deduplication. In *Proceedings of the ACM/IFIP/USENIX Middleware'08 Conference Companion*, pages 12–17, 2008.

[24] João Paulo and José Pereira. Distributed exact deduplication for primary storage infrastructures. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 52–66. Springer, 2014.

[25] A Faritha Banu and C Chandrasekar. A survey on deduplication methods. *International Journal of Computer Trends and Technology*, 3(3):364–368, 2012.

[26] Qinlu He, Zhanhuai Li, and Xiao Zhang. Data deduplication techniques. In *2010 international conference on future information technology and management engineering*, volume 1, pages 430–433. IEEE, 2010.

[27] Ruijin Zhou, Ming Liu, and Tao Li. Characterizing the efficiency of data deduplication for big data storage management. In *2013 IEEE international symposium on workload characterization (IISWC)*, pages 98–108. IEEE, 2013.

[28] Raja Wasim Ahmad, Abdullah Gani, Muhammad Shiraz, Feng Xia, Sajjad A Madani, et al. Virtual machine migration in cloud data centers: a review, taxonomy, and open research issues. *The Journal of Supercomputing*, 71(7):2473–2515, 2015.

[29] Yang Hu, Chao Li, Longjun Liu, and Tao Li. Hope: enabling efficient service orchestration in software-defined data centers. In *Proceedings of the 2016 International Conference on Supercomputing*, pages 1–12, 2016.

[30] D Viji and S Revathy. Various data deduplication techniques of primary storage. In *2019 International conference on communication and electronics systems (ICCES)*, pages 322–327. IEEE, 2019.

[31] Lifang Lin, Yuhui Deng, Yi Zhou, and Yifeng Zhu. Inde: An inline data deduplication approach via adaptive detection of valid container utilization. *ACM Transactions on Storage*, 2022.

[32] Myoungwon Oh, Sejin Park, Jungyeon Yoon, Sangjae Kim, Kang-won Lee, Sage Weil, Heon Y Yeom, and Myoungsoo Jung. Design of global data deduplication for a scale-out distributed storage system. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1063–1073. IEEE, 2018.

[33] Datong Zhang, Yuhui Deng, Yi Zhou, Yifeng Zhu, and Xiao Qin. Improving the performance of deduplication-based backup systems via container utilization based hot fingerprint entry distilling. *ACM Transactions on Storage (TOS)*, 17(4):1–23, 2021.

[34] Xu Chu, Ihab F Ilyas, and Paraschos Koutris. Distributed data deduplication. *Proceedings of the VLDB Endowment*, 9(11):864–875, 2016.

[35] Lei Si, Shujie Pang, Yuhui Deng, Weiheng Zhu, Yi Zhou, and Yifeng Zhu. Dm-pages: Improving energy efficiency of disk storage systems and cache performance using deduplication-based mixed pages. *Journal of Circuits, Systems and Computers*, 31(16):2250275, 2022.

[36] Yizhou Yan and Wenjun Wu. Analysis of energy consumption of deduplication in storage systems. In *2015 International Conference on Cyber-Enabled Distributed*

Computing and Knowledge Discovery, pages 295–301. IEEE, 2015.

[37] Jing Li, Xueming Qian, Qing Li, Yisi Zhao, Liejun Wang, and Yuan Yan Tang. Mining near duplicate image groups. *Multimedia Tools and Applications*, 74(2):655–669, 2015.

[38] G Thippa Reddy, M Praveen Kumar Reddy, Kuruva Lakshmanna, Rajesh Kaluri, Dharmendra Singh Rajput, Gautam Srivastava, and Thar Baker. Analysis of dimensionality reduction techniques on big data. *Ieee Access*, 8:54776–54788, 2020.

[39] Richa Siddavaatam, Isaac Woungang, Glaucio HS Carvalho, and Alagan Anpalagan. Mobile cloud storage over 5g: A mechanism design approach. *IEEE Systems Journal*, 13(4):4060–4071, 2019.

[40] John Gantz and David Reinsel. The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. *IDC iView: IDC Analyze the future*, 2007(2012):1–16, 2012.

[41] Joao Barreto and Paulo Ferreira. Efficient locally trackable deduplication in replicated systems. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 103–122. Springer, 2009.

[42] Senthil Shanmugasundaram and Robert Lourdusamy. A comparative study of text compression algorithms. *International Journal of Wisdom Based Computing*, 1(3):68–76, 2011.

[43] Umesh S Bhadade and AI Trivedi. Lossless text compression using dictionaries. *International Journal of Computer Applications*, 13(8):27–34, 2011.

[44] EH Sibley, IANH Willen, RM Neal, and JG Cleary. Arithmetic coding for data compression. *ACM Trans. on Comm*, 30(6):10–1145, 1987.

[45] Hermine Hovhannisyan, Wen Qi, Kejie Lu, Rongwei Yang, and Jianping Wang. Whispers in the cloud storage: A novel cross-user deduplication-based covert channel design. *Peer-to-Peer Networking and Applications*, 11(2):277–286, 2018.

[46] CL Philip Chen and Chun-Yang Zhang. Data-intensive applications, challenges, techniques and technologies: A survey on big data. *Information sciences*, 275:314–347, 2014.

[47] Nadiah bt Yusof, Amirah Ismail, and Nazatul Aini Abd Majid. Deduplication image middleware detection comparison in standalone cloud database.

[48] Zhenhua Nie, Yu Hua, Dan Feng, Qiuyu Li, and Yuanyuan Sun. Efficient storage support for real-time near-duplicate video retrieval. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 312–324. Springer, 2014.

[49] Mohamed Abouelenien, Xiaohui Yuan, Balathasan Giritharan, Jianguo Liu, and Shoujiang Tang. Cluster-based sampling and ensemble for bleeding detection in

capsule endoscopy videos. *American Journal of Science and Engineering*, 2(1):24–32, 2013.

[50] Ming Chen, Shupeng Wang, and Liang Tian. A high-precision duplicate image deduplication approach. *J. Comput.*, 8(11):2768–2775, 2013.

[51] GuiPing Wang, ShuYu Chen, MingWei Lin, and XiaoWei Liu. Sbbs: A sliding blocking algorithm with backtracking sub-blocks for duplicate data detection. *Expert systems with applications*, 41(5):2415–2423, 2014.

[52] Michael O Rabin. Fingerprinting by random polynomials. *Technical report*, 1981.

[53] Yinjin Fu, Hong Jiang, Nong Xiao, Lei Tian, Fang Liu, and Lei Xu. Application-aware local-global source deduplication for cloud backup services of personal storage. *IEEE transactions on parallel and distributed systems*, 25(5):1155–1165, 2013.

[54] Lifang Lin, Yuhui Deng, Yi Zhou, and Yifeng Zhu. Inde: An inline data deduplication approach via adaptive detection of valid container utilization. *ACM Transactions on Storage*, 19(1):1–27, 2023.

[55] S Venkatesh Babu, P Ramya, Jeffin Gracewell, et al. Content deduplication with granularity tweak based on base and deviation for large text dataset. *Scientific Programming*, 2022, 2022.

[56] Xuming Ye, Jia Tang, Wenlong Tian, Ruixuan Li, Weijun Xiao, Yuqing Geng, and Zhiyong Xu. Fast variable-grained resemblance data deduplication for cloud storage. In *2021 IEEE International Conference on Networking, Architecture and Storage (NAS)*, pages 1–8. IEEE, 2021.

[57] Michael Hirsch, Shmuel T Klein, Dana Shapira, and Yair Toaff. Dynamic determination of variable sizes of chunks in a deduplication system. *Discrete Applied Mathematics*, 274:81–91, 2020.

[58] Zhichao Cao, Hao Wen, Xiongzi Ge, Jingwei Ma, Jim Diehl, and David HC Du. Tddfs: A tier-aware data deduplication-based file system. *ACM Transactions on Storage (TOS)*, 15(1):1–26, 2019.

[59] Yongtao Zhou, Yuhui Deng, Laurence T Yang, Ru Yang, and Lei Si. Ldfs: A low latency in-line data deduplication file system. *IEEE access*, 6:15743–15753, 2018.

[60] Dongchul Park, Ziqi Fan, Young Jin Nam, and David HC Du. A lookahead read cache: improving read performance for deduplication backup storage. *Journal of Computer Science and Technology*, 32(1):26–40, 2017.

[61] A Venish and K Siva Sankar. Study of chunking algorithm in data deduplication. In *Proceedings of the International Conference on Soft Computing Systems*, pages 13–20. Springer, 2016.

[62] Wen Xia, Hong Jiang, Dan Feng, and Lei Tian. Dare: A deduplication-aware resemblance detection and elimination scheme for data reduction with low overheads.

IEEE Transactions on Computers, 65(6):1692–1705, 2015.

[63] Guann-Ling Shen and Che-Rung Lee. Flomd: Fast and low overhead memory deduplication for edge nodes. In *2022 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 83–90. IEEE, 2022.

[64] Zihao Zhang, Huiqi Hu, Zhihui Xue, Changcheng Chen, Yang Yu, Cuiyun Fu, Xuan Zhou, and Feifei Li. Slimstore: A cloud-based deduplication system for multi-version backups. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1841–1846. IEEE, 2021.

[65] Yujuan Tan, Congcong Xu, Jing Xie, Zhichao Yan, Hong Jiang, Witawas Srisa-an, Xianzhang Chen, and Duo Liu. Improving the performance of deduplication-based storage cache via content-driven cache management methods. *IEEE Transactions on Parallel and Distributed Systems*, 32(1):214–228, 2020.

[66] Guangping Xu, Bo Tang, Hongli Lu, Quan Yu, and Chi Wan Sung. Lipa: A learning-based indexing and prefetching approach for data deduplication. In *2019 35th Symposium on mass storage systems and technologies (MSST)*, pages 299–310. IEEE, 2019.

[67] Huijun Wu, Chen Wang, Yinjin Fu, Sherif Sakr, Kai Lu, and Liming Zhu. A differentiated caching mechanism to enable primary storage deduplication in clouds. *IEEE Transactions on Parallel and Distributed Systems*, 29(6):1202–1216, 2018.

[68] Yinjin Fu, Nong Xiao, Hong Jiang, Guyu Hu, and Weiwei Chen. Application-aware big data deduplication in cloud environment. *IEEE transactions on cloud computing*, 7(4):921–934, 2017.

[69] Suzhen Wu, Xiao Chen, and Bo Mao. Exploiting the data redundancy locality to improve the performance of deduplication-based storage systems. In *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*, pages 527–534. IEEE, 2016.

[70] Saja Taha Ahmed and Loay E George. Lightweight hash-based de-duplication system using the self detection of most repeated patterns as chunks divisors. *Journal of King Saud University-Computer and Information Sciences*, 34(7):4669–4678, 2022.

[71] Ahmed Sardar M Saeed and Loay E George. Fingerprint-based data deduplication using a mathematical bounded linear hash function. *Symmetry*, 13(11):1978, 2021.

[72] S Hema and A Kangaiammal. Distributed storage hash algorithm (dsha) for file-based deduplication in cloud computing. In *Second International Conference on Computer Networks and Communication Technologies: ICCNCT 2019*, pages 572–581. Springer, 2020.

[73] Wen Xia, Dan Feng, Hong Jiang, Yucheng Zhang, Victor Chang, and Xiangyu Zou. Accelerating content-defined-chunking based data deduplication by exploiting

parallelism. *Future Generation Computer Systems*, 98:406–418, 2019.

[74] Hala AbdulSalam Jasim and Assmaa A Fahad. New techniques to enhance data deduplication using content based-tttd chunking algorithm. *International Journal of Advanced Computer Science and Applications*, 9(5), 2018.

[75] Panfeng Zhang, Ping Huang, Xubin He, Hua Wang, and Ke Zhou. Resemblance and mergence based indexing for high performance data deduplication. *Journal of Systems and Software*, 128:11–24, 2017.

[76] Ryan NS Widodo, Hyotaek Lim, and Mohammed Atiquzzaman. A new content-defined chunking algorithm for data deduplication in cloud storage. *Future Generation Computer Systems*, 71:145–156, 2017.

[77] Naresh Kumar, Rahul Rawat, and SC Jain. Bucket based data deduplication technique for big data storage system. In *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*, pages 267–271. IEEE, 2016.

[78] Mingyang Song, Zhongyun Hua, Yifeng Zheng, Hejiao Huang, and Xiaohua Jia. Blockchain-based deduplication and integrity auditing over encrypted cloud storage. *IEEE Transactions on Dependable and Secure Computing*, (01):1–18, 2023.

[79] Shuqin Liu, Yusong Yao, Guohua Tian, Jianghong Wei, and Xialin Liu. A blockchain-based compact audit-enabled deduplication in decentralized storage. *Computer Standards & Interfaces*, 85:103718, 2023.

[80] Qingyuan Xie, Chen Zhang, and Xiaohua Jia. Security-aware and efficient data deduplication for edge-assisted cloud storage systems. *IEEE Transactions on Services Computing*, 2022.

[81] Yunlong He, Hequn Xian, Liming Wang, and Shuguang Zhang. Secure encrypted data deduplication based on data popularity. *Mobile networks and applications*, 26:1686–1695, 2021.

[82] Nourah Almrezeq et al. An enhanced approach to improve the security and performance for deduplication. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 12(6):2866–2882, 2021.

[83] Guohua Tian, Hua Ma, Ying Xie, and Zhenhua Liu. Randomized deduplication with ownership management and data sharing in cloud storage. *Journal of Information Security and Applications*, 51:102432, 2020.

[84] Liang Wang, Baocang Wang, Wei Song, and Zhili Zhang. A key-sharing based secure deduplication scheme in cloud storage. *Information Sciences*, 504:48–60, 2019.

[85] Jinbo Xiong, Yuanyuan Zhang, Xuan Li, Mingwei Lin, Zhiqiang Yao, and Guangjun Liu. Rse-pow: A role symmetric encryption pow scheme with authorized dedupli-

cation for multimedia data. *Mobile Networks and Applications*, 23:650–663, 2018.

[86] Wenhai Sun, Ning Zhang, Wenjing Lou, and Y Thomas Hou. Tapping the potential: Secure chunk-based deduplication of encrypted data for cloud backup. In *2018 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE, 2018.

[87] NS Vishalakshi and S Sridevi. Survey on secure de-duplication with encrypted data for cloud storage. *Int J Adv Res Sci Eng Technol*, 4(1):3111–3117, 2017.

[88] Yuhua Wang, Xin Tang, Yiteng Zhou, Xiguang Chen, and Yudan Zhu. Blockchain-based integrity auditing with secure deduplication in cloud storage. In *Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II*, pages 303–318. Springer, 2023.

[89] Xixun Yu, Hui Bai, Zheng Yan, and Rui Zhang. Veridedup: A verifiable cloud data deduplication scheme with integrity and duplication proof. *IEEE Transactions on Dependable and Secure Computing*, 20(1):680–694, 2022.

[90] Xuewei Ma, Wenyuan Yang, Yuesheng Zhu, and Zhiqiang Bai. A secure and efficient data deduplication scheme with dynamic ownership management in cloud computing. In *2022 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pages 194–201. IEEE, 2022.

[91] Ge Kan, Chunhua Jin, Huihui Zhu, Yongliang Xu, and Nian Liu. An identity-based proxy re-encryption for data deduplication in cloud. *Journal of Systems Architecture*, 121:102332, 2021.

[92] Xiaoyu Zheng, Yuyang Zhou, Yalan Ye, and Fagen Li. A cloud data deduplication scheme based on certificateless proxy re-encryption. *Journal of Systems Architecture*, 102:101666, 2020.

[93] Shynu PG, Nadesh RK, Varun G Menon, Mahdi Abbasi, Mohammad R Khosravi, et al. A secure data deduplication system for integrated cloud-edge networks. *Journal of Cloud Computing*, 9(1):1–12, 2020.

[94] Haoran Yuan, Xiaofeng Chen, Jin Li, Tao Jiang, Jianfeng Wang, and Robert H Deng. Secure cloud data deduplication with efficient re-encryption. *IEEE Transactions on Services Computing*, 15(1):442–456, 2019.

[95] Haoran Yuan, Xiaofeng Chen, Tao Jiang, Xiaoyu Zhang, Zheng Yan, and Yang Xiang. Dedupdum: Secure and scalable data deduplication with dynamic user management. *Information Sciences*, 456:159–173, 2018.

[96] Youngjoo Shin, Dongyoung Koo, and Junbeom Hur. A survey of secure data deduplication schemes for cloud storage systems. *ACM computing surveys (CSUR)*, 49(4):1–38, 2017.

[97] MS Pokale, Surabhi Dhok, Vaishnavi Kasbe, Gauri Joshi, and Noopur Shinde.

Data deduplication and load balancing techniques on cloud systems. *Int J Adv Res Comput Commun Eng*, 6(3):878–883, 2017.

[98] P Neelaveni and Muthuswamy Vijayalakshmi. Fc-lid: file classifier based linear indexing for deduplication in cloud backup services. In *International Conference on Distributed Computing and Internet Technology*, pages 213–222. Springer, 2016.

[99] Benjamin Zhu, Kai Li, and R Hugo Patterson. Avoiding the disk bottleneck in the data domain deduplication file system. In *Fast*, volume 8, pages 269–282, 2008.

[100] Biplob Debnath, Sudipta Sengupta, and Jin Li. {ChunkStash}: Speeding up inline storage deduplication using flash memory. In *2010 USENIX Annual Technical Conference (USENIX ATC 10)*, 2010.

[101] Mark Lillibridge, Kave Eshghi, Deepavali Bhagwat, Vinay Deolalikar, Greg Trezis, and Peter Camble. Sparse indexing: Large scale, inline deduplication using sampling and locality. In *Fast*, volume 9, pages 111–123, 2009.

[102] Fanglu Guo and Petros Efstathopoulos. Building a high-performance deduplication system. In *2011 USENIX Annual Technical Conference (USENIX ATC 11)*, 2011.

[103] Wen Xia, Hong Jiang, Dan Feng, and Yu Hua. Similarity and locality based indexing for high performance data deduplication. *IEEE transactions on computers*, 64(4):1162–1176, 2014.

[104] Yinjin Fu, Hong Jiang, and Nong Xiao. A scalable inline cluster deduplication framework for big data protection. In *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, pages 354–373. Springer, 2012.

[105] Wei Dong, Fred Douglis, Kai Li, Hugo Patterson, Sazzala Reddy, and Philip Shilane. Tradeoffs in scalable data routing for deduplication clusters. In *9th USENIX Conference on File and Storage Technologies (FAST 11)*, 2011.

[106] Tian-Ming Yang, Dan Feng, Zhong-ying Niu, and Ya-ping Wan. Scalable high performance de-duplication backup via hash join. *Journal of Zhejiang University SCIENCE C*, 11(5):315–327, 2010.

[107] Kiran Srinivasan, Timothy Bisson, Garth R Goodson, and Kaladhar Voruganti. idedup: latency-aware, inline data deduplication for primary storage. In *Fast*, volume 12, pages 1–14, 2012.

[108] Michal Kaczmarczyk, Marcin Barczynski, Wojciech Kilian, and Cezary Dubnicki. Reducing impact of data fragmentation caused by in-line deduplication. In *Proceedings of the 5th Annual International Systems and Storage Conference*, pages 1–12, 2012.

[109] Sadiq H Abdulhussain, Basheera M Mahmmod, M Iqbal Saripan, SAR Al-Haddad, Thar Baker, Wameedh N Flayyih, Wissam A Jassim, et al. A fast feature extraction

algorithm for image and video processing. In *2019 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2019.

[110] N Pattabhi Ramaiah and C Krishna Mohan. De-duplication of photograph images using histogram refinement. In *2011 IEEE Recent Advances in Intelligent Computational Systems*, pages 391–395. IEEE, 2011.

[111] Simranjot Kaur, Rajneesh Rani, Ritu Garg, and Nonita Sharma. State-of-the-art techniques for passive image forgery detection: a brief review. *International Journal of Electronic Security and Digital Forensics*, 14(5):456–473, 2022.

[112] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[113] Yanwei Pang, Wei Li, Yuan Yuan, and Jing Pan. Fully affine invariant surf for image matching. *Neurocomputing*, 85:6–10, 2012.

[114] Luo Juan and Oubong Gwun. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009.

[115] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pages II–II. IEEE, 2004.

[116] Qiaoliang Li, Guoyou Wang, Jianguo Liu, and Shaobo Chen. Robust scale-invariant feature matching for remote sensing image registration. *IEEE Geoscience and Remote Sensing Letters*, 6(2):287–291, 2009.

[117] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International conference on computer vision*, pages 2548–2555. Ieee, 2011.

[118] Yenewondim Biadgie and Kyung-Ah Sohn. Feature detector using adaptive accelerated segment test. In *2014 International Conference on Information Science & Applications (ICISA)*, pages 1–4. IEEE, 2014.

[119] Ming Chen, Yang Wang, Xiaoxiang Zou, Shupeng Wang, and Guangjun Wu. A duplicate image deduplication approach via haar wavelet technology. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, volume 2, pages 624–628. IEEE, 2012.

[120] Fei Zuo and Peter HN de With. Fast facial feature extraction using a deformable shape model with haar-wavelet based local texture attributes. In *2004 International Conference on Image Processing, 2004. ICIP'04.*, volume 3, pages 1425–1428. IEEE, 2004.

[121] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[122] Jing Zhang, Zhanlei Feng, and Yuting Su. A new approach for detecting copy-move forgery in digital images. In *2008 11th IEEE Singapore International Conference on Communication Systems*, pages 362–366, 2008.

[123] Kadir A Peker. Binary sift: Fast image retrieval using binary quantized sift features. In *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 217–222. IEEE, 2011.

[124] Sugata Banerji, Atreyee Sinha, and Chengjun Liu. New image descriptors based on color, texture, shape, and wavelets for object and scene image classification. *Neurocomputing*, 117:173–185, 2013.

[125] Fudong Nian, Teng Li, Xinyu Wu, Qingwei Gao, and Feifeng Li. Efficient near-duplicate image detection with a local-based binary representation. *Multimedia Tools and Applications*, 75(5):2435–2452, 2016.

[126] Li Liu, Yue Lu, and Ching Y Suen. Variable-length signature for near-duplicate image matching. *IEEE Transactions on Image Processing*, 24(4):1282–1296, 2015.

[127] Jian-Gang Wang, Jun Li, Chong Yee Lee, and Wei-Yun Yau. Dense sift and gabor descriptors-based face representation with applications to gender recognition. In *2010 11th International Conference on Control Automation Robotics & Vision*, pages 1860–1864. IEEE, 2010.

[128] Yan Ke, Rahul Sukthankar, Larry Huston, Yan Ke, and Rahul Sukthankar. Efficient near-duplicate detection and sub-image retrieval. In *ACM multimedia*, volume 4, page 5. Citeseer, 2004.

[129] Wan-Lei Zhao, Chong-Wah Ngo, Hung-Khoon Tan, and Xiao Wu. Near-duplicate keyframe identification with interest point matching and pattern learning. *IEEE Transactions on Multimedia*, 9(5):1037–1048, 2007.

[130] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 675–678, 2014.

[131] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692, 2015.

[132] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[133] Changxing Ding and Dacheng Tao. Robust face recognition via multimodal deep face representation. *IEEE transactions on Multimedia*, 17(11):2049–2058, 2015.

[134] Iftikhar Ahmad, Muhammad Hamid, Suhail Yousaf, Syed Tanveer Shah, and

Muhammad Ovais Ahmad. Optimizing pretrained convolutional neural networks for tomato leaf disease detection. *Complexity*, 2020:1–6, 2020.

[135] Ganesh Bahadur Singh, Rajneesh Rani, Nonita Sharma, and Deepti Kakkar. Identification of tomato leaf diseases using deep convolutional neural networks. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, 12(4):1–22, 2021.

[136] Divyansh Tiwari, Mritunjay Ashish, Nitish Gangwar, Abhishek Sharma, Suhanshu Patel, and Suyash Bhardwaj. Potato leaf diseases detection using deep learning. In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 461–466. IEEE, 2020.

[137] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. Deep-plant: Plant identification with convolutional neural networks. In *2015 IEEE international conference on image processing (ICIP)*, pages 452–456. IEEE, 2015.

[138] Pradeep Kumar Chaudhary, Kritiprasanna Das, and Ram Bilas Pachori. Breast cancer diagnosis using iterative fourier-bessel decomposition method based cnn-kernel features. 2022.

[139] Rishi Rai and Dilip Singh Sisodia. Real-time data augmentation based transfer learning model for breast cancer diagnosis using histopathological images. In *Advances in Biomedical Engineering and Technology: Select Proceedings of ICBEST 2018*, pages 473–488. Springer, 2021.

[140] Ya-nan Zhang, Ke-Rui Xia, Chang-Yi Li, Ben-Li Wei, and Bing Zhang. Review of breast cancer pathologigcal image processing. *BioMed research international*, 2021:1–7, 2021.

[141] Xujing Yao, Xinyue Wang, Shui-Hua Wang, and Yu-Dong Zhang. A comprehensive survey on convolutional neural network in medical image analysis. *Multimedia Tools and Applications*, pages 1–45, 2020.

[142] Krit Sriporn, Cheng-Fa Tsai, Chia-En Tsai, and Paohsi Wang. Analyzing lung disease using highly effective deep learning techniques. In *Healthcare*, volume 8, page 107. MDPI, 2020.

[143] Soumya Ranjan Nayak, Deepak Ranjan Nayak, Utkarsh Sinha, Vaibhav Arora, and Ram Bilas Pachori. Application of deep learning techniques for detection of covid-19 cases using chest x-ray images: A comprehensive study. *Biomedical Signal Processing and Control*, 64:102365, 2021.

[144] DR Sarvamangala and Raghavendra V Kulkarni. Convolutional neural networks in medical image understanding: a survey. *Evolutionary intelligence*, 15(1):1–22, 2022.

[145] Maryellen L Giger. Machine learning in medical imaging. *Journal of the American*

College of Radiology, 15(3):512–520, 2018.

[146] Dilip Singh Sisodia, Shruti Nair, and Pooja Khobragade. Diabetic retinal fundus images: Preprocessing and feature extraction for early detection of diabetic retinopathy. *Biomedical and Pharmacology Journal*, 10(2):615–626, 2017.

[147] Sai Lakshmi Bhamidipati, Sai Sudha Mindagudla, Harsha Vardhan Devalla, Hima Sagar Goodi, and Hemanth Nag. Analysis of different discrete wavelet transform basis functions in speech signal compression. *IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)*, 4(1):34–38, 2014.

[148] Shreyas Mistry and Arpita Patel. Image stitching using harris feature detection. *International Research Journal of Engineering and Technology (IRJET)*, 3(04):2220–6, 2016.

[149] Nilanjan Dey, Subhendu Das, and Pranati Rakshit. A novel approach of obtaining features using wavelet based image fusion and harris corner detection. *Int J Mod Eng Res*, 1(2):396–399, 2011.

[150] Hari Kumar Singh, SK Tomar, and Pooja Singh. Analysis of multispectral image using discrete wavelet transform. In *2013 Third International Conference on Advanced Computing and Communication Technologies (ACCT)*, pages 59–62. IEEE, 2013.

[151] Maneesha Gupta and Amit Kumar Garg. Analysis of image compression algorithm using dct. *International Journal of Engineering Research and Applications (IJERA)*, 2(1):515–521, 2012.

[152] Yu Hua, Wenbo He, Xue Liu, and Dan Feng. Smarteye: Real-time and efficient cloud image sharing for disaster environments. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1616–1624. IEEE, 2015.

[153] Yu Hua. Smart hashing based queries in the cloud. In *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2015.

[154] Chun-Che Chen and Shang-Lin Hsieh. Using binarization and hashing for efficient sift matching. *Journal of Visual Communication and Image Representation*, 30:86–93, 2015.

[155] Jinliang Yao, Bing Yang, and Qiuming Zhu. Near-duplicate image retrieval based on contextual descriptor. *IEEE signal processing letters*, 22(9):1404–1408, 2014.

[156] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[157] Deepak Geetha Viswanathan. Features from accelerated segment test (fast). In *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*, pages 6–8, 2009.

[158] Muhammad Kashif, Thomas M Deserno, Daniel Haak, and Stephan Jonas. Feature description with sift, surf, brief, brisk, or freak? a general question answered for bone age assessment. *Computers in biology and medicine*, 68:67–75, 2016.

[159] Bing Han, Dingyi Li, and Jia Ji. People detection with dsift algorithm.

[160] Jayanth Koushik. Understanding convolutional neural networks. *arXiv preprint arXiv:1605.09081*, 2016.

[161] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pages 253–256. IEEE, 2010.

[162] Liang Tan, Keping Yu, Ali Kashif Bashir, Xiaofan Cheng, Fangpeng Ming, Liang Zhao, and Xiaokang Zhou. Toward real-time and efficient cardiovascular monitoring for covid-19 patients by 5g-enabled wearable medical devices: A deep learning approach. *Neural Computing and Applications*, pages 1–14, 2021.

[163] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53:5455–5516, 2020.

[164] V Suma, R Amog Shetty, Rishab F Tated, Sunku Rohan, and Triveni S Pujar. Cnn based leaf disease identification and remedy recommendation system. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 395–399. IEEE, 2019.

[165] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[166] Junyi Chai, Hao Zeng, Anming Li, and Eric WT Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

[167] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2(6):420, 2021.

[168] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.

[169] Adnan Qayyum, Syed Muhammad Anwar, Muhammad Awais, and Muhammad Majid. Medical image retrieval using deep convolutional neural network. *Neurocomputing*, 266:8–20, 2017.

[170] Wenping Ma, Haoxiang Ma, Hao Zhu, Yating Li, Longwei Li, Licheng Jiao, and Biao Hou. Hyperspectral image classification based on spatial and spectral kernels

generation network. *Information Sciences*, 578:435–456, 2021.

[171] Paul Bergmann, Kilian Batzner, Michael Fauser, David Sattlegger, and Carsten Steger. The mvtec anomaly detection dataset: a comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision*, 129(4):1038–1059, 2021.

[172] Ching-Yung Lin and Shih-Fu Chang. A robust image authentication method distinguishing jpeg compression from malicious manipulation. *IEEE Transactions on Circuits and Systems for Video Technology*, 11(2):153–168, 2001.

[173] Satendra Pal Singh and Gaurav Bhatnagar. A robust image hashing based on discrete wavelet transform. In *2017 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pages 440–444. IEEE, 2017.

[174] Christoph Zauner. Implementation and benchmarking of perceptual image hash functions. 2010.

[175] Mengjuan Fei, Jing Li, and Honghai Liu. Visual tracking based on improved foreground detection and perceptual hashing. *Neurocomputing*, 152:413–428, 2015.

[176] Vishal Monga and Brian L Evans. Perceptual image hashing via feature points: performance evaluation and tradeoffs. *IEEE transactions on Image Processing*, 15(11):3452–3465, 2006.

[177] Pablo Chamoso, Alberto Rivas, Javier J Martín-Limorti, and Sara Rodríguez. A hash based image matching algorithm for social networks. In *Trends in Cyber-Physical Multi-Agent Systems. The PAAMS Collection-15th International Conference, PAAMS 2017 15*, pages 183–190. Springer, 2018.

[178] Nasir Ahmed, T_ Natarajan, and Kamisetty R Rao. Discrete cosine transform. *IEEE transactions on Computers*, 100(1):90–93, 1974.

[179] Mark J Shensa et al. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[180] Xuan Li, Liqiong Chang, and Xue Liu. Qhash: An efficient hashing algorithm for low-variance image deduplication. In *2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pages 9–15. IEEE, 2021.

[181] Mengjuan Fei, Zhaojie Ju, Xiantong Zhen, and Jing Li. Real-time visual tracking based on improved perceptual hashing. *Multimedia Tools and Applications*, 76:4617–4634, 2017.

[182] Andrew B Watson et al. Image compression using the discrete cosine transform. *Mathematica journal*, 4(1):81, 1994.

[183] M Narasimha and A Peterson. On the computation of the discrete cosine transform. *IEEE Transactions on Communications*, 26(6):934–936, 1978.

[184] Wen-Hsiung Chen, CH Smith, and Sam Fralick. A fast computational algorithm for the discrete cosine transform. *IEEE Transactions on communications*, 25(9):1004–1009, 1977.

[185] Randhir Kumar, Rakesh Tripathi, Ningrinla Marchang, Gautam Srivastava, Thippa Reddy Gadekallu, and Neal N Xiong. A secured distributed detection system based on ipfs and blockchain for industrial image and video data security. *Journal of Parallel and Distributed Computing*, 152:128–143, 2021.

[186] Qingying Hao, Licheng Luo, Steve TK Jan, and Gang Wang. It's not what it looks like: Manipulating perceptual hashing based applications. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 69–85, 2021.

[187] D de Martin-Roche, Carmen Sanchez-Avila, and Raul Sanchez-Reillo. Iris recognition for biometric identification using dyadic wavelet transform zero-crossing. In *Proceedings IEEE 35th Annual 2001 International Carnahan Conference on Security Technology (Cat. No. 01CH37186)*, pages 272–277. IEEE, 2001.

[188] Li Ma, Tieniu Tan, Yunhong Wang, and Dexin Zhang. Efficient iris recognition by characterizing key local variations. *IEEE Transactions on Image processing*, 13(6):739–750, 2004.

[189] Thangasamy Jeyapoovan and M Murugan. Surface roughness classification using image processing. *Measurement*, 46(7):2065–2072, 2013.

[190] Mohammed Gharib and MohammadAmin Fazli. Secure cloud storage with anonymous deduplication using id-based key management. *The Journal of Supercomputing*, 79(2):2356–2382, 2023.

[191] N Mageshkumar and L Lakshmanan. Intelligent data deduplication with deep transfer learning enabled classification model for cloud-based healthcare system. *Expert Systems with Applications*, 215:119257, 2023.

[192] Shubham Kumar, Soumya Mukherjee, and Arup Kumar Pal. An improved reduced feature-based copy-move forgery detection technique. *Multimedia Tools and Applications*, 82(1):1431–1456, 2023.

[193] An Qin, Mengbai Xiao, Ben Huang, and Xiaodong Zhang. Maze: A cost-efficient video deduplication system at web-scale. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3163–3172, 2022.

[194] Cai Deng, Qi Chen, Xiangyu Zou, Erci Xu, Bo Tang, and Wen Xia. imdedup: A lossless deduplication scheme to eliminate fine-grained redundancy among images. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages

1071–1084. IEEE, 2022.

[195] G Sujatha, D Hemavathi, K Sornalakshmi, and S Sindhu. Analysis of cryptographic hashing algorithms for image identification in deduplication process. In *Mobile Computing and Sustainable Informatics*, pages 813–823. Springer, 2022.

[196] Hana Matatov, Mor Naaman, and Ofra Amir. Dataset and case studies for visual near-duplicates detection in the context of social media. *arXiv preprint arXiv:2203.07167*, 2022.

[197] KK Thyagharajan and G Kalaiarasi. A review on near-duplicate detection of images using computer vision techniques. *Archives of Computational Methods in Engineering*, 28(3):897–916, 2021.

[198] An Qin, Mengbai Xiao, Yongwei Wu, Xinjie Huang, and Xiaodong Zhang. Mixer: efficiently understanding and retrieving visual content at web-scale. *Proceedings of the VLDB Endowment*, 14(12):2906–2917, 2021.

[199] Hengxiang Xie, Yuhui Deng, Hao Feng, and Lei Si. Pxdedup: deduplicating massive visually identical jpeg image data. *Big Data Research*, 23:100171, 2021.

[200] Wei Jia, Li Li, Zhu Li, Shuai Zhao, and Shan Liu. Scalable hash from triplet loss feature aggregation for video de-duplication. *Journal of Visual Communication and Image Representation*, 72:102908, 2020.

[201] Zhili Zhou, Meimin Wang, Yi Cao, and Yuecheng Su. Cnn feature-based image copy detection with contextual hash embedding. *Mathematics*, 8(7):1172, 2020.

[202] Shangpeng Yan, Xiaoyun Zhang, Li Chen, Wenbo Bao, and Zhiyong Gao. Large scale near-duplicate image retrieval via patch embedding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[203] Yeguang Li, Liang Hu, Ke Xia, and Jie Luo. Fast distributed video deduplication via locality-sensitive hashing with similarity ranking. *EURASIP Journal on Image and Video Processing*, 2019(1):1–11, 2019.

[204] Firoj Alam, Ferda Ofli, and Muhammad Imran. Processing social media images by combining human and machine computing during crises. *International Journal of Human–Computer Interaction*, 34(4):311–327, 2018.

[205] Hongyang Yan, Xuan Li, Yu Wang, and Chunfu Jia. Centralized duplicate removal video storage system with privacy preservation in iot. *Sensors*, 18(6):1814, 2018.

[206] Weiming Hu, Yabo Fan, Junliang Xing, Liang Sun, Zhaoquan Cai, and Stephen Maybank. Deep constrained siamese hash coding network and load-balanced locality-sensitive hashing for near duplicate image detection. *IEEE Transactions on Image Processing*, 27(9):4452–4464, 2018.

[207] S Preetha Bini and S Abirami. Secure image deduplication using spiht compres-

sion. In *2017 International Conference on Communication and Signal Processing (ICCSP)*, pages 0276–0280. IEEE, 2017.

[208] Zhili Zhou, QM Jonathan Wu, Fang Huang, and Xingming Sun. Fast and accurate near-duplicate image elimination for visual sensor networks. *International Journal of Distributed Sensor Networks*, 13(2):1550147717694172, 2017.

[209] Xuan Li, Jin Li, and Faliang Huang. A secure cloud storage system supporting privacy-preserving fuzzy deduplication. *Soft Computing*, 20(4):1437–1448, 2016.

[210] Amol S Deshmukh and PD Lambhate. A methodological survey on mapreduce for identification of duplicate images. *Int J Sci Res (IJSR)*, 5(1):206–210, 2016.

[211] Fatema Rashid, Ali Miri, and Isaac Woungang. Secure image deduplication through image compression. *Journal of Information Security and Applications*, 27:54–64, 2016.

[212] Fang Huang, Zhili Zhou, Tianliang Liu, and Xiya Liu. Original image tracing with image relational graph for near-duplicate image elimination. In *International Conference on Cloud Computing and Security*, pages 324–336. Springer, 2016.

[213] Yifeng Zheng, Xingliang Yuan, Xinyu Wang, Jinghua Jiang, Cong Wang, and Xiaolin Gui. Enabling encrypted cloud media center with secure deduplication. In *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, pages 63–72, 2015.

[214] Anum Javeed Zargar, Ninni Singh, Geetanjali Rathee, and Amit Kumar Singh. Image data-deduplication using the block truncation coding technique. In *2015 international conference on futuristic trends on computational analysis and knowledge management (ABLAZE)*, pages 154–158. IEEE, 2015.

[215] Xuan Li, Jie Lin, Jin Li, and Biao Jin. A video deduplication scheme with privacy preservation in iot. In *International symposium on computational intelligence and intelligent systems*, pages 409–417. Springer, 2015.

[216] Yu Hua, Hong Jiang, and Dan Feng. Fast: Near real-time searchable data analytics for the cloud. In *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 754–765. IEEE, 2014.

[217] Li Li and John Zic. Image matching algorithm based on feature-point and daisy descriptor. *J. Multim.*, 9(6):829–834, 2014.

[218] Yanqiang Lei, Guoping Qiu, Ligang Zheng, and Jiwu Huang. Fast near-duplicate image detection using uniform randomized trees. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(4):1–15, 2014.

[219] Bart Thomee, Mark J Huiskes, Erwin M Bakker, and Michael S Lew. An evaluation of content-based duplicate image detection methods for web search. In *2013 IEEE*

International Conference on Multimedia and Expo (ICME), pages 1–6. IEEE, 2013.

[220] Zhaofeng Li and Xiaoyan Feng. Near duplicate image detecting algorithm based on bag of visual word model. *Journal of Multimedia*, 8(5):557, 2013.

[221] Xin-Jing Wang, Lei Zhang, and Ce Liu. Duplicate discovery on 2 billion internet images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 429–436, 2013.

[222] S Leutenegger, M Chli, and RY Siegwart. Brisk: Binary robust invariant scalable keypoints stefan. In *2011 International Conference on Computer Vision*, 2012.

[223] Wei Dong, Zhe Wang, Moses Charikar, and Kai Li. High-confidence near-duplicate image detection. In *Proceedings of the 2nd acm international conference on multimedia retrieval*, pages 1–8, 2012.

[224] K Velmurugan and Lt Dr S Santhosh Baboo. Content-based image retrieval using surf and colour moments. *Global Journal of Computer Science and Technology*, 2011.

[225] Atul Katiyar and Jon Weissman. {ViDeDup}: An {Application-Aware} framework for video de-duplication. In *3rd Workshop on Hot Topics in Storage and File Systems (HotStorage 11)*, 2011.

[226] Jie Zhao, Li-Juan Xue, and Guo-Zun Men. Optimization matching algorithm based on improved harris and sift. In *2010 International Conference on Machine Learning and Cybernetics*, volume 1, pages 258–261. IEEE, 2010.

[227] Nikos Nikolaidis and Ioannis Pitas. Still image and video fingerprinting. In *2009 Seventh International Conference on Advances in Pattern Recognition*, pages 3–8. IEEE, 2009.

[228] Xin Yang, Qiang Zhu, and Kwang-Ting Cheng. Near-duplicate detection for images and videos. In *Proceedings of the First ACM workshop on Large-scale multimedia retrieval and mining*, pages 73–80, 2009.

[229] Xinghua Yu and Tiejun Huang. An image fingerprinting method robust to complicated image modifications. In *2008 International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, pages 227–230. IEEE, 2008.

[230] Ondrej Chum, James Philbin, Andrew Zisserman, et al. Near duplicate image detection: Min-hash and tf-idf weighting. In *Bmvc*, volume 810, pages 812–815, 2008.

[231] S H Srinivasan and Neela Sawant. Finding near-duplicate images on the web using fingerprints. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 881–884, 2008.

[232] Heng Tao Shen Xiaofang Zhou Zi and Huang Jie Shao Xiangmin Zhou. Uqlips: a real-time near-duplicate video clip detection system. *VLDB07*, pages 1374–1377,

2007.

[233] Jun Jie Foo, Ranjan Sinha, and Justin Zobel. Sico: a system for detection of near-duplicate images during search. In *2007 IEEE International Conference on Multimedia and Expo*, pages 595–598. IEEE, 2007.

[234] Marios A Gavrielides, Elena Sikudova, and Ioannis Pitas. Color-based descriptors for image fingerprinting. *IEEE transactions on multimedia*, 8(4):740–748, 2006.

[235] Xavier Naturel and Patrick Gros. A fast shot matching strategy for detecting duplicate sequences in a television stream. In *Proceedings of the 2nd international workshop on Computer vision meets databases*, pages 21–27, 2005.

[236] Chun-Shien Lu and Chao-Yong Hsu. Geometric distortion-resilient image hashing scheme and its applications on copy detection and authentication. *Multimedia systems*, 11(2):159–173, 2005.

[237] Jin S Seo, Jaap Haitsma, Ton Kalker, and Chang D Yoo. A robust image fingerprinting system using the radon transform. *Signal Processing: Image Communication*, 19(4):325–339, 2004.

[238] Jagpreet Singh and Nitin Auluck. Dvfs and duplication based scheduling for optimizing power and performance in heterogeneous multiprocessors. In *Proceedings of the High Performance Computing Symposium*, pages 1–8, 2014.

[239] Ali Alnoman, Glaucio HS Carvalho, Alagan Anpalagan, and Isaac Woungang. Energy efficiency on fully cloudified mobile networks: Survey, challenges, and open issues. *IEEE Communications Surveys & Tutorials*, 20(2):1271–1291, 2017.

[240] Santosh Chalise, Amir Golshani, Shekhar Raj Awasthi, Shanshan Ma, Bijen Raj Shrestha, Labi Bajracharya, Wei Sun, and Reinaldo Tonkoski. Data center energy systems: Current technology and future direction. In *2015 IEEE Power & Energy Society General Meeting*, pages 1–5. IEEE, 2015.

[241] P Sanjeevi and P Viswanathan. A green energy optimized scheduling algorithm for cloud data centers. In *2015 International Conference on Computing and Network Communications (CoCoNet)*, pages 941–945. IEEE, 2015.

[242] Dejene Boru, Dzmitry Kliazovich, Fabrizio Granelli, Pascal Bouvry, and Albert Y Zomaya. Energy-efficient data replication in cloud computing datacenters. *Cluster computing*, 18(1):385–402, 2015.

[243] Pei Huang, Benedetta Copertaro, Xingxing Zhang, Jingchun Shen, Isabelle Löfgren, Mats Rönnelid, Jan Fahlen, Dan Andersson, and Mikael Svanfeldt. A review of data centers as prosumers in district energy systems: Renewable energy integration and waste heat reuse for district heating. *Applied Energy*, 258:114109, 2020.

[244] Basappa B Kodada and Demian Antony D'Mello. Secure data deduplication (sd 2 e d up) in cloud computing: Threats, techniques and challenges. In *Advances*

in Communication and Computational Technology: Select Proceedings of ICACCT 2019, pages 1239–1251. Springer, 2020.

[245] Ricardo Bianchini and Ram Rajamony. Power and energy management for server systems. *Computer*, 37(11):68–76, 2004.

[246] Ravneet Kaur, Inderveer Chana, and Jhilik Bhattacharya. Data deduplication techniques for efficient cloud storage management: a systematic review. *The Journal of Supercomputing*, 74(5):2035–2085, 2018.

[247] Suruchi Talwani, Khaled Alhazmi, Jimmy Singla, Hasan J Alyamani, and Ali Kashif Bashir. Allocation and migration of virtual machines using machine learning. *Computers, Materials and Continua*, 70(2):3349–3364, 2022.

[248] Bin Li, Shilei Ding, and Xu Yang. A privacy-preserving scheme for jpeg image retrieval based on deep learning. In *Journal of Physics: Conference Series*, volume 1856, page 012007. IOP Publishing, 2021.

[249] Minxian Xu and Rajkumar Buyya. Managing renewable energy and carbon footprint in multi-cloud computing environments. *Journal of Parallel and Distributed Computing*, 135:191–202, 2020.

[250] Fang Yan, Xi Yang, Jiamou Liu, HengLiang Tang, Yu-An Tan, and YuanZhang Li. Optimizing the restoration performance of deduplication systems through an energy-saving data layout. *Annals of Telecommunications*, 74:461–471, 2019.

[251] Syed Arshad Ali, Mohammad Affan, and Mansaf Alam. A study of efficient energy management techniques for cloud computing environment. In *2019 9th international conference on cloud computing, data science & engineering (confluence)*, pages 13–18. IEEE, 2019.

[252] Sambit Kumar Mishra, Deepak Puthal, Bibhudatta Sahoo, Prem Prakash Jayaraman, Song Jun, Albert Y Zomaya, and Rajiv Ranjan. Energy-efficient vm-placement in cloud data center. *Sustainable computing: informatics and systems*, 20:48–55, 2018.

[253] Sambit Kumar Mishra, Deepak Puthal, Bibhudatta Sahoo, Sajay Kumar Jena, and Mohammad S Obaidat. An adaptive task allocation technique for green cloud computing. *The Journal of Supercomputing*, 74:370–385, 2018.

[254] Seyed Yahya Zahedi Fard, Mohamad Reza Ahmadi, and Sahar Adabi. A dynamic vm consolidation technique for qos and energy consumption in cloud environment. *The Journal of Supercomputing*, 73(10):4347–4368, 2017.

[255] Nguyen Trung Hieu, Mario Di Francesco, and Antti Ylä-Jääski. Virtual machine consolidation with multiple usage prediction for energy-efficient cloud data centers. *IEEE Transactions on Services Computing*, 13(1):186–199, 2017.

[256] Zhou Zhou, Zhigang Hu, and Keqin Li. Virtual machine placement algorithm for

both energy-awareness and sla violation reduction in cloud data centers. *Scientific Programming*, 2016, 2016.

[257] Awada Uchechukwu, Keqiu Li, Yanming Shen, et al. Energy consumption in cloud computing data centers. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, 3(3):31–48, 2014.

[258] Young Choon Lee and Albert Y Zomaya. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 60:268–280, 2012.

[259] Mohammed Hazim Alkawaz, Ghazali Sulong, Tanzila Saba, and Amjad Rehman. Detection of copy-move image forgery based on discrete cosine transform. *Neural Computing and Applications*, 30(1):183–192, 2018.

[260] Yusof Nbt, A Ismail, and NAA Majid. Deduplication image middleware detection comparison in standalone cloud database. *Int J Adv Comput Sci Technol (IJACST)*, 5(3):12–18, 2016.

[261] Haseeb Hassan, Ali Kashif Bashir, Muhammad Ahmad, Varun G Menon, Imran Uddin Afridi, Raheel Nawaz, and Bin Luo. Real-time image dehazing by superpixels segmentation and guidance filter. *Journal of Real-Time Image Processing*, 18:1555–1575, 2021.

[262] Suraj Srinivas, Ravi Kiran Sarvadevabhatla, Konda Reddy Mopuri, Nikita Prabhu, Srinivas SS Kruthiventi, and R Venkatesh Babu. A taxonomy of deep convolutional neural nets for computer vision. *Frontiers in Robotics and AI*, 2:36, 2016.

[263] Mattis Paulin, Matthijs Douze, Zaid Harchaoui, Julien Mairal, Florent Perronin, and Cordelia Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 91–99, 2015.

[264] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[265] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[266] Yu Han Liu. Feature extraction and image recognition with convolutional neural networks. In *Journal of Physics: Conference Series*, volume 1087, page 062032. IOP Publishing, 2018.

[267] Afshan Latif, Aqsa Rasheed, Umer Sajid, Jameel Ahmed, Nouman Ali, Naeem Iqbal Ratyal, Bushra Zafar, Saadat Hanif Dar, Muhammad Sajid, and Tehmina Khalil. Content-based image retrieval and feature extraction: a comprehensive review. *Mathematical Problems in Engineering*, 2019, 2019.

[268] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

[269] M-E Nilsback and Andrew Zisserman. A visual vocabulary for flower classification. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1447–1454. IEEE, 2006.

[270] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[271] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[272] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[273] Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: A benchmark dataset for performance evaluation of remote sensing image retrieval. *ISPRS journal of photogrammetry and remote sensing*, 145:197–209, 2018.

[274] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.

[275] Jonathan Takeshita, Ryan Karl, and Taeho Jung. Secure single-server nearly-identical image deduplication. In *2020 29th International Conference on Computer Communications and Networks (ICCCN)*, pages 1–6. IEEE, 2020.

[276] Manoj Diwakar and Manoj Kumar. A review on ct image noise and its denoising. *Biomedical Signal Processing and Control*, 42:73–88, 2018.

[277] Manoj Diwakar and Manoj Kumar. Ct image denoising using nlm and correlation-based wavelet packet thresholding. *IET Image Processing*, 12(5):708–715, 2018.

[278] Manoj Diwakar and Prabhishek Singh. Ct image denoising using multivariate model and its method noise thresholding in non-subsampled shearlet domain. *Biomedical Signal Processing and Control*, 57:101754, 2020.

[279] Xuchao Lu, Li Song, Rong Xie, Xiaokang Yang, Wenjun Zhang, et al. Deep binary representation for efficient image retrieval. *Advances in Multimedia*, 2017, 2017.

[280] Fengcai Qiao, Cheng Wang, Xin Zhang, and Hui Wang. Large scale near-duplicate celebrity web images retrieval using visual and textual features. *The Scientific World Journal*, 2013, 2013.

[281] Jun-yi Li and Jian-hua Li. Fast image search with deep convolutional neural networks and efficient hashing codes. In *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pages 1285–1290. IEEE, 2015.

[282] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 28, 2014.

[283] Huei-Fang Yang, Kevin Lin, and Chu-Song Chen. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):437–451, 2017.

[284] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[285] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[286] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[287] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[288] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

List of Publications

1. Ravneet Kaur, Inderveer Chana, Jhilik Bhattacharya *"Data deduplication techniques for efficient cloud storage management. A Systematic Review."*, The Journal of Supercomputing, Springer, 74(5):2035-2085, 2020. [SCI, IF 3.3]
2. Ravneet Kaur, Jhilik Bhattacharya, Inderveer Chana *"Deep CNN based online image deduplication technique for cloud storage system"*, Multimedia Tools and Applications, Springer, 1-34, 2022. [SCI, IF 3.6]
3. Ravneet Kaur, Jhilik Bhattacharya, Inderveer Chana, *"EsDeDUP: An Energy-saving image deduplication technique for Scalable Exact or Near-Exact Image Duplicate Detection"*..... [Communicated]