Signal Processing Systems
Mekelweg 4,
2628 CD Delft
The Netherlands
https://sps.ewi.tudelft.nl/

SPS-2023-5354560

M.Sc. Thesis

Efficient Content-Based Image Retrieval from Videos using Compact Deep Learning Networks with Re-ranking

Doruk Barokas Profeta

Abstract

The rise of streaming and video technologies highlights the need for efficient access and navigation of digital content, especially for scholars in history and art. Scholars seek streamlined methods to index, retrieve, and explore digital content, with an emphasis on finding specific instances. Searching for these instances in video content is intricate, involving video sequence analysis and relevant segment identification. Utilizing advanced techniques and algorithms is crucial for effective content-based retrieval.

In response to the escalating demand for accurate and swift access to relevant visual data in video resources, our research focuses on novel efficient content-based image retrieval from videos using deep learning. The system involves keyframe extraction, where significant frames are extracted, and content-based image retrieval, which finds similar frames to query images through feature extraction and ranking. This thesis analyzes various feature extraction techniques from compact deep learning networks and compares our proposed system to state-of-the-art systems for accuracy and speed. Our proposed method leverages compact deep learning network features for the first stage of ranking, effectively ranking frames, and subsequently incorporates re-ranking using a larger network. This approach presents a promising avenue to achieve high efficiency while maintaining valid accuracy in content-based video retrieval.

Efficient Content-Based Image Retrieval from Videos using Compact Deep Learning Networks with Re-ranking

Thesis

submitted in partial fulfillment of the
requirements for the degree of

Master of Science

in

Electrical Engineering

by

Doruk Barokas Profeta
born in Istanbul, Türkiye

This work was performed in:

Signal Processing Systems Group
Department of Microelectronics
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology

Delft University of Technology

Delft University of Technology
Department of
Microelectronics

The undersigned hereby certify that they have read and recommend to the Faculty of Electrical Engineering, Mathematics and Computer Science for acceptance a thesis entitled **"Efficient Content-Based Image Retrieval from Videos using Compact Deep Learning Networks with Re-ranking"** by **Doruk Barokas Profeta** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 20-09-2023

Chairman: _____

prof.dr.ir. Justin Dauwels

Advisor: _____

prof.dr.ir. Justin Dauwels

Committee Members: _____

dr. Jan van Gemert

_____

Abstract

The rise of streaming and video technologies has underscored the significance of efficient access and navigation of digital content, particularly for scholars in fields like history and art. Scholars actively seek streamlined approaches to index, retrieve, and explore digital content, with a focus on locating specific instances. The process of searching for specific instances in video search is complex that requires the analysis of video sequences and the identification of relevant video segments. Advanced techniques and algorithms are necessary to ensure effective content-based retrieval of the required information.

In response to the escalating demand for accurate and swift access to relevant visual data within the vast spectrum of video resources, our research has been dedicated to the development of novel, efficient content-based image retrieval methods tailored for videos by integrating deep learning methodologies. Our comprehensive system contains two crucial components: keyframe extraction and content-based image retrieval. Keyframe extraction involves identifying significant frames within videos, while content-based image retrieval enables the retrieval of similar frames to a query image through feature extraction and ranking.

A unique aspect of our research lies in the exploration and analysis of a diverse range of feature extraction techniques derived from compact deep learning networks. We have compared our proposed method with state-of-the-art retrieval systems, evaluating performance metrics in terms of both accuracy and speed. Our method harnesses the power of compact deep learning network features in the initial ranking stage, effectively sublisting frames, and subsequently introduces re-ranking using a larger network. This innovative approach promises to deliver the best of both worlds: exceptional efficiency without compromising retrieval accuracy. The code for our proposed system is available at https://github.com/dorukbarokas/Efficient-CBVIR.git.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Ir. Justin Dauwels, for his tireless guidance, invaluable insights, and continuous support throughout my thesis journey. His expertise and mentorship have been instrumental in shaping the direction of my research. I am truly grateful for all the opportunities he provided throughout this journey, allowing me to explore and grow in the field of machine learning.

I extend my heartfelt thanks to the Signal Processing Systems (SPS) group for providing me with exceptional opportunities and a collaborative environment in which to conduct my research.

My thesis project ran in parallel with Sinian Li's research, and I am immensely grateful for her dedication and outstanding work on keyframe extraction research. Her contributions have played a pivotal role in the development of our system, and I am thankful for her tireless commitment.

I also want to express my appreciation to my friends for their support and encouragement throughout this journey. Their friendship and shared experiences have made the challenges more manageable and the successes more enjoyable. Additionally, I extend my gratitude to my colleagues at Nevermore Tech. Their collaboration and shared expertise have been instrumental in my research and overall development.

I would like to extend my appreciation to Dr. Jan van Gemert for graciously accepting the invitation to join my thesis committee. Special thanks are due to Dr. Andrea Nanetti for generously providing us with access to a video database in historical research. This invaluable resource has greatly facilitated our exploration of the research topic, and I am deeply thankful for the opportunity to collaborate.

To my parents, Suzi and Moris, and my brother, Sarp, I owe a debt of gratitude for their relentless support and encouragement throughout my academic journey. Their guidance and belief in me have been a constant source of inspiration.

Lastly, I would like to express my heartfelt appreciation to my significant other, Selena, for patiently listening to my endless conversations about my graduation project, even though she had no clue about the concept. Her support and willingness to engage have been a source of immense comfort and encouragement throughout my journey.

Thank you to all those who have been a part of this journey, directly or indirectly, and have contributed to my growth and the successful completion of my thesis.

Doruk Barokas Profeta
Delft, The Netherlands
20-09-2023

Contents

Abstract **v**

Acknowledgments **vii**

1 Introduction **1**
1.1 Image Retrieval from Videos . . . 1
1.1.1 Problem Statement . . . 1
1.1.2 Objectives . . . 2
1.1.3 General Pipeline . . . 2
1.1.4 Contributions and Innovation . . . 4
1.2 Efficient Feature Extraction . . . 4
1.2.1 Significant Factors in Feature Extraction: . . . 5
1.3 Outline . . . 5

2 Related Work **7**
2.1 Background on Image Retrieval . . . 7
2.1.1 Early Approaches to Image Retrieval . . . 7
2.1.2 Content Based Image Retrieval(CBIR) . . . 8
2.2 Traditional Feature Extraction in Image Retrieval . . . 9
2.2.1 Local Descriptors for Image Retrieval . . . 10
2.3 Deep Learning in Image Retrieval . . . 14
2.3.1 Convolutional Neural Networks (CNNs) . . . 14
2.3.2 Notable Compact Network Architectures . . . 15
2.4 Video Image Retrieval . . . 19
2.4.1 Keyframe Extraction . . . 21
2.4.2 Nearest Neighbour Search . . . 21
2.5 Re-ranking in Image Retrieval . . . 22

3 Methodology **25**
3.1 Keyframe Extraction . . . 25
3.2 Feature Extraction . . . 26
3.2.1 Compact Deep Learning Network: MobileNetV2 Features . . . 26
3.2.2 Larger Deep Learning Network: ResNet101 + SOLAR Features 29
3.3 Search and Match Algorithms . . . 37
3.3.1 ANNOY: Approximate Nearest Neighbors with Tree-Based Methods . . . 37
3.3.2 Linear Search . . . 40
3.4 Storage and Optimization Strategies . . . 40
3.4.1 Shortlisting Keyframes . . . 41

4 Experiments and Results **43**
4.1 Experiment Setup 44
4.1.1 Datasets 44
4.1.2 Evaluation Metrics 49
4.2 Experiments 52
4.2.1 Tests 1: Filtering Algorithm 52
4.2.2 Test 2: Computation Time 54
4.2.3 Test 3: Accuracy 57
4.3 Discussion 58
4.4 Visualization of the Retrieval 59

5 Conclusion **61**
5.1 Summary 61
5.2 Future Work 61

List of Figures

1.1 General Pipeline of Content-Based Image Retrieval from Videos. 3

2.1 A Typical CBIR System Architecture [1]. 9
2.2 A Typical CNN Architecture [2]. 15
2.3 VGG16 Architecture [3]. 16
2.4 ResNet50 Architecture [4]. 18
2.5 The Representation of the Nearest Neighbour Search (The Example Provides Top 5 Neighbours Closer to Query) [5]. 22
2.6 The Representation of the Local Keypoints (Top) and Global Descriptors (Bottom). 23

3.1 An illustration of the SOLAR descriptor representing second-order spatial relations. The feature maps are re-weighted to provide a better global representation of the image [6]. 30
3.2 The Pipeline of SOLAR Global Features. Incorporating several Second-Order Attention (SOA) blocks at various stages of the ResNet101 backbone, along with subsequent steps involving GeM pooling, whitening, and L2 normalization, is performed [6]. 31
3.3 The pipeline of SOA layer [7]. 34
3.4 Configuration of the partitioning between hyperplanes [8]. 38
3.5 Building a binary tree in ANNOY [9]. 38
3.6 Searching with the binary tree in ANNOY [9]. 39
3.7 Configuration of the search process on hyperplanes [8]. 39

4.1 Example of a Query and Relevant Images per Category from ROxford5k Database. 45
4.2 Historical Videos Dataset. 46
4.3 Mapping Videos and Original Names. 47
4.4 Keyframe Extraction and Gallery Creation. 47
4.5 Annotation of Extracted Keyframes. 48
4.6 Selection and Validation Process for Query Images and Relevance Annotations. 48
4.7 Example of Query and Selected Ground Truth Relevant Frames. 49
4.8 Comparison of visual search results for easy and challenging tasks. . . . 60

List of Tables

2.1 Detailed structure of VGG16. 17
2.2 Detailed structure of ResNet50. 18
2.3 Detailed structure of MobileNetV2. 19

3.1 Detailed structure of MobileNetV2 [10]. 26
3.2 Detailed structure of ResNet101 [11]. 32

4.1 Summary of Databases. 45
4.2 Ranked list and Correctness of Ranking for Query 1. 51
4.3 Comparison of Average Ratios for Models from Compact Networks. . . 53
4.4 Comparison of Average Ratios for Models from Traditional Algorithms. 53
4.5 Performance Comparison of Different Models on Oxford5k and Paris6k Datasets. 53
4.6 Performance of Traditional Algorithms on Oxford5k and Paris6k. . . . 53
4.7 Comparison of Computation Time and Video Duration. 55
4.8 The Computation Time for Each Stage of Offline Phase. 55
4.9 The Computation Time for Each Stage of Online Phase. 55
4.10 Comparison of Duration of Feature Extraction per Frame for Different CNN Backbones. 56
4.11 Comparison of mAP (%) for Different Models on Oxford5k and Paris6k Databases. 57
4.12 Comparison of mAP (%) for Different Models on ROxford5k and RParis6k Databases. 57
4.13 Comparison of mAP (%) for Different Models on Easy, Medium, and Hard Splits of ROxford5k and RParis6k Databases with 1M Distractors. 58
4.14 Comparison of mAP (%) for Different Models on Historical Videos Database. 58

1 Introduction

The main objective of this project is to construct an effective and reliable content-based retrieval system for video images, enabling the automatic identification of query images within video content. While the potential applications of this system are broad, our primary focus is directed toward its particular utility in historical research. This initiative is an integral part of the broader project known as 'Engineering Historical Memory' (EHM), spearheaded by Dr. Andrea Nanetti. The central aim of this project is to leverage emerging digital technologies, including artificial intelligence, with the purpose of consolidating and sharing historical knowledge [12].

This chapter provides an introduction to the video-based image search engine powered by deep learning. We start with the project's underlying reasoning and goals and then delve into a comprehensive overview of the entire process. Notably, this thesis focuses on the content-based image retrieval aspect of the proposed pipeline, which will be elaborated on in detail further. The concluding section gives an outline of the thesis's layout.

1.1 Image Retrieval from Videos

Image retrieval from videos refers to the process of identifying and extracting specific frames from video content that matches a given query image. This could be based on visual features, semantic content, or a combination of both. Essentially, instead of sifting through an entire video or multiple videos manually to locate a particular scene or image, this automated process rapidly scans and identifies the relevant frames, streamlining access to desired visual content within vast video databases.

1.1.1 Problem Statement

With the evolution of streaming and video technologies, fields such as history and art have seen a revolutionary shift in how information is conveyed. Efficiently accessing, filtering, and navigating digital historical content is a challenge for scholars. Therefore, optimizing research endeavors is urgent. The aspiration to maximize the utility of time and resources has transformed from a mere desire to an essential need.

Deep learning has emerged as a promising solution in this environment [13]. By refining searches, trimming away redundant data, and spotlighting only the most relevant content, deep learning techniques have revolutionized the research process. This technological innovation has not only rendered searching more effective but has also provided scholars with the tools necessary to harness the overwhelming volume of data that now lies at their fingertips.

However, the challenge persists in constructing an adept content-based image retrieval (CBIR) system capable of proficiently retrieving relevant images from videos. One promising method is the adoption of compact deep-learning networks [14]. These models, characterized by their diminished layers or parameters, contrast with their deeper, more intricate counterparts. While compact networks bring to the table benefits like reduced computational complications, swifter processing speeds, and diminished memory demands—qualities that make them indispensable in scenarios with limited resources or when real-time actions are essential—their architecture can have both benefits and drawbacks. Their streamlined design, although encouraging interpretability, may fail when tasked with capturing challenging visual representations. This could compromise retrieval efficacy, potentially sidelining critical details and subtle concepts that deeper networks might effortlessly discern.

Hence, the main problem we confront is: The need to develop an efficient content-based video image retrieval (CBVIR) system that can adeptly pull relevant images from videos. This system should integrate the advantages of compact deep learning networks while incorporating re-ranking techniques, thereby ensuring both high retrieval accuracy and computation speed.

1.1.2 Objectives

Motivated by a multitude of practical applications, we aim to introduce and verify our image retrieval method that meets the subsequent objectives:

1. Enabling accurate representation of image content feature vectors by at most 5% drop of mAP compared to a state-of-the-art system.

2. The feature representation is efficient and adaptable enough to support many types of datasets as digital historical media.

3. The search performance of the semantic features of image material is effective and can reply fast and precisely to user search query requests by at least 10 times faster than the duration of input videos.

4. To establish a video-based image search system for digital historical materials and construct an open-source retrieval platform with an improved model algorithm and scheme.

1.1.3 General Pipeline

We present a two-stage strategy for our image retrieval task: initially detecting the keyframes and subsequently engaging in content-based image retrieval (CBIR), as portrayed in Figure 1.1.

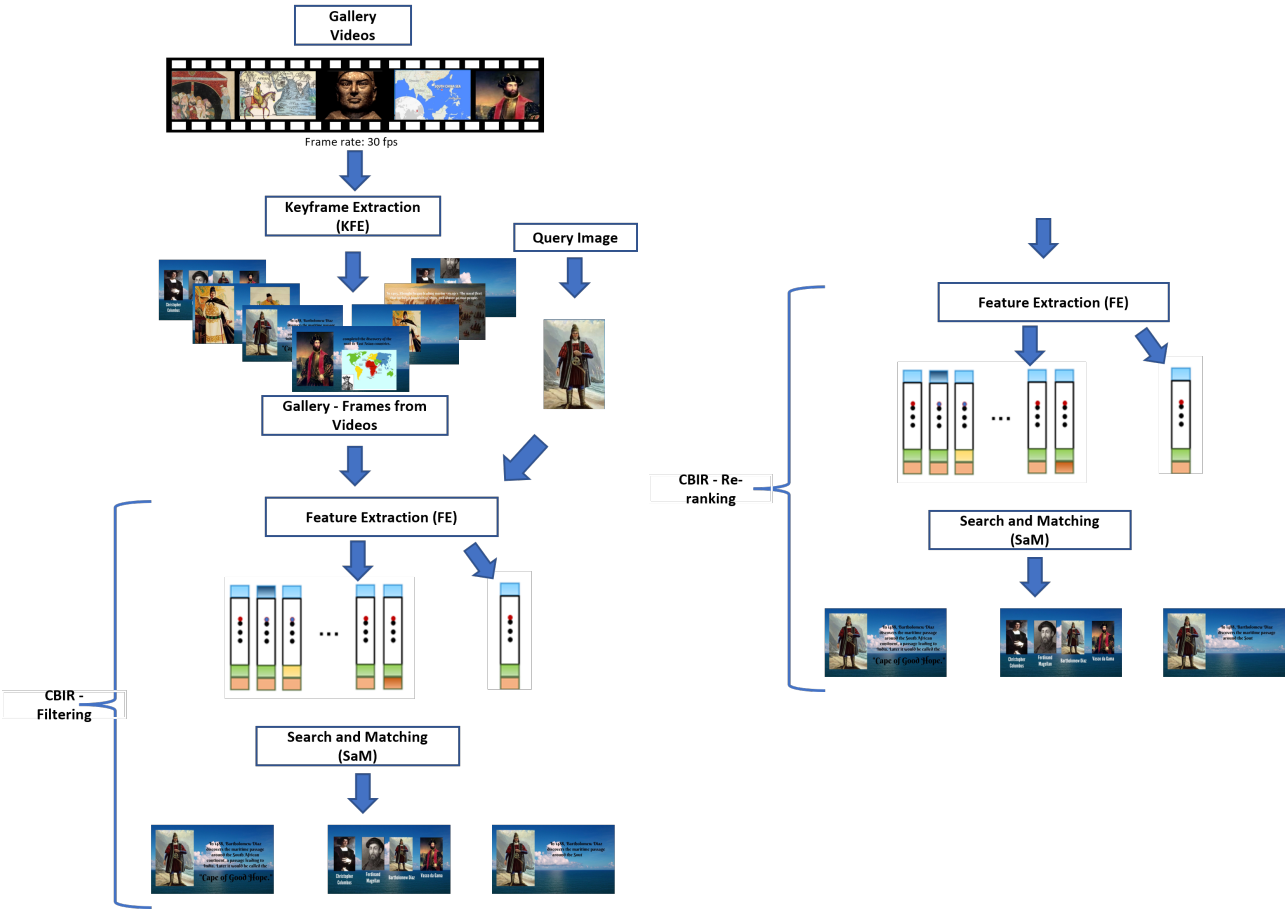


Figure 1.1: General Pipeline of Content-Based Image Retrieval from Videos.

In the first stage, our innovative video image retrieval approach initiates by subsampling the input videos. Given that videos typically consist of numerous frames, we extract keyframes from these videos to efficiently analyze and retrieve relevant visual content. This is achieved by employing a color histogram-centric clustering mechanism. To further enhance the process's efficacy, Singular Value Decomposition (SVD) is employed for dimensionality reduction on the feature matrix. By leveraging cosine similarity, the clustering algorithm assesses the feature space, facilitating the extraction of keyframes. This technique ensures that while crucial frames are singled out, essential data remains intact, fortifying subsequent retrieval steps.

Transitioning to the second phase, CBIR can be broadly split into Feature Extraction (FE) and the Search and Match (SaM) processes. Here, feature extraction emerges as a pivotal element, instrumental in striking a balance between efficiency and precision. Initial feature extraction is executed via a pre-trained, compact CNN model, namely MobileNetV2 [10]. Remarkably, compared with other cutting-edge pre-trained frameworks, MobileNetV2 exhibits superior computational performance, thus positioning it as the most rapid alternative for video querying within a CBIR framework. Its efficiency is underlined by its capability to operate at double the speed of contenders like

VGG16 or ResNet50. While employing a less complex network as the foundation results in time savings, it might marginally compromise accuracy, resulting trade-off. The search procedure capitalizes on approximate nearest-neighbor (ANN) methodologies to discern the top-K features resonating with the query attribute, defining this segment of CBIR as the filtration process. Through relevant threshold selections, non-essential frames are strategically filtered out.

Addressing potential accuracy drops, a subsequent re-ranking module is introduced, harnessing global attributes to refine retrieval outcomes. This utilizes a CNN configuration of ResNet101 combined with SOLAR (Second-Order Loss and Attention for Image Retrieval). This prototype was co-developed by Sinian Li and myself. I was responsible for the development of the Content Bases Image Retrieval part by using CNN-based feature extraction with filtering using compact networks and jointly building the final retrieval system.

1.1.4 Contributions and Innovation

The primary contribution of this thesis lies in showcasing a content-based video image retrieval system that prioritizes speed, tolerating a nominal accuracy decline, thus making it a valuable asset for historical researchers aiming to expedite their investigative endeavors.

Our work introduces a novel approach to image retrieval from videos, significantly transforming the landscape of historical research methodologies. The center of our innovation is the development of an efficient CBIR system. Our primary contribution lies in the enhancement of the CBIR phase where we've built a filtering stage. This stage is adept at meticulously filtering out irrelevant frames from the gallery of keyframes, ensuring that researchers are presented with only the most relevant visual data. The keystone to this filtering process is our choice of the compact network - MobileNetV2. By harnessing the computational efficiency and streamlined architecture of MobileNetV2, we achieve rapid filtering without compromising the richness of relevant data. This approach not only amplifies the speed but also provides an innovative method of feature extraction, setting a new benchmark for CBIR systems. As such, our approach stands as a testament to the integration of advanced deep learning techniques in the service of academic and historical research with swift, efficient, and accurate video-based image retrieval.

1.2 Efficient Feature Extraction

Imagine you are trying to describe a friend to someone who has never met them. You would not describe every tiny detail about your friend. Instead, you would mention the most noticeable features, such as "she has curly red hair," or "he is really tall and always wears glasses." In this scenario, these distinguishing attributes like hair color, height, or always wearing glasses are the 'features' you have extracted to describe your friend. Similarly, for an image, feature extraction involves identifying and describing the unique parts or patterns that can help recognize or categorize that image. It is like the computer's way of understanding or describing an image.

In the case of 'efficient' feature extraction, it is like being able to describe your friend quickly yet accurately. In the world of images, this means identifying those unique parts or patterns in an image without spending too much time or computer resources. The essential factors are fast and precise.

1.2.1 Significant Factors in Feature Extraction:

1. **Adaptability:** A major quality of the proposed feature extraction methodology is its adaptability across countless image types. This can be understood as the adeptness in determining challenging tasks of comparing buildings in Oxford, a testament to its versatile capabilities.

2. **Reliability:** Central to the effectiveness of any feature extraction technique is its reliability. Upon repeated exposure to an identical or similar image, the procedure should invariably yield consistent features. A suitable analogy would be the persistent recognition of a familiar individual predicated upon a distinctive attribute, such as curly hair.

3. **Scalability:** A robust feature extraction algorithm should be resilient to the scale of data it encounters. Identical to the capability of consistently recognizing an expanding circle of understandings without failing in descriptiveness, the method should remain productive regardless of the magnitude of the dataset.

4. **Relevance:** The extracted features must encapsulate the salient characteristics of an image. The emphasis should align with consistently defining attributes rather than transient or irrelevant details. One might compare this to highlighting notable traits like hair color and facial features over mutable aspects like clothing.

In summation, the essence of efficient feature extraction in imaging is to fast and accurately contrast pivotal and distinct patterns within the image. Such work should consistently maintain adaptability, reliability, and relevance, irrespective of the volume of images encountered.

1.3 Outline

The outline of this thesis is as follows:

Chapter 1 initiates with an introduction to a digital historical video image retrieval system. It delves into the motivation behind the research, and the set objectives, and offers a comprehensive view of the pipeline. This chapter underscores the primary aims and the significance of the image retrieval module, before discussing the methodology employed. It concludes by presenting the overall framework of the thesis.

Chapter 2 delves into the exploration of various aspects related to content-based image retrieval systems, beginning with a review of early approaches to image retrieval, including traditional feature extraction methods such as Local Descriptors (e.g., SIFT,

ORB) and Compact CNN Networks. Additionally delves into the domain of deep learning in image retrieval, focusing on Convolutional Neural Networks and notable compact network architectures like VGG, ResNet, and MobileNetV2. Moreover, presents video image retrieval techniques, including sections on keyframe extraction, nearest neighbor search, and re-ranking in image retrieval.

Chapter 3 provides details of the key steps and techniques involved in our image retrieval system, beginning with keyframe extraction and feature extraction methods with mentioning the distinct separation between the online and offline phases. Followed by an exploration of compact and larger deep learning network architectures (MobileNetV2 and ResNet101 + SOLAR features), and then delving into search and match algorithms, including ANNOY for approximate nearest neighbors and linear search. Additionally, the chapter includes a discussion of storage and optimization strategies, with a particular focus on shortlisting keyframes.

Chapter 4 provides a comprehensive analysis of our proposed image retrieval system's performance through a series of tests and experiments. These experiments will encompass tests related to the filtering algorithm selection of compact network, computation time, and accuracy, with each test exploring specific aspects of the system's functionality. Further provides details on the experiment setup, including the datasets used, evaluation metrics such as Mean Average Precision (mAP), and computation time, and then delves into the results obtained from these experiments. Additionally, a discussion will be included to provide insights and interpretations of the findings.

Chapter 5 summarizes the main contributions of the work and offers insights for potential future research avenues.

2 Related Work

This chapter delves into the landscape of video-based image retrieval, underscoring its evolution and current methodologies. Beginning with exploring the conception and development of image retrieval in Section 2.1, we delve into the historical development of image retrieval, including early approaches and the emergence of Content-Based Image Retrieval. Section 2.2 monitors efficient traditional feature extraction methods, encompassing local descriptors like SIFT[15], ORB [16], and AKAZE [17]. Moving on to section 2.3, we focus on the transformative impact of deep learning in image retrieval, emphasizing the significance of Convolutional Neural Networks (CNNs) and highlighting notable compact network architectures such as VGG [18], ResNet [11], and MobileNetV2 [10]. In the final section, 2.4, our exploration extends to video image retrieval, covering keyframe extraction, nearest neighbor search, and the pivotal role of re-ranking. Throughout this chapter, we trace the evolution of image retrieval, providing insights into both traditional and modern methodologies, and ultimately setting the stage for the subsequent research chapters.

2.1 Background on Image Retrieval

Since its inception, image retrieval as a domain has gone through enormous developments. Historically, the principal methods of image retrieval were based on text-based annotations, in which images in databases were manually labeled with descriptive terms, allowing search and retrieval based on textual queries [19, 20]. While beneficial, such systems have inherent drawbacks such as the tediousness of human labeling, the subjectivity of descriptions, and the inability to express the rich essence of an image with words alone. As visual data grew exponentially in the digital era, the necessity for more advanced, autonomous, and content-centric retrieval methods became clear. This paved the way for Content-Based Image Retrieval (CBIR), a game-changing strategy that enabled systems to search and retrieve images based on their inherent visual properties such as patterns, colors, and dimensions [21].

2.1.1 Early Approaches to Image Retrieval

In the initial stages of image retrieval, text-based image retrieval (TBIR) methods dominated, where images were annotated with textual descriptions for retrieval purposes [19, 20]. While straightforward, this approach had notable limitations. Manual tagging was a primary method, which was both time-intensive and inconsistent due to its reliance on human interpretation. As image databases expanded, the challenges of maintaining uniformity across tags became apparent. Moreover, the inherent subjectivity of textual descriptions often led to varied interpretations of an image's content, resulting

in retrieval mismatches. This inconsistency underscored the need for more objective, content-driven retrieval techniques. Consequently, image retrieval based on text does not reach optimal efficiency and effectiveness [21]. To address these constraints of TBIR systems and find a better search method, more intuitive and user-centric content-based image retrieval systems were introduced.

2.1.2 Content Based Image Retrieval(CBIR)

Content-based image retrieval, also known as Query By Image Content (QBIC) [22], represents an innovative approach to image searching. Instead of relying on textual descriptors, CBIR processes use an image's inherent visual attributes like color, texture, shape, and spatial arrangements to identify and retrieve similar images from a database [21]. When a user provides an image or sketch, the system converts it into a feature vector, similar to those in the database. It then calculates the similarity or distance between the query's feature vector and those of the target images, using an efficient indexing mechanism to facilitate retrieval.

The advantages of CBIR over traditional text-based image retrieval are significant. Firstly, CBIR offers a retrieval method that aligns more closely with human visual perception, eliminating the need for severe manual annotations. Furthermore, to refine the process, many systems incorporate user feedback, ensuring that results are not only visually but also semantically relevant.

Despite its advantages, CBIR faces challenges, primarily the 'semantic gap'. This gap emphasizes the disparity between low-level features that computers extract and the high-level human interpretations of images [23, 24]. Bridging this gap has been a central challenge in CBIR research, and various strategies, such as machine learning, relevance feedback, and semantic templates, have been developed to address it.

Historically, the journey of CBIR has been marked by significant advancements since the 1990s. The USA's National Science Foundation [25], recognizing the need for more intuitive image management solutions, catalyzed early discussions. This era heralded a collaboration of experts from diverse fields like computer vision, database management, and human-computer interaction. Since then, both research and practical applications in the domain have expanded exponentially, with a myriad of systems being introduced worldwide.

In summary, CBIR stands at the intersection of computer vision and human interpretation. While it offers automated, efficient image retrieval based on visual features, its true challenge and potential lie in harmonizing this process with the complex realm of human perception.

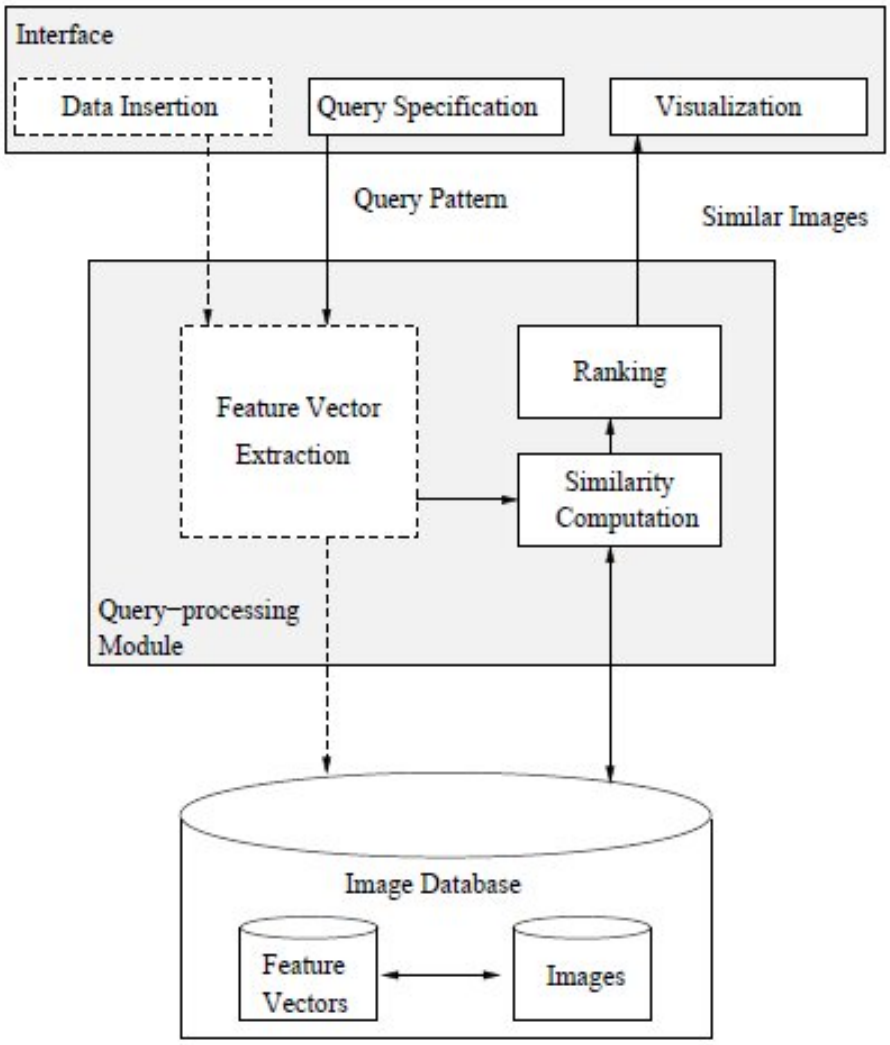


Figure 2.1: A Typical CBIR System Architecture [1].

2.2 Traditional Feature Extraction in Image Retrieval

Before the emergence of deep learning methodologies, which enable computers to learn image features directly from data, traditional feature extraction techniques served as the cornerstone of CBIR systems. These handcrafted features, meticulously designed by experts, were tailored to capture specific visual properties of images such as color [26, 27], texture [28, 29], and shape [30, 31]. In contrast to learning-based approaches, these handcrafted features are predefined and are not learned from data. During their peak usage, these techniques played a critical role in a wide array of applications, including image retrieval. Despite the rise of deep learning, these traditional techniques continue to be valued for their unique strengths, especially in scenarios where interpretability, computational efficiency, and resistance to overfitting are paramount.

This section provides an overview of some of the most influential traditional fea-

ture extraction methods, shedding light on their key principles and delineating their enduring relevance in the ever-evolving landscape of image retrieval.

2.2.1 Local Descriptors for Image Retrieval

Local feature descriptors [32] are techniques that extract and describe localized features or key points from images. These key points are distinctive and can be used for tasks like image matching, object recognition, and image retrieval. The approach of local feature extraction is more efficient compared to the global feature extraction process.

2.2.1.1 Scale-Invariant Feature Transform (SIFT)

SIFT [15], introduced by David Lowe in 1999, is a well-regarded algorithm for detecting and describing local features inside the target images. The key strengths of SIFT include its invariance to image scaling, rotation, and its robustness to changes in viewpoint and illumination. SIFT works by identifying key points (or interest points) in an image, and then computing a descriptor for each key point based on the local image gradients.

The algorithm demonstrates resilience against alterations in scale, rotation, and illumination conditions. The SIFT methodology can be divided into four primary stages [33]:

1. **Scale-Space Extreme Detection:**
   - Scanning the entire image to identify potential keypoints.
   - Constructing the scale-space.
   - Using Gaussian blurring to approximate the Laplacian.
2. **Keypoint Localization:**
   - Selecting stable keypoints resilient to changes in scale and orientation.
   - Removing less significant keypoints to diminish noise.
3. **Orientation Determination:**
   - Assigning a consistent orientation for each keypoint that remains unaltered across image transformations.
4. **Descriptor Generation for Keypoints:**
   - Formulating vectors that characterize each feature of keypoints.

During the construction of the scale space, the image undergoes re-scaling to determine pronounced and enduring features, resulting in the creation of octaves. Following this, a scale-space pyramid is assembled, encompassing these octaves ranked from the largest to the smallest. The subsequent phase involves the application of Gaussian blur, facilitated by a specific Gaussian operator,

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.1}$$

where L represents the resulting image, G denotes the Gaussian operator, and I is the input image. Subsequently, the Laplacian is determined to identify edges. Ideally, this should involve calculating the second derivative. However, this operation is computationally intensive. To address this challenge, the Difference of Gaussians (DoG) method is employed. The ensuing phase in the SIFT algorithm pertains to the localization of keypoints, containing two crucial procedures:

- Detecting local extrema on DoG images.
- Defining the position of these extrema.

The strategy for local extrema detection involves comparing pixel values with their neighboring counterparts. In a discrete image representation, the most luminous pixel does not always coincide with the position of the local extrema. To rectify this inconsistency, Taylor's theorem is utilized.

$$D(x) = D + \frac{\partial D^T}{\partial x}x + \frac{1}{2}x^T\frac{\partial^2 D}{\partial x^2}x \tag{2.2}$$

Each keypoint is characterized by two attributes: intensity and the direction in which it points. These are determined by the gradients of surrounding pixels. The resulting SIFT keypoint descriptor comprises two vectors. The primary vector encompasses the point's coordinates (x, y), the identified scale, the feature's response or intensity, orientation (measured counter-clockwise from the positive x-axis), and the Laplacian's sign (utilized for rapid matching). The secondary vector carries a descriptor of length 128 [34].

2.2.1.2 Oriented FAST and Rotated BRIEF (ORB)

ORB [16] is a fast binary descriptor based on the combination of the FAST [35] keypoint detector and the BRIEF [36] descriptor. It was developed as a free alternative to patented algorithms like SIFT. ORB is rotation invariant and resistant to noise. By combining the strengths of FAST and BRIEF, ORB provides a highly efficient and robust method for feature extraction and matching.

To break down the algorithm,

1. FAST Keypoint Detection

The FAST (Features from Accelerated Segment Test) algorithm examines a circle of sixteen pixels around the corner candidate p. The symbol I denotes the intensity function of the image, where given a pixel location p, $I(p)$ provides the intensity value of the image at that pixel location. If a set of n contiguous pixels in the circle are either brighter or darker than the intensity of the candidate p by a threshold t, then p is considered a corner. Mathematically:

$$I(p) + t < I(p_i) \quad \text{or} \quad I(p) - t > I(p_i) \tag{2.3}$$

where p_i are the pixels in the circle around p.

2. Orientation Assignment

For each feature detected by FAST, ORB computes the centroid C using:

$$C = \left(\sum m_{01} \times x, \sum m_{10} \times y \right) \tag{2.4}$$

where m_{01} and m_{10} are the central image moments. The orientation θ is then:

$$\theta = \arctan\left(\frac{m_{01}}{m_{10}}\right) \tag{2.5}$$

3. Rotation-Invariant BRIEF Descriptor

ORB utilizes the rBRIEF [37] descriptor, a rotated version of BRIEF. The learning of pair (x, y) in BRIEF is rotated by θ:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{2.6}$$

4. Binary Descriptor

Given the rotated test locations, the rBRIEF descriptor is computed. For each pair of locations (x_i, y_i) and (x_j, y_j), the binary test is:

$$\tau(p; x_i, x_j) = \begin{cases} 1 & \text{if } I(p + x_i) < I(p + x_j) \\ 0 & \text{otherwise} \end{cases} \tag{2.7}$$

Here, $I(p + x_i)$ and $I(p + x_j)$ represent the intensity values at pixel locations offset by x_i and x_j from the keypoint p, respectively. The BRIEF descriptor is then the concatenated results of these binary tests.

5. Efficiency Techniques

ORB processes the image in resized versions to detect features at multiple scales using a pyramid approach. The Harris corner measure is:

$$R = \det(M) - k(\text{trace}(M))^2 \tag{2.8}$$

where M is the structure tensor matrix and k is an empirically determined constant.

The ORB algorithm stands out as an efficient and effective feature descriptor, especially in comparison to other traditional methods such as SIFT [38]. Its combination of the FAST keypoint detector with the rBRIEF descriptor results in a binary descriptor that's computationally more efficient than its predecessors. Furthermore, ORB's rotation invariance and resistance to noise make it particularly robust in practical applications. ORB offers a compelling balance between performance and computational efficiency, reinforcing its popularity in numerous computer vision tasks.

2.2.1.3 Accelerated KAZE (AKAZE)

AKAZE [17] is an evolution of the KAZE [39] algorithm, designed to improve speed without compromising performance. The term 'KAZE' is derived from the Basque word for 'wind', and 'AKAZE' from the word for 'fast wind'. AKAZE is particularly adept at handling wide baselines and significant rotations. It employs a novel approach to scale-space called the nonlinear scale space, which helps it detect features in images more effectively. Like SIFT, AKAZE provides a descriptor for each detected key point, which can be used for matching [38].

1. Nonlinear Scale Space

AKAZE uses a nonlinear scale space, which is built using a diffusion process rather than Gaussian blurs. The diffusion equation for this is:

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, t)\nabla L) \tag{2.9}$$

where L is the image at a given scale, c is the conductivity function (which controls the amount of diffusion), and ∇L is the gradient of the image.

2. Keypoint Detection

Keypoints are detected by finding the local extrema across scales and space in the nonlinear scale space. The determinant of the Hessian matrix is used for this purpose:

$$\text{Det}(H) = D_{xx} \times D_{yy} - (D_{xy})^2 \tag{2.10}$$

where D_{xx}, D_{yy}, and D_{xy} are the second order partial derivatives.

3. Orientation Assignment

Similar to other feature detectors, AKAZE also assigns an orientation to each keypoint to achieve rotation invariance. The orientation θ is computed using gradient information around the keypoint.

4. Modified Local Difference Binary (MLDB) Descriptor

AKAZE uses a descriptor called Modified Local Difference Binary (MLDB). It computes the binary string by comparing the intensity differences between pixels around the keypoint:

$$\tau(p; x_i, x_j) = \begin{cases} 1 & \text{if } L(p + x_i) < L(p + x_j) \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

This binary descriptor is more memory efficient and faster to compute than traditional descriptors.

5. Multi-scale Feature Matching

AKAZE utilizes a multi-scale approach to detect features at various resolutions, enhancing its robustness against varying image scales.

The AKAZE algorithm is recognized for its efficiency, especially when comparing the time-performance trade-offs. Its use of a nonlinear scale space and a binary descriptor results in a fast and memory-efficient feature detection method. Furthermore, AKAZE's capability to reliably detect features across different scales makes it versatile for various computer vision applications like image retrieval.

2.3 Deep Learning in Image Retrieval

Over the past decade, deep learning has not only emerged as a powerful tool but has also radically transformed the domain of image retrieval [40]. The intricate design and architectures of neural networks, notably CNNs, stand as a testament to this revolution. Their inception has steered the field towards outstanding levels of accuracy across a variety of image retrieval benchmarks. What sets these networks apart is their adeptness at seamlessly integrating low-level image features into more comprehensive, high-level representations. This transformative shift has been characterized by a marked move from hand-crafted classical image-processing feature representation to a learning-based approach, largely attributable to the emergence of deep learning [41]. This is achieved through intricate non-linear transformations, which act as the cornerstone for deriving deep, semantically rich interpretations directly from visual data.

Taking a closer look, CNNs emerge as a pivotal entity that has left a memorable mark on the broader spectrum of computer vision [40]. Their dominance is underpinned by an overload of empirical studies, each echoing their remarkable performance over conventional feature extraction techniques. The wide array of tasks where CNNs includes scene recognition, fine-grained recognition, attribute detection, and image retrieval [42].

Their success is not merely a sensation in academia; the real-world implications are profound. This is evident as image search services, with an emphasis on image-by-image search, have trended across mainstream search engines.

2.3.1 Convolutional Neural Networks (CNNs)

CNNs are specialized neural architectures specifically tailored for processing grid-like data structures, with images being their primary focus. Originating from their foundational design, these networks boast convolutional layers. These layers are recognized by their ability to automatically and adaptively discern spatial hierarchies of features, making them particularly adept at recognizing patterns, textures, and objects within visual data.

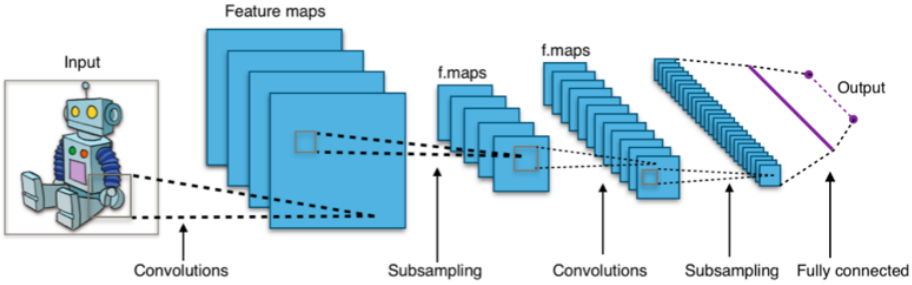


Figure 2.2: A Typical CNN Architecture [2].

Furthermore, the resilience and versatility of CNNs stem from their multi-layered structure, which allows them to capture both coarse details and broad contextual information from images. As they process an image, early layers typically detect simple features like edges, while deeper layers capture more complex structures and patterns.

Over the years, the consistent superiority of CNNs over traditional image-processing techniques has become evident. Their ability to learn from vast amounts of data, adapt to varied tasks, and provide precise results has made them the preferred choice in numerous visual recognition challenges [40]. This relentless ascent and unmatched performance have solidified their position as the gold standard in image retrieval, setting a benchmark that's hard to surpass [42].

2.3.2 Notable Compact Network Architectures

In the evolving landscape of deep learning, there's a rapidly increasing interest in compact neural network architectures. As deep learning models, notably CNNs, expand in complexity, there arises an inherent trade-off between computational time and accuracy. While deeper architectures can potentially offer better performance, they invariably require greater computational resources and storage [40]. This often renders them unsuitable for real-time applications or deployment on devices with limited resources, like smartphones or edge devices. Compact architectures elegantly navigate this trade-off by designing efficient, yet powerful models. By leveraging techniques such as parameter pruning, quantization, and knowledge distillation, these architectures manage to drastically reduce the model's size without a proportional decline in performance. The significance of these shallow structures cannot be overdrawn, particularly in the age of widespread computing. They enable the deployment of sophisticated deep learning capabilities on devices with limited processing power, ensuring that advanced image retrieval and recognition tasks can be executed swiftly, even in decentralized settings.

VGG

The VGG [18] network, short for Visual Geometry Group, represents a significant milestone in the development of convolutional neural networks (CNNs). Its architecture, characterized by its depth and simplicity, laid the foundation for many subsequent neural network designs. One notable variant of the VGG architecture is VGG16, which

comprises 16 weight layers, as well as an additional fully connected layer, making a total of 19 layers.

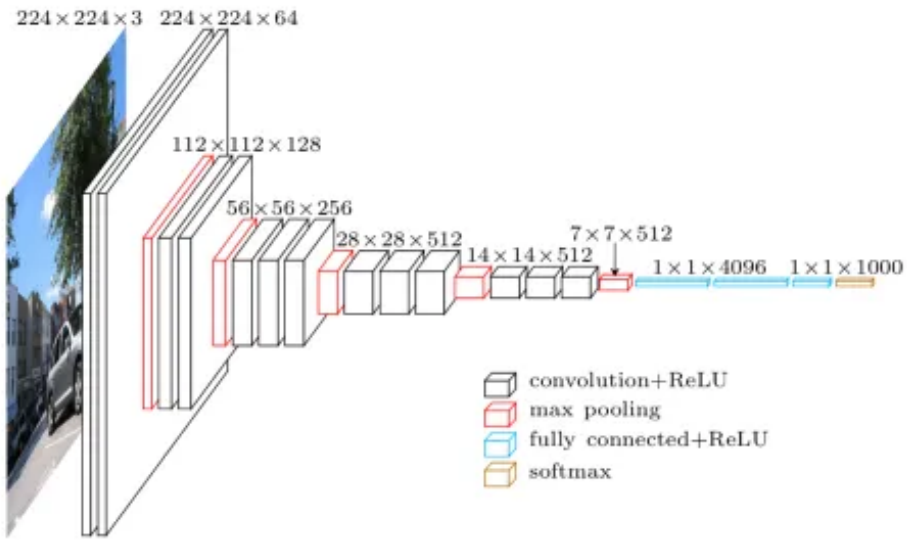


Figure 2.3: VGG16 Architecture [3].

VGG16's structure is defined by stacking convolutional layers and using small 3x3 filters, followed by max-pooling layers that downsample the spatial dimensions. This repeated pattern allows VGG16 to learn progressively complex features from the input image.

Below is a table outlining the layers and sizes of VGG16:

Layer Name	Type	Output Size	Number of Parameters
Input Image	-	224x224x3	0
Conv1-64	Conv2D	224x224x64	1,792
Conv1-64	Conv2D	224x224x64	36,928
Pool1	MaxPooling2D	112x112x64	0
Conv2-128	Conv2D	112x112x128	73,856
Conv2-128	Conv2D	112x112x128	147,584
Pool2	MaxPooling2D	56x56x128	0
Conv3-256	Conv2D	56x56x256	295,168
Conv3-256	Conv2D	56x56x256	590,080
Conv3-256	Conv2D	56x56x256	590,080
Pool3	MaxPooling2D	28x28x256	0
Conv4-512	Conv2D	28x28x512	1,180,160
Conv4-512	Conv2D	28x28x512	2,359,808
Conv4-512	Conv2D	28x28x512	2,359,808
Pool4	MaxPooling2D	14x14x512	0
Conv5-512	Conv2D	14x14x512	2,359,808
Conv5-512	Conv2D	14x14x512	2,359,808
Conv5-512	Conv2D	14x14x512	2,359,808
Pool5	MaxPooling2D	7x7x512	0
Flatten	-	25088	0
FC1	Dense	4096	102,764,544
FC2	Dense	4096	16,781,312
FC3	Dense	1000	4,097,000

Table 2.1: Detailed structure of VGG16.

The choice of VGG16, despite its computational intensity, stems from its remarkable feature extraction capabilities. Its relatively shallow architecture, when compared to more modern counterparts, allows it to strike a balance between performance and computational complexity. This makes VGG16 a preferred choice when resource constraints need to be considered, such as in edge computing scenarios or when working with limited hardware resources. VGG16's approach to deep learning serves as a cornerstone in the evolution of neural network architectures, paving the way for subsequent innovations that balance efficiency and performance.

ResNet

ResNet [11], short for Residual Networks, introduced a novel concept of skip connections or shortcut connections. These connections help in training very deep networks by easing the vanishing gradient problem. ResNet architectures have been widely adopted in various image-processing tasks due to their remarkable performance. Among the notable variants of ResNet, ResNet50 stands out as a milestone in the evolution of deep neural networks.

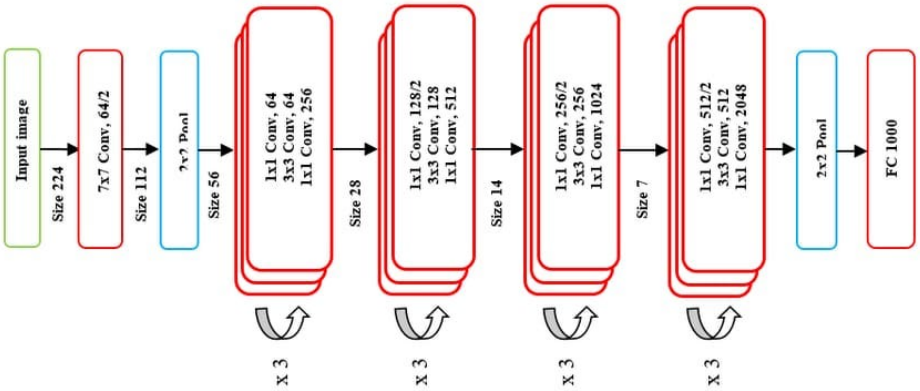


Figure 2.4: ResNet50 Architecture [4].

ResNet50 [11], as the name suggests, comprises 50 weight layers. It has demonstrated exceptional performance and efficiency in various image classification and feature extraction tasks. The structure of ResNet50 is defined by the use of residual blocks, which allow information to flow across layers more efficiently. The skip connections in each block bypass the vanishing gradient issue, making it possible to train deep networks effectively. This architecture enables ResNet50 to extract intricate features from images, enhancing its image retrieval capabilities.

Below is a table outlining the layers and sizes of ResNet50:

Layer Name	Type	Output Size	Number of Parameters
Input Image	-	224x224x3	0
Conv1	Conv2D	112x112x64	9,408
Conv2	MaxPooling2D	56x56x64	0
Conv3x	Residual Blocks	28x28x256	-
Conv4x	Residual Blocks	14x14x512	-
Conv5x	Residual Blocks	7x7x1024	-
AvgPool	GlobalAveragePooling2D	1x1x2048	0
FC	Dense	1000 (output classes)	-

Table 2.2: Detailed structure of ResNet50.

The choice of ResNet50 is motivated by its impressive performance and its capability to handle deeper architectures. Despite its depth, ResNet50 has managed to maintain computational efficiency, making it a versatile choice for image retrieval applications. Its success lies in its ability to extract meaningful features while avoiding the pitfalls of vanishing gradients. This balance of depth and efficiency makes ResNet50 a go-to choice for various computer vision tasks, particularly image retrieval, where complicated feature extraction is crucial [43].

MobileNetV2

MobileNetV2 [10], a compact network tailored with the aim for mobile devices, represents a significant leap in the field of embedded vision applications. Introduced by

Sandler, Howard, Zhu, Zhmoginov, and Chen in 2018 [10], this architecture not only achieves state-of-the-art performance across a spectrum of tasks but also ensures reasonable utilization of computational resources. Central to MobileNetV2 is the innovative 'inverted residual block' that employs depthwise separable convolutions. This block encompasses two novel attributes: linear bottlenecks and shortcut residual connections situated between these bottlenecks. Within the mechanics of the inverted residual layer, an input tensor with k channels is initially expanded into a higher-dimensional space employing pointwise 1×1 convolutions, succeeded by the Relu6 activation function. Subsequent to this, a depth-wise convolution is deployed, capturing the inherent spatial correlations. Following this is another pointwise convolution, reverting the data to its original low-dimensional tensor state. Crucially, this step is linear, safeguarding against undue information attrition. Concluding the process, a residual connection seamlessly links the input and output mappings, solidifying the network's effectiveness. Constituted of 54 layers and trained exhaustively on the ImageNet dataset, MobileNetV2 is optimized for an input image resolution of 224×224. Through its intricately efficient design, it establishes a harmonious balance between processing speed and accuracy, marking it as an optimal choice for diverse mobile and embedded vision initiatives.

Below is a table outlining the layers and sizes of MobileNetV2:

Layer Name	Type	Output Size	Number of Parameters
Input Image	-	224x224x3	0
Conv1	Conv2D	112x112x32	864
Conv2	Bottleneck Blocks	112x112x16	5,056
Conv3	Bottleneck Blocks	56x56x24	9,408
Conv4	Bottleneck Blocks	28x28x32	14,592
Conv5	Bottleneck Blocks	14x14x64	21,760
Conv6	Bottleneck Blocks	7x7x96	32,320
Conv7	Bottleneck Blocks	7x7x160	46,720
Conv8	Conv2D	7x7x320	51,520
AvgPool	GlobalAveragePooling2D	1x1x1280	0
FC	Dense	1000 (output classes)	-

Table 2.3: Detailed structure of MobileNetV2.

2.4 Video Image Retrieval

Video image retrieval poses unique challenges, given the temporal nature of videos, as they consist of a sequence of frames with valuable information contributing to the overall context. Video instance retrieval, a specific aspect of video retrieval, further highlights these challenges by requiring the localization and retrieval of specific instances within video data. Traditional image-based retrieval techniques may fall short in scenarios like video surveillance systems aimed at identifying criminals. As video data increases, the need to locate specific objects, places, or actions within videos grows more pronounced, underscoring its importance for various applications.

To address the sophistication of video instance retrieval, advanced techniques are

being developed to extract meaningful representations from video content. This involves constructing models that leverage 3D Convolutional Neural Networks (3D-CNNs) to grasp the temporal information of videos [44]. These models are tailored to learn and encode the dynamic changes and interactions over time, enabling the computation of semantic similarities between instances. The utilization of 3D-CNNs represents a significant step forward in capturing the nuances of videos beyond static frames [45]. By modeling the temporal evolution of scenes, objects, or actions, these models enhance the precision and relevance of video instance retrieval. However, due to the added dimensionality and increased computational complexity, 3D CNNs often require more computational resources compared to their 2D counterparts. Therefore, in order to achieve ultimate efficiency our proposed system will be based on 2D CNNs.

In contrast to the recent advancements focusing on real-time Content-based Image Retrieval and precise intelligent image search capabilities, the topic of video-based retrieval is relatively novel [46]. VISIONE, developed for the Video Browser Showdown 2019 challenge, introduces novel capabilities in supporting various query types, such as keyword-based search and object location search through real-time object detection algorithms [47]. Yet, these methods heavily lean on text-based indexing, potentially limiting their ability to capture complex visual nuances [48, 49]. A review of CBVIR highlights the challenges posed by excessive video frames, necessitating techniques like shot boundary detection and keyframe selection [50]. While some progress has been made in image retrieval techniques, the focus on video content retrieval remains relatively under-explored.

In the domain of content-based visual media retrieval, an adaptable framework has emerged that tackles Convolutional Neural Networks (2D CNNs), 3D Convolutions (3D CNNs), and Long short-term memory networks (LSTMs) to process both images and videos for retrieval purposes [51]. This framework employs a recurrent convolutional architecture, incorporating LSTM processing to generate a comprehensive feature representation for videos, thereby enabling effective retrieval for both images and videos [44].

Their 3D model exhibits proficiency in retrieving familiar data instances, while their 2D model excels at handling unseen data, thus showcasing complementary strengths. The incorporation of Long Short-Term Memory units within their framework contributes to enriched video comprehension and the identification of key moments. This architectural configuration takes into account both the sequence information and scene relationships present within the video, as reported [45]. However, it's important to note that the entire processing pipeline aims to clarify the entirety of a video into a single feature vector, encapsulating the video's core essence. Although this approach may trade off temporal information for efficiency gains, it underscores the multifaceted nature of designing video retrieval systems.

In the context of video image retrieval, there exists a noticeable absence of significant literature within the academic sphere. This scarcity highlights the evolving nature of this research area and the considerable scope for further investigation.

2.4.1 Keyframe Extraction

In the context of video instance retrieval, the process of identifying keyframes takes on a new dimension. These keyframes are not only representative snapshots but also hold the potential to encapsulate instances of interest. Their selection must align with the specific requirements of video instance retrieval, aiming to contain instances that carry vital semantic significance. Thus, the field of video instance retrieval seeks to bridge the gap between the intricacies of video data and the efficiency of retrieval systems, setting the stage for robust video content analysis and retrieval in complex real-world scenarios. This part is investigated by my colleague Sinian Li, further literature study is available in her thesis [52].

2.4.2 Nearest Neighbour Search

Once the feature vectors of keyframes and query images are extracted, the nearest neighbor search is performed to retrieve similar frames from a database. This step is crucial and determines the efficiency and accuracy of the retrieval process.

In our pursuit of optimizing our system with efficient searching and matching algorithms, rigorously searched a range of methods from various categories and integrated the most suitable ones into our final pipeline. Our selection criteria included the ability to seamlessly integrate with other modules, state-of-the-art performance in relevant categories, and applicability in real-world scenarios. Notably as mentioned by Yuanyuan Yao [53], ANNOY [9] emerged as a standout choice among tree-based methods such as Product Quantization [54], Product Quantization Network [55], Greedy Hash [56], and Hierarchical Navigable Small World [57] methods, consistently demonstrating competitive results in prior research. Its practical effectiveness makes it a robust selection for our purposes. We further delve into the specifics of these chosen methods in the subsequent chapters, with a focus on their application in Approximate Nearest Neighbors (ANN) search.

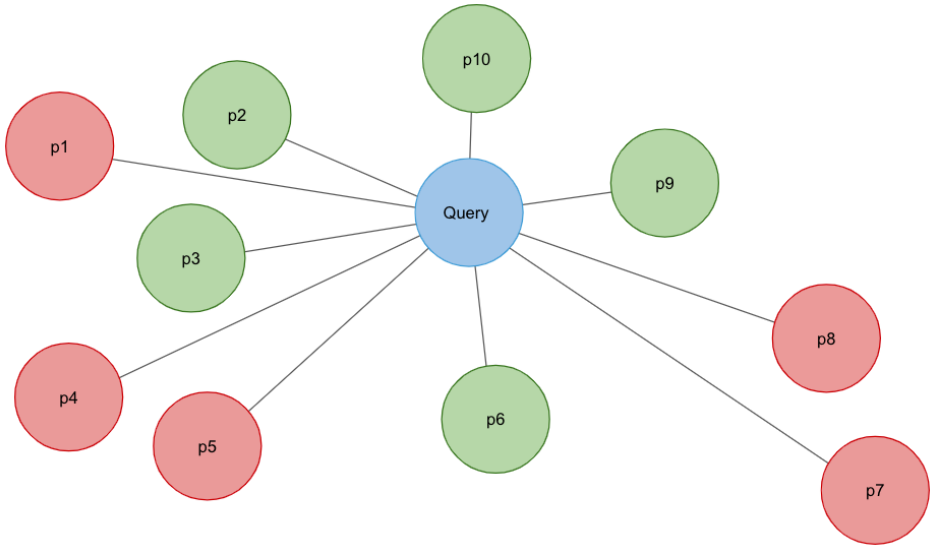


Figure 2.5: The Representation of the Nearest Neighbour Search (The Example Provides Top 5 Neighbours Closer to Query) [5].

2.5 Re-ranking in Image Retrieval

Re-ranking is a post-processing step in image retrieval, where initially retrieved results are further refined or sorted based on additional criteria or algorithms. This process helps in enhancing the precision of retrieval, especially when the initial results are sub-optimal. Various methods, ranging from spatial verification to utilizing contextual information, have been proposed for re-ranking in image retrieval.

In the context of image retrieval, it's essential to highlight the current state of re-ranking methods. These re-ranking approaches can be broadly categorized into two fundamental types: local feature-based methods and global feature-based methods. Local feature-based re-ranking, often distinguished as geometric re-ranking or spatial verification methods by esteemed researchers [58, 59, 60, 61], revolves around harnessing the geometric properties of image features to refine search results. In contrast, global feature-based re-ranking, commonly labeled as non-geometric re-ranking methods [61], is centered on exploiting the overall characteristics and semantic information embedded within image features to optimize retrieval outcomes.

These dual facets of re-ranking techniques form the keystone of modern image retrieval, and understanding their complications is crucial for navigating the dynamic field of information retrieval.

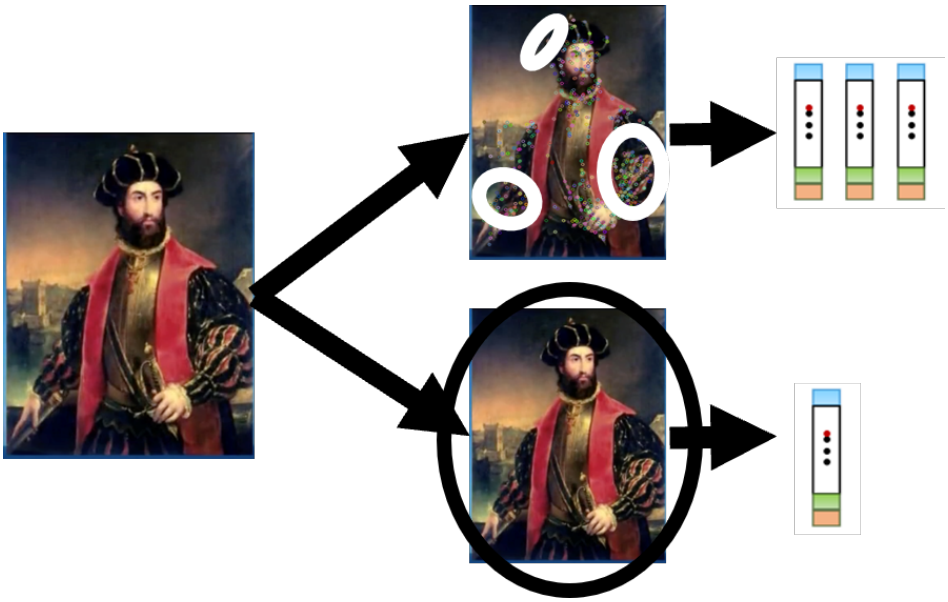


Figure 2.6: The Representation of the Local Keypoints (Top) and Global Descriptors (Bottom).

3 Methodology

In the domain of content-based image retrieval, the efficient retrieval of relevant images from a vast database remains a significant challenge. This work proposes a novel CBIR on video image system that effectively combines the power of deep learning feature extraction with the speed of the Approximate Nearest Neighbors Oh Yeah (ANNOY) [9] nearest neighbor algorithm. The system employs a two-phase approach, strategically divided into an offline phase for the pre-processing and preparation stage and an online phase for query processing. The proposed phases are described as follows:

- **Offline Phase:** This is a preparation stage consisting of KFE, MobileNetV2 feature extraction of keyframes, and building ANNOY index tree for MobileNetV2 features.

- **Online Phase:** This stage requires the user to input query images. Both MobileNetV2 and ResNet101 + SOLAR features are extracted for query images. Next, the pre-built ANNOY index for MobileNetV2 features is then utilized to search and match the most similar keyframes. These keyframes are then shortlisted for the re-ranking stage. The ResNet101 + SOLAR features of the shortlisted keyframes are extracted, and finally, re-ranking is applied through linear search. ResNet101 + SOLAR features are stored for each run, enabling faster performance progressively as more input queries are inserted into the system by the user.

In the next sections, each section of the system will be explained in detail and the offline phase and online phase will be described in per module of our proposed system.

3.1 Keyframe Extraction

The offline phase includes KFE, to prepare the significant frames (keyframes) among the videos to form the gallery of the CBIR. The thesis covers only the CBIR part of the proposed system. For keyframe extraction analysis Sinian Li provided her thesis research on various methods of KFE using deep-learning and traditional methods [52]. The most promising method utilizes the traditional algorithm of the color-based method with an improvement to the algorithm by constructing sub-matrices within the feature matrix and reducing the dimension by singular value decomposition to accelerate the process of keyframe extraction.

3.2 Feature Extraction

In our proposed system, we employ two distinct deep-learning architectures for descriptor extraction for images: MobileNetV2 and ResNet101 + SOLAR. The choice of these architectures is driven by the specific needs of our CBIR system.

MobileNetV2, renowned for its compact structure, serves as an excellent choice for our system. Its design prioritizes computational efficiency, making it an ideal candidate for feature extraction. This computational efficiency translates into faster processing times and reduced resource requirements, ensuring that our system remains responsive and cost-effective [62].

In contrast, we have also incorporated ResNet101 + SOLAR features into our system. ResNet101 is a deep convolutional neural network known for its exceptional accuracy in image recognition tasks [63]. When combined with second-order loss and attention, this architecture produces wider and more precise feature vectors. By leveraging ResNet101 + SOLAR, our system excels in delivering accurate and contextually relevant search results [6].

In the following subsections, we will delve deeper into the feature extraction methods for both MobileNetV2 and ResNet101 + SOLAR. This comprehensive explanation will shed light on how these architectures contribute to the effectiveness of our CBIR system.

3.2.1 Compact Deep Learning Network: MobileNetV2 Features

- **Offline:** Extraction of MobileNetV2 features from keyframes. These features will be used to build ANNOY indexes with MobileNetV2 features.
- **Online:** Extraction of MobileNetV2 features from query images.

Detailed Mechanics of MobileNetV2:

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	32	1	2
$112^2 \times 32$	bottleneck	1	16	1	1
$112^2 \times 16$	bottleneck	6	24	2	2
$56^2 \times 24$	bottleneck	6	32	3	2
$28^2 \times 32$	bottleneck	6	64	4	2
$14^2 \times 64$	bottleneck	6	96	3	1
$14^2 \times 96$	bottleneck	6	160	3	2
$7^2 \times 160$	bottleneck	6	320	1	1
$7^2 \times 320$	conv2d 1x1	-	1280	1	1
$7^2 \times 1280$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 1280$	conv2d 1x1	-	k	-	

Table 3.1: Detailed structure of MobileNetV2 [10].

- t: Expansion factor, determining the number of output channels in a bottleneck block compared to the number of input channels.

- c: Number of output channels produced by a specific layer or block in the network.
- n: Number of times a particular type of block is repeated within the architecture.
- s: Stride used in convolutional layers or blocks, determining the spatial downsampling factor of the output compared to the input.

Standard Convolutional Layer

The initial phase of the MobileNetV2 architecture involves processing the input image X through a standard convolutional layer. This layer employs a set of 32 filters with learnable weights and biases W_{conv} to perform convolutions on the input data. Importantly, this convolutional layer operates with a stride of 2 (stride = 2), which results in a down-sampling effect. Mathematically, the output C of this convolutional layer can be represented as:

$$C = \text{Conv2D}(X, W_{conv}, \text{stride} = 2)$$

This operation efficiently reduces the spatial dimensions of the data while simultaneously increasing the number of feature channels. The transformed output serves as a critical starting point for the subsequent feature extraction stage.

Bottleneck Residual Blocks

Following the initial convolutional layer, the architecture integrates a sequence of 'bottleneck' residual blocks. These blocks represent the balanced mixture of computational efficiency and feature representation richness. Each bottleneck block comprises three fundamental layers:

1. **1x1 Convolution (Pointwise Convolution):** This layer strategically introduces a 1x1 convolution, often denoted as 'pointwise convolution'. The main goal of this layer is to expand the number of channels, thereby enhancing the capacity of the model to represent complex visual patterns and transformations. Mathematically, the output P of this layer can be expressed as:

$$P = \text{Conv2D}(C, W_{pw})$$

Here, W_{pw} represents the pointwise convolution weights.

2. **3x3 Depthwise Convolution:** The subsequent layer entails a 3x3 depthwise convolution, a distinctive operation that applies a single convolutional filter per input channel. This 'depthwise' convolution is followed by batch normalization and the Rectified Linear Unit (ReLU) activation function. This combination enhances the model's capability to learn intricate patterns and features. Mathematically, the output D of the depthwise convolution can be represented as:

$$D = \text{DepthwiseConv2D}(P, W_{dw}, \text{stride} = 1)$$

Here, W_{dw} represents the depth-wise convolution weights.

3. **1x1 Convolution (Pointwise Convolution):** The final layer within each bottleneck block is another 1x1 convolution. This operation projects the channel dimensions into a more compact space, effectively reducing computational load and conserving memory. Mathematically, the output B of this convolution can be defined as:

$$B = \text{Conv2D}(D, W_{pw2})$$

Here, W_{pw2} represents the point-wise convolution weights.

The orchestration of these layers in each bottleneck block leads to the extraction of multi-faceted, nuanced features, all while maintaining computational efficiency.

Average Pooling Layer

After spanning the sequence of bottleneck residual blocks, the model's output B undergoes a pivotal transformation via the average pooling layer. This layer plays an instrumental role in further diminishing the spatial dimensions of the data while retaining essential features. The average pooling operation calculates the mean value within predefined windows of the feature maps, resulting in a profound reduction in the number of spatial dimensions. Mathematically, the output $AvgP$ of the average pooling layer can be defined as:

$$AvgP = \text{AveragePooling2D}(B, \text{pool\_size})$$

Here, pool_size signifies the dimensions of the pooling windows used for aggregation. The completion of the average pooling process highlights the model's focus on essential features while mitigating the influence of irrelevant spatial information.

Fully Connected Layer

In the final phase of the MobileNetV2 architecture, the output F that has undergone average pooling is channeled into a fully connected (dense) layer. This layer holds immense significance for the classification task, as it diligently computes probabilities corresponding to the target classes. Each neuron in this dense layer corresponds to a specific class, illustrating a set of weights and a bias term. The output of the fully connected layer FC is obtained through a linear combination of the transformed features F and the weights W_{fc}, followed by the addition of the bias term b_{fc}:

$$FC = F \times W_{fc} + b_{fc}$$

Afterward, the softmax activation function is applied to the output FC, yielding the final probabilities that signify the model's prediction for each class. This operation is the final step in MobileNetV2's feature extraction and transformation journey, enabling it to make precise and informed class predictions.

The process of feature extraction within the MobileNetV2 architecture culminates in the creation of representations known as feature maps, manifested in the form of 4-dimensional (4D) tensors. These tensors encapsulate crucial visual information about

the input images and serve as the foundation for subsequent analysis and processing. The specific structure and properties of these tensors hold valuable insights into the characteristics of the extracted features.

The feature maps are selected at a particular layer within the MobileNetV2 architecture, specifically the penultimate layer before the final classification layer. This choice of layer is strategic, as it ensures that the feature vectors capture a high-level abstraction of the input image, making them suitable for a wide range of tasks.

The shape of these 4D tensors can be denoted as $(\text{batch\_size}, 1280, 1, 1)$. This representation encompasses several dimensions, each of which contributes to the overall understanding of the features extracted:

- **batch_size:** This dimension refers to the number of images that are processed simultaneously within a single batch. Each batch may contain a variable number of images, and this parameter influences the efficiency of computation during training and inference.

- **1280:** Within the MobileNetV2 architecture, this value holds significant importance. It signifies the number of filters present in the final convolutional layer of the model. These filters play a pivotal role in detecting and emphasizing various visual patterns and attributes within the input images.

- **1:** The dimensions of height and width for each feature map are both represented by the value 1. This signifies that each feature map retains a condensed, one-dimensional representation of the visual features extracted from the original image.

- **1:** Similarly, the second instance of the value 1 underscores that the feature map maintains a single-dimensional representation of the visual elements. This emphasizes the focus of the model on capturing and conveying essential features in a brief manner.

Each image within a batch is characterized by a distinct 1280-dimensional vector, which encapsulates the intricate features extracted by the MobileNetV2 architecture. These vectors serve as robust descriptors, encapsulating distinctive aspects of the visual content present in each image. Due to their compact comprehensive nature, these descriptors are well-suited for subsequent processing steps, enabling efficient and effective handling of a wide array of content within the image video database.

3.2.2 Larger Deep Learning Network: ResNet101 + SOLAR Features

- **Online:** Extraction of ResNet101 + SOLAR features from both query images and shortlisted keyframes.

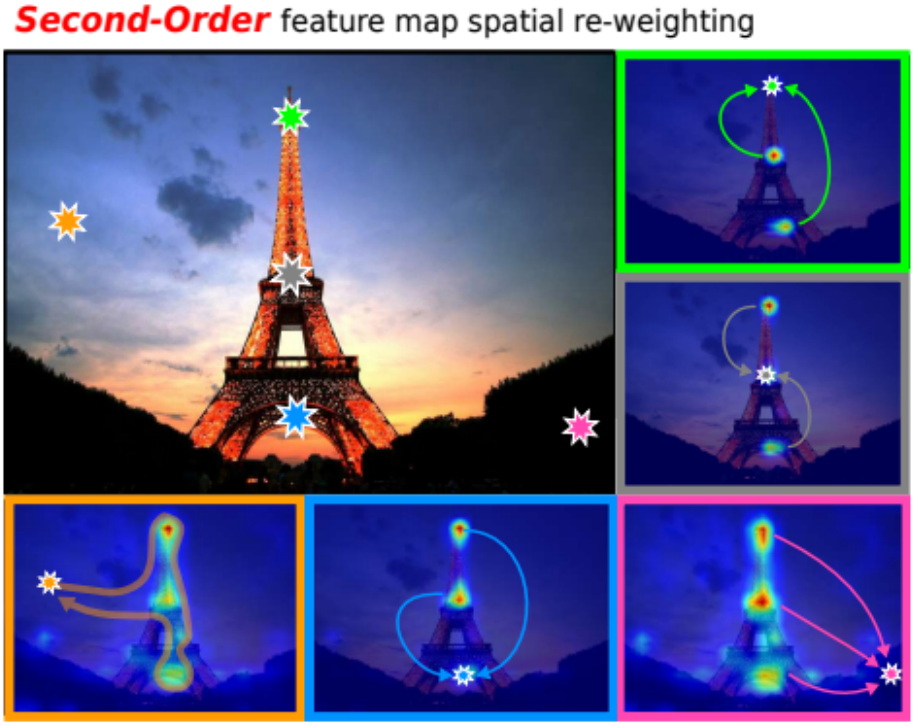


Figure 3.1: An illustration of the SOLAR descriptor representing second-order spatial relations. The feature maps are re-weighted to provide a better global representation of the image [6].

ResNet101 + SOLAR is a combination of the ResNet101 [11] architecture and the SOLAR [6] global descriptor. ResNet101 is a deep convolutional neural network that is widely used for image classification tasks. SOLAR, on the other hand, is a global descriptor that leverages second-order information through spatial attention and descriptor similarity to improve large-scale image retrieval. By combining ResNet101 as a CNN backbone with SOLAR, the performance of image retrieval can be enhanced, achieving state-of-the-art results in terms of mean average precision and top-k precision [6].

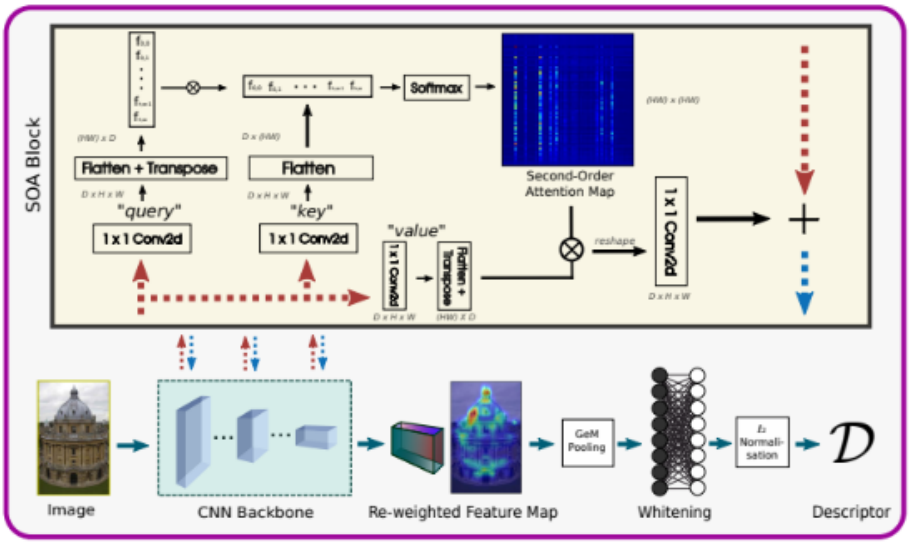


Figure 3.2: The Pipeline of SOLAR Global Features. Incorporating several Second-Order Attention (SOA) blocks at various stages of the ResNet101 backbone, along with subsequent steps involving GeM pooling, whitening, and L2 normalization, is performed [6].

The pipeline of ResNet101 + SOLAR involves utilizing the ResNet101 architecture as the backbone network for feature extraction. The SOLAR method is then applied to enhance the global descriptor representation for large-scale image retrieval. SOLAR incorporates second-order information through both spatial attention (SOA) and descriptor similarity (SOS) to re-weight feature maps and produce better representations for retrieval. The ResNet101 + SOLAR pipeline includes fine-tuning the SOAs and the whitening layer using the Google Landmark 18 [64] dataset, training with the triplet loss, and utilizing SOSNet for local descriptor learning. The pipeline aims to improve image retrieval performance by leveraging second-order attention and similarity information.

CNN Backbone: ResNet101

ResNet101, short for Residual Network with 101 layers, represents a pioneering architecture in the domain of deep convolutional neural networks. Introduced as an evolution of the original ResNet, it is renowned for its unprecedented depth, comprising a staggering 101 convolutional layers [11]. The fundamental innovation behind ResNet101 is the introduction of residual connections, also known as skip connections or shortcut connections [65, 66, 67]. These connections enable the network to bypass certain layers during forward propagation, allowing the model to learn residual mappings. In essence, ResNet101 mitigates the vanishing gradient problem encountered in very deep networks and facilitates the training of exceptionally deep architectures. The network is organized into a series of residual blocks, each containing multiple convolutional layers and a shortcut connection that skips one or more blocks. This architecture has revealed remarkable performance across a wide array of computer vision tasks, from image classification to object detection and semantic segmentation [11].

The architecture of ResNet101 displays a hierarchical structure, beginning with a standard convolutional layer that captures low-level image features. Subsequently, a series of residual blocks are stacked together, and these blocks become increasingly complex as they progress through the network. The depth of ResNet101 enables it to learn intricate and abstract features, making it particularly suited for tasks requiring fine-grained image analysis. Within each residual block, convolutional layers are interleaved with batch normalization and rectified linear unit (ReLU) activation functions. Additionally, bottleneck architectures are employed within the blocks, consisting of 1x1, 3x3, and 1x1 convolutions, further enhancing the model's representational power while minimizing computational overhead. ResNet101 is capped with a global average pooling layer, followed by fully connected layers for classification tasks. The unique combination of depth, residual connections, and bottleneck structures make ResNet101 a cornerstone architecture in modern deep learning, facilitating the development of highly accurate and robust computer vision models [11].

Detailed Mechanics of ResNet101:

Input	Operator	t	c	n	s
$224^2 \times 3$	conv2d	-	64	1	2
$112^2 \times 64$	m axpool	-	-	-	3
$56^2 \times 64$	Residual Block 1 (x3)				
$56^2 \times 256$	Residual Block 2 (x4)				
$28^2 \times 512$	Residual Block 3 (x23)				
$14^2 \times 1024$	Residual Block 4 (x3)				
$7^2 \times 2048$	avgpool 7x7	-	-	1	-
$1 \times 1 \times 2048$	fc	-	k	-	-

Table 3.2: Detailed structure of ResNet101 [11].

Second Order Attention (SOA) Layer

Figure 3.2 illustrates the structure of the SOA Layer, a critical component of the architecture. Second-order attention is crucial in image retrieval tasks because it allows for the re-weighting of feature maps, emphasizing salient image locations. This re-weighting helps the network learn the relative contributions of various features into the final descriptor. By focusing on distinctive regions within landmarks and connecting structures within patches, second-order attention improves the performance of image descriptors and enhances the network's ability to learn from different features. The consistent attention maps generated by second-order attention provide qualitative evidence of its effectiveness in assisting the network in learning and improving image retrieval results [6].

From an input image $I \in \mathbb{R}^{H \times W \times 3}$ processed through a Fully-Convolutional Network denoted by θ, we obtain a feature map $f = \theta(I) \in \mathbb{R}^{h \times w \times d}$ where h, w, and d are the height, width, and feature dimensionality, respectively.

Generalized Mean (GeM) Pooling

After obtaining the feature maps, the global descriptor of the input image I is generated by the GeM pooling [68] operation. which takes the weighted average of the feature map according to the absolute magnitude of each feature. At first, the feature map is consolidated into the global descriptor vector $d = \mathrm{GeM}(f, p)$, where p denotes the pooling parameter.

The GeM pooling operation is mathematically defined as:

$$\mathrm{GeM}(\mathbf{f}, p) = \left(\frac{1}{N} \sum_{i=0}^{N} f_i^p \right)^{\frac{1}{p}}, \tag{3.1}$$

plays a crucial role in shaping the global descriptor. In the following, we briefly explain the idea behind GeM.

The Generalized Mean Pooling technique [68], strikes a balance between the characteristics of maximum and average pooling methods. This equilibrium is achieved through the incorporation of a learnable pooling parameter p. The versatility of GeM arises from its ability to adapt to various fine-grained regions within an image by adjusting the value of p. When applied to an input feature map X, GeM generates a single D-dimensional feature vector denoted as f.

To shed light on its behavior, GeM can be likened to maximum pooling when p takes on values approaching infinity (e.g., $p \to \infty$), and it resembles average pooling when p equals 1 (i.e., $p = 1$). This characteristic makes GeM a flexible pooling method capable of capturing different aspects of information within an image, depending on the chosen value of the pooling parameter p. This vector in the case of the traditional global max pooling [69] is presented by

$$\mathbf{f}^{(m)} = \left[\mathrm{f}_1^{(m)} \dots \mathrm{f}_k^{(m)} \dots \mathrm{f}_K^{(m)} \right]^\top, \quad \mathrm{f}_k^{(m)} = \max_{x \in \mathcal{X}_k} x,$$

while for average pooling [70] the vector is represented by

$$\mathbf{f}^{(a)} = \left[\mathrm{f}_1^{(a)} \dots \mathrm{f}_k^{(a)} \dots \mathrm{f}_K^{(a)} \right]^\top, \quad \mathrm{f}_k^{(a)} = \frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x.$$

The generalized mean pooling is represented by

$$\mathbf{f}^{(g)} = \left[\mathrm{f}_1^{(g)} \dots \mathrm{f}_k^{(g)} \dots \mathrm{f}_K^{(g)} \right]^\top, \quad \mathrm{f}_k^{(g)} = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{pk} \right)^{\frac{1}{p_k}}.$$

The pooling parameter p_k can be manually set or learned through the back-propagation algorithm, as it is differentiable. The corresponding derivatives are provided.

$$\frac{\partial \mathrm{f}_k}{\partial x_i} = \frac{1}{|\mathcal{X}_k|} \mathrm{f}_k^{1-p_k} x_i^{p_k - 1},$$

$$\frac{\partial \mathrm{f}_k}{\partial p_k} = \frac{\mathrm{f}_k}{p_k^2} \left(\log \frac{|\mathcal{X}_k|}{\sum_{x \in \mathcal{X}_k} x^{p_k}} + p_k \frac{\sum_{x \in \mathcal{X}_k} x^{p_k} \log x}{\sum_{x \in \mathcal{X}_k} x^{p_k}} \right).$$

The pooling parameter, p, can be adjusted for each local contribution of feature map F to the global descriptor d based on its associated feature activation. The GeM [68] method is considered a first-order measure and assumes that each location on the map is independent, disregarding the influence of each spatial characteristic in relation to others. However, the global descriptor d has a finite receptive region, resulting in limited contribution from each local feature. To incorporate spatial information into feature pooling, the second-order attention (SOA) layer [6] is a useful method. Figure 3.3 illustrates the structure of the SOA layer.

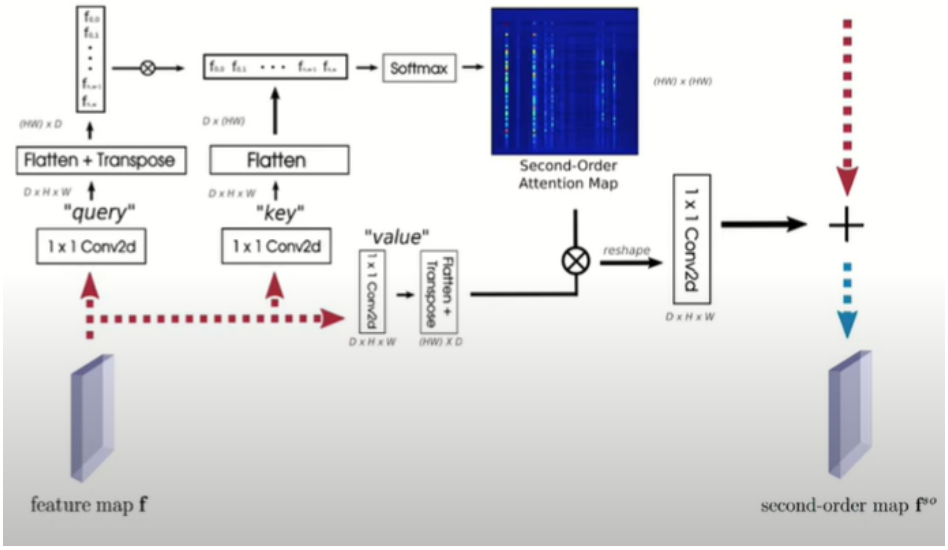


Figure 3.3: The pipeline of SOA layer [7].

To begin, two projections from the feature map f are created, corresponding to the query q and the key k. Subsequently, both the query q and the key k are reshaped and converted into a matrix with dimensions $D \times HW$. The subsequent step involves the computation of the second-order attention map, denoted as z, and can be expressed as follows:

$$\mathbf{z} = \text{softmax}\left(\alpha \cdot \mathbf{q}^{\top}\mathbf{k}\right).$$

Here, α serves as the scaling factor and z takes on the shape of $HW \times HW$, enabling the feature map f to combine all the local features across the entire map. Subsequently, as same as of q and k, a third projection value head v for f is generated with dimensions $HW \times D$. As a final step, the second-order attentional feature map f^{so} is derived from the first-order attentional feature map. An additional 1×1 convolution, denoted as ψ, is employed to supervise the influence of the attention. Within the feature map f^{so}, a novel feature $f^{so}_{i,j}$ is redefined as a function of features from all positions within f, thus making it interpretable as a function encompassing the entirety of the input image:

$$\mathbf{f}^{so} \equiv \mathbf{f} + \psi(\mathbf{z} \times \mathbf{v}),$$
$$f^{so}_{i,j} = g\left(\mathbf{z}_{ij} \odot \mathbf{f}\right).$$

After the conclusion of the last second-order attention layer, the recently-formed feature map f^{so} will be employed as the input for the subsequent pooling layer, facilitating further feature enhancement and pooling steps [6]. This culminates in the generation of final global features through GeM pooling, which encapsulates a richer set of local information:

$$\mathrm{GeM}\left(\mathbf{f}^{so}p\right) = \left(\frac{1}{N}\sum_{i=0}^{N} f_i^{so^p}\right)^{\frac{1}{p}}.$$

Whitening

In this section, we delve into the post-processing of feature vectors. Following the pooling aggregation, it becomes critical to apply whitening to the feature descriptors. Whitening, a crucial step in image retrieval, plays a pivotal role in handling high-dimensional features [71]. These features carry an abundance of information, and whitening serves the purpose of eliminating inter-feature correlations through linear transformations. Its importance lies in not only reducing the dimensionality of feature vectors but also in eliminating redundant information. This significance is particularly pronounced when dealing with CNN-based descriptors [72, 70].

3.2.2.1 Discriminative Learned Whitening

The discriminative learned whitening [68] approach makes use of labeled data provided by 3D models and linear discriminant projections, as initially introduced by Mikolajczyk [33]. This projection consists of two components: whitening and rotation. The whitening coefficient is calculated as the inverse of the square root of the covariance matrix of matching pairs, denoted as $C_S^{-\frac{1}{2}}$.

Here, C_S is computed as:

$$C_S = \sum_{Y(i,j)=1} (\overline{\mathbf{f}}(i) - \overline{\mathbf{f}}(j))(\overline{\mathbf{f}}(i) - \overline{\mathbf{f}}(j))^\top.$$

The rotation component involves Principal Component Analysis (PCA) applied to the covariance matrix of non-matching pairs in the whitened space, denoted as eig $\left(C_S^{-\frac{1}{2}} C_D C_S^{-\frac{1}{2}}\right)$, where C_D is calculated as:

$$C_D = \sum_{Y(i,j)=0} (\overline{\mathbf{f}}(i) - \overline{\mathbf{f}}(j))(\overline{\mathbf{f}}(i) - \overline{\mathbf{f}}(j))^\top.$$

The projection can be expressed as:

$$P = C_S^{-\frac{1}{2}} \,\mathrm{eig}\left(C_S^{-\frac{1}{2}} C_D C_S^{-\frac{1}{2}}\right).$$

where $P^{\top\top}(\overline{\mathbf{f}}(i) - \mu)$, and μ represents the pooled feature vector.

To limit the dimensionality of the descriptors to D, only the eigenvectors corresponding to the D largest eigenvalues are utilized. Subsequently, $l2$ normalization is applied to the projected vectors.

This whitening method leverages all available training pairs while focusing on the discriminative information between matched and unmatched pairs, ultimately achieving superior performance compared to the commonly used PCA whitening.

3.2.2.2 End-to-End Whitening

The end-to-end whitening method [73] incorporates a whitening layer at the network's end. This whitening is optimized end-to-end during fine-tuning, using the same training data in batch mode and convolutional filters.

The whitening layer is directly modeled from a fully connected layer and encompasses both the projection and rotation discussed earlier. In essence, discriminative learning whitening is integrated into a layer responsible for whitening the training batch directly. This approach requires a significant amount of training data. While it offers high integration and ease of application to multiple datasets, it has the drawbacks of being computationally intensive and slow to converge.

In terms of final mAP performance, both discriminative learned whitening and end-to-end whitening are comparable, and they both significantly outperform PCA whitening [68].

L2 Normalization

After obtaining the feature maps, the next step is to perform L2 normalization on them. L2 normalization, also known as Euclidean normalization, is a common technique in deep learning to scale feature vectors to have a unit norm. It ensures that all feature vectors have the same scale, which can be important for many machine-learning algorithms.

The L2 normalization is applied element-wise to each feature vector $\mathbf{v}$ of dimension D in the feature maps:

$$\mathbf{v}_{\text{normalized}} = \frac{\mathbf{v}}{\|\mathbf{v}\|_2}, \tag{3.2}$$

where $\|\mathbf{v}\|_2$ represents the L2 norm of the feature vector $\mathbf{v}$. This operation scales each feature vector so that its Euclidean (L2) norm becomes equal to 1.

L2 normalization is an important step in many deep learning applications, as it helps assemble the model's training more stable and ensures that features with different scales do not dominate the learning process [74].

The ResNet101 + SOLAR system combines the power of the ResNet101 deep convolutional neural network for feature extraction with the advanced capabilities of the SOLAR system for enhancing feature representations. The feature extraction process begins with ResNet101, which takes as input an image and produces a feature map with dimensions of $1 \times 1 \times 2048$. Subsequently, the SOLAR system further processes this feature map. Finally, the system applies GeM pooling to generate global features,

resulting in a feature vector whose size is typically $1 \times 1 \times D$, where D represents the feature dimensionality determined by the specifics of the SOLAR processing and pooling operations following by whitening and L2 normalization steps. The output descriptor produced by this comprehensive pipeline is typically a feature vector with dimensions of $1 \times 1 \times 2048$, representing an informative global feature representation of the input image. This combined approach leverages deep learning and attention mechanisms to create discriminative feature representations for our retrieval task.

3.3 Search and Match Algorithms

Following the acquisition of feature vectors from both query images and keyframes, the subsequent step involves employing search and matching algorithms for video image retrieval. This section will delve into a detailed exploration of these search and matching algorithms.

3.3.1 ANNOY: Approximate Nearest Neighbors with Tree-Based Methods

Tree-based methods for approximate nearest neighbor search are known for their effectiveness in partitioning high-dimensional spaces efficiently. A notable example is the ANNOY library [9], which stands out by its randomized approach to space partitioning. In contrast to structured partitioning methods like kd-trees, the partitioning scheme of ANNOY is driven by randomness rather than strict mathematical criteria [9].

The split of the algorithm for each phase is described as follows:

- **Offline:** Build an ANNOY index for MobileNetV2 features using the extracted features from the gallery images.
- **Online:** Search and match relevant keyframes to the query using a pre-build ANNOY tree with MobileNetV2 features.

In the ANNOY framework, the construction of a tree involves the following steps:

1. **Random Hyperplane Selection:** Two data points are arbitrarily selected, and a hyperplane centered from these points is used to partition the space into two subspaces.
2. **Node Creation:** The random split is saved as a node, and two branches are created. Data points on the left subspace are assigned to the left branch, while those in the right subspace go to the right branch. This process is repeated recursively for each subspace, incrementing the depth, until a specified condition, such as having at most K items in each area created by the splitting hyperplanes, is met.

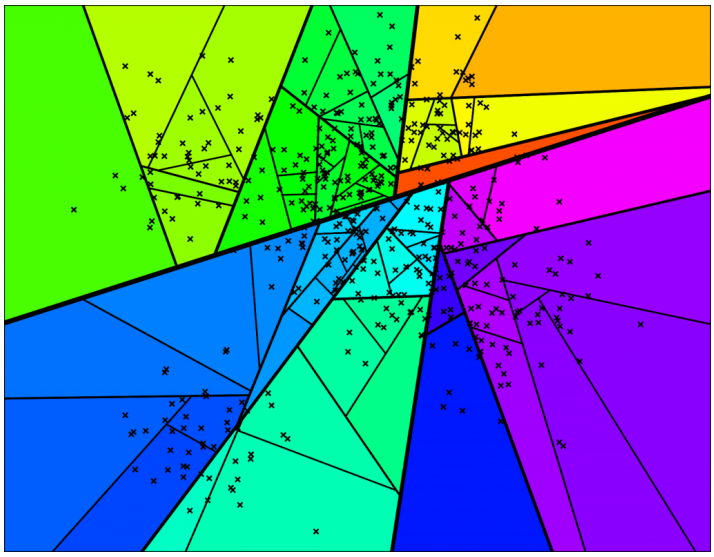


Figure 3.4: Configuration of the partitioning between hyperplanes [8].

An illustration of this tree-building process is depicted in Figure 3.5, where squares represent intermediate nodes holding splitting hyperplanes, and circles denote data points within subspaces.

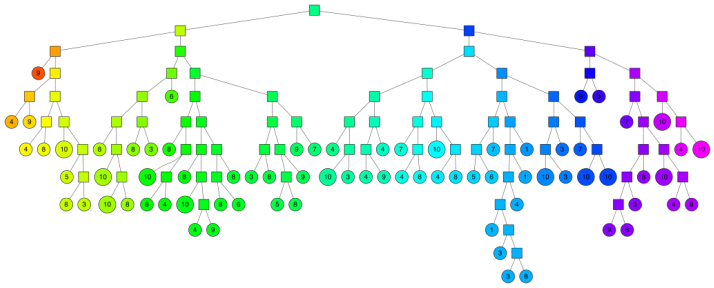


Figure 3.5: Building a binary tree in ANNOY [9].

To find the nearest neighbor of a given query point, ANNOY employs a search mechanism that traverses the tree. The search starts at the root, following the same rule used during tree construction: moving to the left or right branch depending on whether the query lies on the left or right side of the splitting plane. The search terminates upon reaching a leaf node, whose enclosed data points are considered the nearest neighbors. Figure 3.7 provides an example of this search process, with the red cross indicating the query and the colored route representing the steps taken to find the relevant data points.

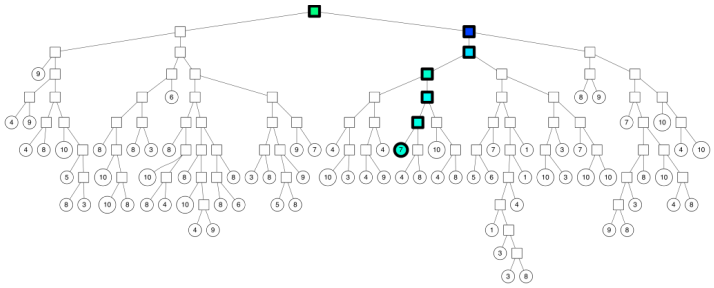


Figure 3.6: Searching with the binary tree in ANNOY [9].

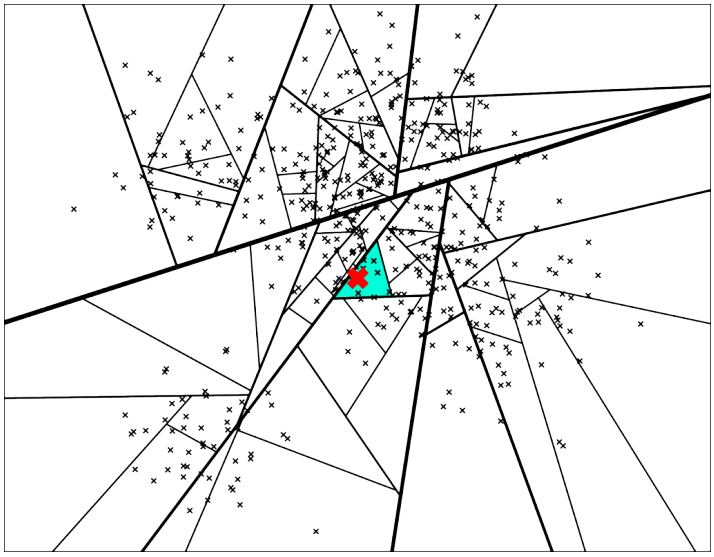


Figure 3.7: Configuration of the search process on hyperplanes [8].

However, two challenges arise in this process. First, some neighbors closer to the query may be overlooked as they fall into adjacent subspaces. Second, the number of data points within the shaded region may not meet the desired criteria. To address these issues, ANNOY incorporates a priority queue mechanism. It builds a forest of trees, aggregates the results from all trees into a list, and sorts them based on their distances to the query. Additionally, ANNOY provides an option to retain the wrong side of the tree when the points in that branch are not farther away than a predetermined threshold.

ANNOY is versatile and is effective for vectors of dimensions up to 1000, despite being initially designed for vectors with fewer than 100 dimensions. It offers a search complexity close to $O(\log N)$, making it suitable for various applications. However, it's important to note that ANNOY lacks theoretical guarantees on its search performance.

In practice, increasing the number of trees is often a recommended strategy, provided memory constraints are not an issue. ANNOY's randomized approach to tree-based partitioning, combined with these search optimizations, makes it a valuable tool for approximate nearest neighbor search in high-dimensional spaces.

3.3.2 Linear Search

Linear comparison, also known as brute-force comparison [75], represents the simplest and most direct method for comparing feature vectors. In this approach, each query vector is systematically compared with every keyframe vector within the dataset. As the term implies, the computational complexity of linear search grows linearly in proportion to the size of the data, resulting in a time complexity of $O(n)$, where 'n' represents the total number of keyframes in the dataset.

The linear search algorithm is available only in the online phase.

- **Online:** Search and match relevant keyframes to the query using linear search with ResNet101 + SOLAR features of shortlisted keyframes.

The primary advantage of employing linear search lies in its straightforwardness and its lack of reliance on data partitioning or complex data structures. In contrast to tree-based or indexed approaches, linear search directly conducts pairwise comparisons between query vectors and keyframe vectors. Consequently, the total time required for linear search depends solely on the duration of individual vector comparisons.

Mathematically, the time complexity (T_{linear}) of linear search can be expressed as:

$$T_{\text{linear}} = O(n)$$

Here: T_{linear} signifies the time complexity of linear search and 'n' denotes the number of keyframes in the dataset.

While linear search is a conceptually simple method, its applicability to large datasets may be limited due to its linear time complexity. As the dataset size increases, the duration of comparisons grows proportionally, potentially resulting in impractical execution times for extensive collections of keyframes [53].

3.4 Storage and Optimization Strategies

Storage and optimization are crucial aspects of our proposed efficient image retrieval system. The main advantage of this proposed method is directly related to efficient feature extraction results in energy saving with a memory-friendly structure.

- **Efficient:** Employing the compact MobileNetV2 network for feature extraction accelerates the offline phase's feature extraction process significantly. This swiftness is particularly advantageous in scenarios requiring instant searches, such as forensic applications where rapidly creating galleries from video frames is essential.
- **Energy Cost:** Leveraging the compact MobileNetV2 network for feature extraction results in a faster process, translating into energy savings. Since the offline stage typically consumes a substantial amount of energy and computation time, this optimization directly addresses a key bottleneck.
- **Memory Friendly:** The utilization of the compact MobileNetV2 network yields feature vectors that are half the length of those obtained using ResNet + SOLAR. Consequently, in cases involving extensive video databases, employing shortlisting

techniques for similar frames enables the avoidance of storing large descriptions of irrelevant keyframes.

The online phase of our system progressively accelerates as users submit more queries. This phase is intelligently designed to store ResNet101 + SOLAR features of keyframes when they match input query images. For consecutive runs with similar query images, the online phase eliminates the need for redundant feature extraction from ResNet101 + SOLAR, substantially reducing processing requirements. As a result, re-ranking only necessitates the loading of features from ResNet101 + SOLAR, streamlining the online retrieval process.

3.4.1 Shortlisting Keyframes

The selection of an appropriate threshold for shortlisting keyframes depends on the dataset characteristics and user priorities. Our system inherently involves a trade-off between efficiency and accuracy.

For scenarios necessitating rapid retrieval with acceptable accuracy, a lower threshold, such as shortlisting 100 keyframes per query image, suffices. This approach effectively filters out challenging matches, ensuring prompt results.

In contrast, when precision is vital, selecting a higher threshold is advisable, with the expense of increased computation time and a slightly slower search process. This fine-tuning of the threshold allows users to strike the desired balance between speed and accuracy in the retrieval process.

4 Experiments and Results

In order to comprehensively evaluate the effectiveness of our proposed Content-Based Video Image Retrieval (CBVIR) system, a series of three distinct tests were conducted, each shedding light on different aspects of its performance and capabilities. These tests were executed using the Condor server of TU Delft equipped with the following setup: **CPU:** 64 processors, 32 cores, AMD Ryzen Threadripper PRO 3975WX CPU @ 2.20GHz, 125 GiB RAM, **GPU:** 1x NVIDIA RTX A6000 48 GiB RAM.

Tests 1: Filtering Algorithm

The initial test focused on benchmarking our filtering algorithm against various compact network features and well-established classical image processing algorithms. The primary metrics employed in Test 1 were the mean Average Precision (mAP) and the computation time required for feature extraction. This test aimed to establish the system's ability to effectively filter irrelevant frames and identify relevant candidates within the gallery images. By comparing mAP scores and computation times, we gained insights into the trade-offs between feature extraction efficiency and retrieval accuracy. The features extraction algorithm for filtering is implemented using the best-fitted algorithm on the results of these tests.

Test 2: Computation Time

The second test revolved around the computation time involved in different stages of our proposed CBIR system. We meticulously measured the time taken by each component, from feature extraction to re-ranking, and contrasted these timings with those of state-of-the-art CBIR systems. This test provided a comprehensive understanding of the computational demands of our system and identified potential areas for optimization. In addition, we compared the feature extraction methods of our proposed system to various well-known techniques.

Test 3: Accuracy

The third and final test delved into the accuracy of our CBIR system using well-known image databases such as Oxford5k [76], Paris6k [77], ROxford5k [78], RParis6k [78], and our proposed database: Historical databases. The evaluation criterion for this test was once again the mAP score, which quantified the system's ability to retrieve relevant images accurately. Additionally, we conducted a comparative analysis by comparing our proposed system against several other state-of-the-art CBIR systems. This test not only validated the accuracy of our system but also highlighted its performance in real-world scenarios.

Through a combination of these three tests, we were able to comprehensively assess the effectiveness, efficiency, and accuracy of our proposed CBIR system. The results obtained from these experiments provided valuable insights into the system's strengths and areas for enhancement, ultimately contributing to its refinement and potential future advancements.

4.1 Experiment Setup

The Experiment Setup section of the thesis will comprehensively outline the methodology and framework used to evaluate the proposed Content-Based Image Retrieval system. It will define the core evaluation metrics, including mean Average Precision and computation time. The three pivotal tests will be explained: Test 1, where the efficiency of the filtering algorithm will be assessed against diverse compact network features and classical image processing algorithms; Test 2, which will monitor the computation time across different system components in comparison to state-of-the-art CBIR systems; and Test 3, focusing on accuracy by evaluating mAP scores across multiple datasets and comparing the proposed CBIR system with other prominent solutions. Furthermore, the section will introduce the datasets used for experimentation and establish relevant benchmarks. The Experiments subsection will delve into specific test cases, beginning with an in-depth exploration of experiments conducted on ROxford5k and RParis6k datasets, followed by an analysis of experiments on historical databases. Lastly, the Discussion section will reflect upon the implications of the experimental results, bridging the gap between the conducted experiments and the subsequent conclusions.

4.1.1 Datasets

Datasets play a pivotal role in evaluating the performance of CBIR systems. In our experiments, we utilize a diverse collection of image databases to comprehensively assess the system's retrieval capabilities across various domains. The datasets selected include well-established benchmarks such as Oxford5k [76], Paris6k [77], ROxford5k [78], RParis6k [78], and historical video databases. These datasets encompass a wide range of image types, including urban scenes, landmarks, historical artifacts, and natural landscapes. The diversity in content and context allows us to evaluate the system's robustness in handling different image categories and its adaptability to varied retrieval scenarios. By conducting experiments across multiple datasets, we gain insights into the generalization and scalability of the proposed CBIR system, enhancing its applicability to real-world use cases.

Name of Database	Number of Gallery Images	Number of Query Images
Oxford5k	5063	55
Paris6k	6392	55
ROxford5k	4993	70
RParis6k	6322	70
ROxford + 1M Distractors	4993 + 1M	70
RParis6k + 1M Distractors	6322 + 1M	70
Historical Database	5435 (from 51 videos)	25

Table 4.1: Summary of Databases.

Oxford5k and Paris6k Datasets:

The Oxford5k [76] and Paris6k [77] datasets are widely used benchmarks for evaluating CBIR systems in urban scene recognition and landmark retrieval. The Oxford5k dataset contains 5,062 images of Oxford landmarks, while the Paris6k dataset contains 6,412 images of Paris landmarks. These datasets come with ground truth annotations that specify the correct retrieval results for each query image. Our experiments involve querying these datasets with various test images and analyzing the retrieval performance in terms of precision, recall, and mAP.

Revisited Oxford5k and Revisited Paris6k Datasets:

ROxford5k and RParis6k [78] are enhanced versions of Oxford5k and Paris6k, respectively. These databases address annotation errors present in the original datasets and expand the number of query images, providing a more robust evaluation framework. For both Revisited databases, the number of query images increased from 55 to 70. For both ROxford5k and RParis6k, three difficulty settings—Easy, Medium, and Hard—are defined, each considering different subsets of images as positive and adjusting the treatment of other images accordingly. Notably, these datasets introduce an additional challenge through the inclusion of one million distractor images, significantly augmenting the difficulty in terms of accuracy and speed. This challenge is intended to test the performance limits of image search pipelines under extreme conditions, demanding high-level hardware capabilities. While only a limited number of methods have tested ROxford5k and RParis6k with one million distractors, we are committed to taking on this challenge, further pushing the boundaries of our system's capabilities.

Figure 4.1: Example of a Query and Relevant Images per Category from ROxford5k Database.

Historical Video Database:

In our quest to evaluate the proposed CBIR system comprehensively, we also include a historical video database containing frames from selected 51 videos related to historical figures such as Ibn Battuta, Zheng He, and Marco Polo. This dataset introduces unique challenges owing to variations in imaging conditions like degradation on frames or various spectrums of contracts. Through experiments on historical datasets, we aim to shed light on the system's adaptability to specialized domains, showcasing its potential to contribute to the preservation and aid researchers in the history area. The total duration of the videos in this database is 26632 seconds from 51 videos.

Preparation of Historical Video Database:

The preparation of a well-structured and representative database is a fundamental step in the evaluation of our Content-Based Video Image Retrieval system. In our pursuit of a comprehensive assessment of historical content, we build a database tailored to historical video content. This involved a series of steps to ensure the authenticity, relevance, and accuracy of the dataset.

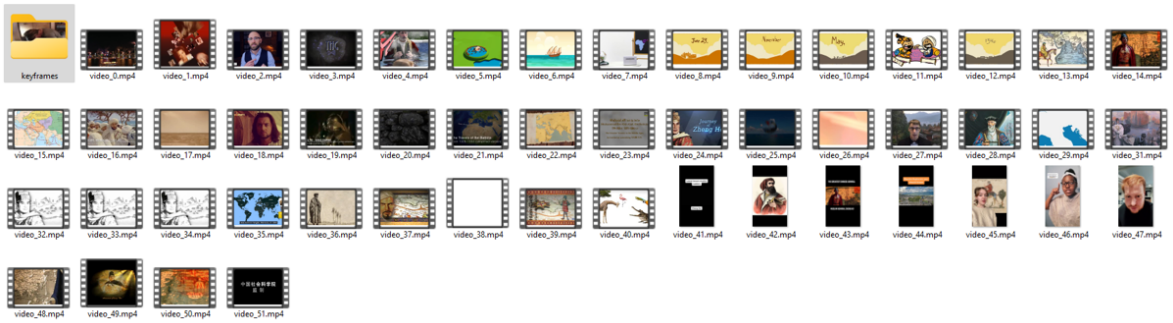


Figure 4.2: Historical Videos Dataset.

Firstly, we collaborated with Professor Andrea Nanetti, a recognized expert in the field, to curate a test set of 51 historical videos. These videos were carefully selected to encompass a diverse range of historical figures and events. Each video was mapped to retain its original identification, preserving the integrity of the historical context.

To facilitate the retrieval process, we proceeded to extract keyframes from the historical videos. Employing a color histogram algorithm, we identified and extracted keyframes that captured pivotal moments within each video. These keyframes were then compiled to form the gallery of our CBIR system. In total, our gallery boasts an impressive collection of 5,435 keyframes, each serving as a potential retrieval candidate.

	A	B
1	Original Name	Mapping
2	01.How+China+Could+Have+Conquered+The+World+_+When+China+Ruled+The+Waves+_+Timeline.mp4	video_0.mp4
3	02.Marco+Polo+-+Journalist+&+Explorer+Biography.mp4	video_1.mp4
4	04.Marco+Polo_+The+World's+Greatest+Explorer.mp4	video_2.mp4
5	05.The+Adventures+of+Marco+Polo.mp4	video_3.mp4
6	06.History-Makers_+Marco+Polo.mp4	video_4.mp4
7	07.The+Life+Story+of+Marco+Polo+in+Under+3+Minutes.mp4	video_5.mp4
8	08.Mystery+of+Marco+Polo's+Journey+to+China+-+A+Prisoner+of+Kublai+Khan_+_+Marco+Polo+Documentary.mp4	video_6.mp4
9	09.Columbus,+de+Gama,+and+Zheng+He!+15th+Century+Mariners.+Crash+Course_+World+History+#21.mp4	video_7.mp4
10	10.Ibn+Battuta+-+The+Great+Traveler+-+Extra+History+-+#1.mp4	video_8.mp4
11	11.Ibn+Battuta+-+Mongols+and+Mystics+-+Extra+History+-+#2.mp4	video_9.mp4
12	12.Ibn+Battuta+-+The+Mad+Sultan+-+Extra+History+-+#3.mp4	video_10.mp4
13	13.Ibn+Battuta+-+Escape+to+China+-+Extra+History+-+#4.mp4	video_11.mp4
14	14.Ibn+Battuta+-+Plague+and+Homecoming+-+Extra+History+-+#5.mp4	video_12.mp4
15	15.Ibn+BATTUTA+–+The+Greatest+EXPLORER+of+All+time+-+KJ+Vids.mp4	video_13.mp4
16	16.The+Greatest+CHINESE+MUSLIM+Explorer+-+KJ+Vids.mp4	video_14.mp4
17	17.Ibn+Battuta_+The+Greatest+Traveller+in+History_.mp4	video_15.mp4
18	18.Ibn+Battuta+1+_+Pilgrimage+&+Journey+to+Persia,+Anatolia+&+Central+Asia.mp4	video_16.mp4
19	19.Ibn+Battuta+2+From+India+to+China.mp4	video_17.mp4
20	20.Ibn+Battuta+3_+Andalusia+to+inland+Africa.mp4	video_18.mp4
21	21.Zheng+He+Voyage+(Ming+Treasure+Fleet).mp4	video_19.mp4
22	22.The+amazing+journey+of+Ibn+Battuta.mp4	video_20.mp4
23	23.Travels+of+Ibn+Battuta.mp4	video_21.mp4
24	24.The+Travels+of+Ibn+Battuta+_+10+Minute+Islamic+History.mp4	video_22.mp4
25	25.The+Tripline+Of+Ibn+Battuta+-+The+Greatest+Explorer+In+History.mp4	video_23.mp4
26	29.The+Journeys+Of+Zheng+He.mp4	video_24.mp4
27	31.Zhenghe+(Chinese+explorer)_+facts+and+his+accomplishments,+the+untold+story.mp4	video_25.mp4
28	32.Zheng+He's+Floating+City_+When+China+Dominated+the+Oceans.mp4	video_26.mp4
29	33.The+Voyages+of+Zheng+He.mp4	video_27.mp4
30	34.Zheng+He_+The+Chinese-Muslim+Explorer+Who+Reached+America+Before+Columbus.mp4	video_28.mp4
31	35.The+Travels+of+Marco+Polo+-+Summary+on+a+Map (1).mp4	video_29.mp4
32	Admiral+Zheng+He+by+Globe+Creative.mp4	video_31.mp4
33	Art+of+Collaboration+Pt1.mp4	video_32.mp4
34	Art+of+Collaboration+Pt2.mp4	video_33.mp4
35	Art+of+Collaboration+Pt3.mp4	video_34.mp4
36	ibn_battuta_pbs.mp4	video_37.mp4

Figure 4.3: Mapping Videos and Original Names.

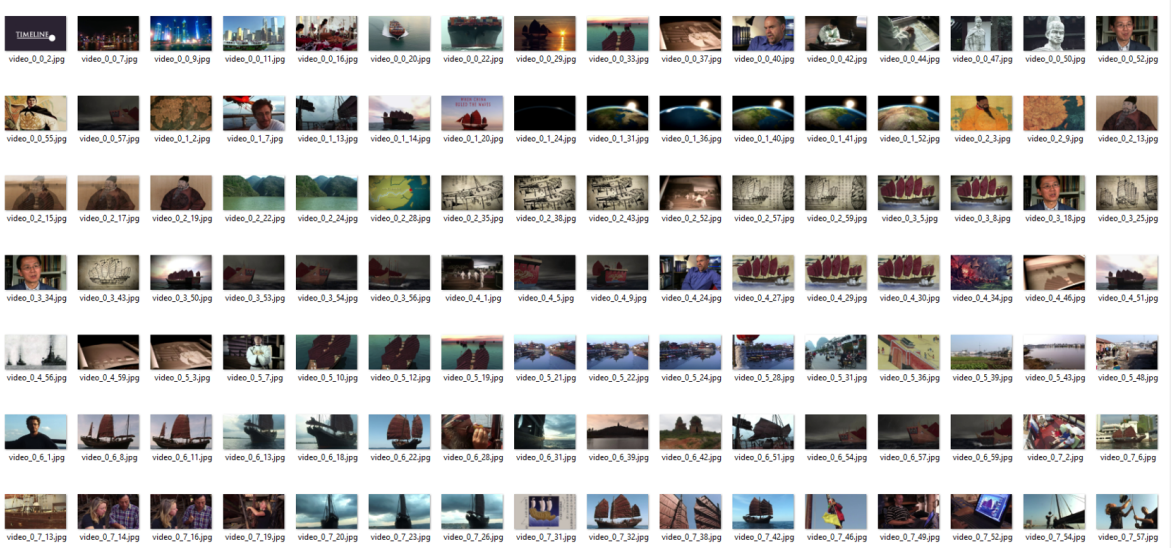


Figure 4.4: Keyframe Extraction and Gallery Creation.

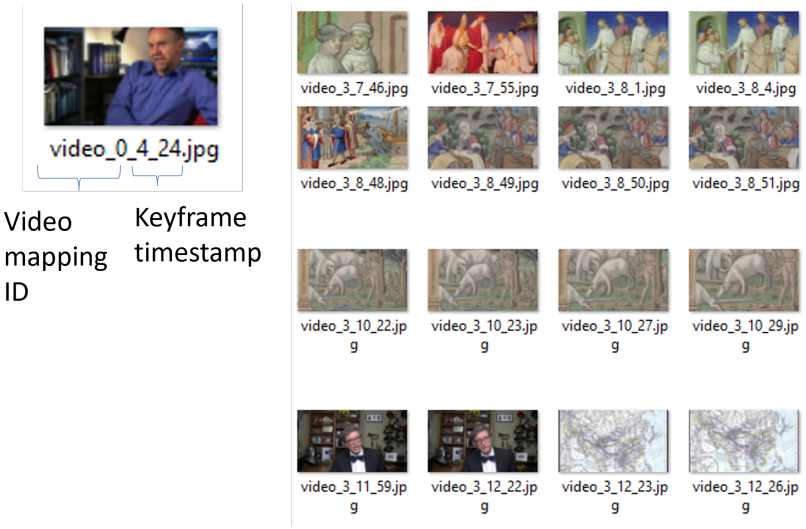


Figure 4.5: Annotation of Extracted Keyframes.

In order to validate the accuracy and efficacy of our CBIR system, a rigorous validation process was undertaken. We handpicked 25 query images, each featuring historical figures of significance. To ensure precision, the relevant frames within the videos were carefully annotated as ground truth. It's important to note that the selection of both query images and their corresponding relevant frames underwent validation by Professor Andrea Nanetti.

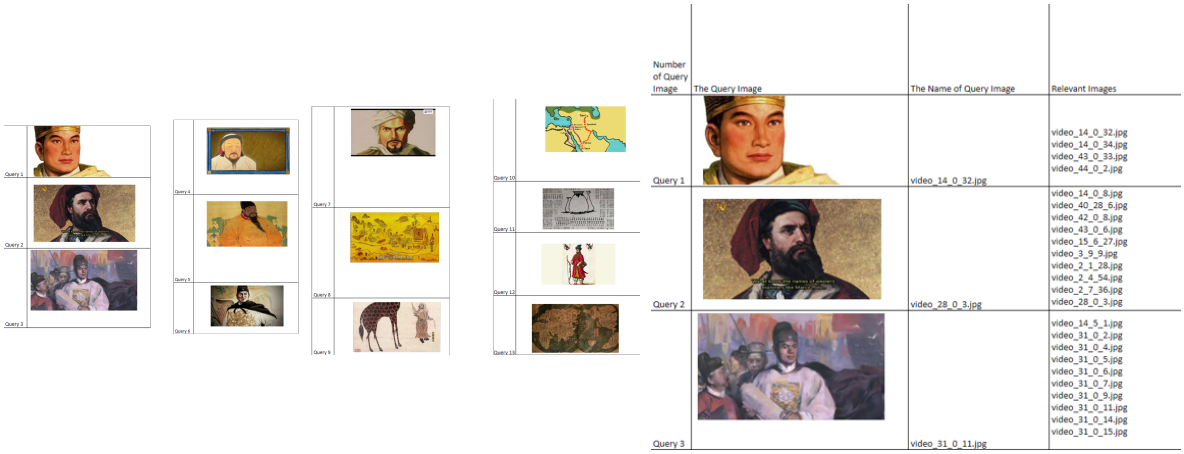


Figure 4.6: Selection and Validation Process for Query Images and Relevance Annotations.

This preparation process forms the keystone of our evaluation efforts. By assembling an authentic and expert-validated database, we aim to guarantee that our experiments reflect real-world retrieval scenarios. The collaboration with Professor Nanetti, his expertise, and validation further reinforce the reliability and accuracy of our assessment. With a curated gallery of keyframes and a carefully annotated set of queries, we are well-equipped to comprehensively evaluate the performance and capabilities of our CBIR

system in the context of historical video content.

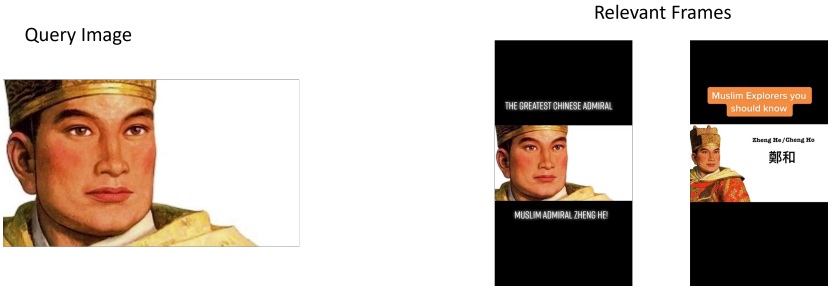


Figure 4.7: Example of Query and Selected Ground Truth Relevant Frames.

Through our experimentation with diverse datasets, we aim to provide a comprehensive evaluation of the proposed CBIR system's capabilities, addressing various challenges associated with image retrieval across different contexts and domains.

4.1.2 Evaluation Metrics

The evaluation of the proposed system is guided by a comprehensive set of metrics, each tailored to assess different facets of its performance. One of the fundamental metrics employed is the *mean Average Precision*. mAP is a widely adopted measure to evaluate the retrieval accuracy of the system. It quantifies the system's ability to rank and retrieve relevant images across various query instances. For each query, the precision values at different ranks are computed, and the average of these precision values is taken, yielding the mAP score. Additionally, the computation time is a crucial metric in assessing the system's efficiency. The time taken for different stages of the system, including feature extraction and retrieval, is meticulously measured. This provides insights into the computational demands of each component and allows for comparisons with state-of-the-art CBIR systems. For instance, the computation time of feature extraction from a query image can be calculated as the difference between the start and end timestamps of the process. These evaluation metrics collectively offer a comprehensive understanding of the system's accuracy, efficiency, and retrieval capabilities.

4.1.2.1 Mean Average Precision (mAP)

The Mean Average Precision is a widely used metric in information retrieval and object detection tasks, including CBIR. It provides a comprehensive measure of the retrieval accuracy of the system by considering both precision and recall across multiple queries. Notably, while the core concept of mAP remains consistent between instance retrieval and object detection, there are nuanced differences in their application. In instance retrieval, mAP assesses the retrieval performance of relevant images from a dataset in response to specific queries. In contrast, in object detection, mAP evaluates the accuracy of localizing and classifying objects within images, often considering detection at varying levels of precision and overlap. Despite these variations, mAP remains a valuable tool for quantifying retrieval accuracy across diverse applications.

Precision and recall are fundamental concepts in information retrieval. Precision is the ratio of relevant items retrieved to the total number of retrieved items, while recall is the ratio of relevant items retrieved to the total number of relevant items in the dataset.

Mathematically, precision and recall can be expressed as:

$$P = \frac{\text{Number of Relevant Items Retrieved}}{\text{Total Number of Retrieved Items}} = \frac{TP}{TP + FP}$$

$$R = \frac{\text{Number of Relevant Items Retrieved}}{\text{Total Number of Relevant Items in the Dataset}} = \frac{TP}{TP + FN}$$

Here, TP represents the number of true positives (relevant items correctly retrieved), FP represents the number of false positives (irrelevant items incorrectly retrieved), and FN represents the number of false negatives (relevant items not retrieved).

For each query, the precision-recall curve is constructed by varying the retrieval threshold. Precision is plotted on the y-axis, and recall on the x-axis. The area under this curve is the Average Precision (AP) for that query. AP captures the system's ability to rank relevant images higher than irrelevant ones. Mathematically, the AP for a query is calculated as the area under the precision-recall curve:

$$\text{AP@}n = \frac{1}{\text{GTP}} \sum_{k}^{n} \text{P@}k \times \text{rel@}k$$

Where:

AP@n is the Average Precision at threshold n

GTP is the total number of ground truth positive items in the dataset

$\sum_{k}^{n}$ signifies summation over the range of retrieved items from 1 to n

P@k is the Precision at the kth retrieved item

rel@k is a binary indicator (0 or 1) for the relevance of the kth retrieved item

The mAP is computed by averaging the AP values across all queries. It provides a single value that summarizes the system's retrieval performance over the entire dataset. Mathematically, mAP can be expressed as:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

Where N is the total number of queries, and AP_i is the Average Precision for the ith query.

Example on the calculation of mAP:

Consider a Content-Based Image Retrieval (CBIR) experiment using a dataset of 10 images. We will calculate the Mean Average Precision (mAP) for two different queries within this dataset.

Query 1: Out of the 10 images, assume that 4 images are relevant to Query 1 (True Positives, TP = 4), and 6 images are irrelevant (False Positives, FP = 6). Our retrieval system ranks these images based on similarity scores as follows:

Ranked list for Query 1: **Image 2: Correct**, **Image 5: Incorrect**, **Image 8: Incorrect**, **Image 9: Correct**.

Image	Is Relevant?
1. Image 2	✓
2. Image 5	×
3. Image 8	×
4. Image 9	✓

Table 4.2: Ranked list and Correctness of Ranking for Query 1.

We calculate the precision and recall for each position in the ranked list for Query 1:

- Precision (P@1) = 1/1 = 1.0, Recall (R@1) = 1/4 = 0.25.
- Precision (P@2) = 1/2 = 0.5, Recall (R@2) = 1/4 = 0.25.
- Precision (P@3) = 1/3 = 0.33, Recall (R@3) = 1/4 = 0.25.
- Precision (P@4) = 2/4 = 0.5, Recall (R@4) = 2/4 = 0.5.

Calculating the Average Precision (AP) for Query 1 using the earlier definition:

$$AP@Query1 = \frac{1}{4}(1 \times 1 + 0.5 \times 0 + 0.33 \times 0 + 0.5 \times 1) = 0.208.$$

Query 2: Now, assume that 3 images are relevant to Query 2 (TP = 3), and 7 images are irrelevant (FP = 7). The ranked list for Query 2 is not provided for briefness.

Precision, recall, and AP for Query 2 are also calculated in the same manner as for Query 1.

Mean Average Precision (mAP): The mAP is the average of the AP values for all queries. In this case, let's assume that AP@Query2 = 0.65.

$$mAP = \frac{AP@Query1 + AP@Query2}{2} = \frac{0.208 + 0.65}{2} = 0.429.$$

In this example, the mAP value of 0.429 indicates the average correctness of the retrieval system in returning relevant images across both queries. It captures the system's performance in terms of both precision and recall, providing a comprehensive measure of its retrieval accuracy across different scenarios.

4.1.2.2 Computation Time

Efficiency is a crucial aspect of a CBIR system, extending beyond retrieval accuracy to encompass the system's processing speed. Computation time serves as a pivotal performance metric, directly influencing the user experience and the system's applicability to real-time scenarios. In our comprehensive evaluation, we not only focus on retrieval accuracy but also delve into the intricacies of computation time associated with different stages of the proposed CBIR system.

For computation time, we consider two distinct metrics that provide valuable insights into the system's efficiency. The first metric involves calculating the ratio between the input video length and the total computation time. This ratio offers a measure of the system's throughput, highlighting how quickly the system can process queries relative to the duration of the input video. A higher ratio indicates greater efficiency in handling longer videos.

The second metric corresponds to the computation time for individual stages within the system. We specifically examine the time required for critical processes such as feature extraction per frame and matching per query image. By quantifying the time needed for these stages, we can identify potential bottlenecks or areas for optimization. This detailed breakdown of computation time aids in fine-tuning the system's performance and ensuring that each stage operates efficiently.

To provide a comprehensive assessment, we compare these computed times with those of state-of-the-art CBIR systems, allowing us to measure the efficiency of our approach in the context of existing solutions.

Examining computation time offers a practical perspective on the feasibility of the proposed system. It enables us to determine its suitability for handling large-scale image databases and its ability to cater to real-time retrieval demands. Ultimately, a well-rounded evaluation of computation time contributes to a holistic understanding of the system's performance, ensuring that it not only delivers accurate results but also does so efficiently, meeting the requirements of a wide range of applications.

4.2 Experiments

4.2.1 Tests 1: Filtering Algorithm

In this section, we conduct a series of experiments to evaluate various algorithms, including both compact deep learning models and traditional algorithms, to identify a suitable, efficient, and accurate algorithm for filtering out irrelevant frames. The experiments focus on two key metrics: computation speed and retrieval accuracy.

4.2.1.1 Speed Test:

For the speed test, we evaluate the efficiency of different algorithms in terms of their processing time. We use two metrics to assess the speed performance. First, we calculate the average ratio of video duration divided by the computation time of the CBIR system using the Historical Videos database. This metric helps us understand how

efficiently the algorithms can process videos of varying lengths. We experiment with compact deep-learning models and traditional algorithms to compare their performance.

We first evaluate the speed performance of various compact deep learning models, including MobileNetV2, VGG16, ResNet50, and InceptionV3. The following table presents the average ratios for these models:

Deep Learning Models	MobileNetV2	VGG16	ResNet50	InceptionV3
Average Ratio	**39**	18	22	30

Table 4.3: Comparison of Average Ratios for Models from Compact Networks.

We also assess the speed performance of traditional algorithms, including ORB, SIFT, and AKAZE. The following table displays the average ratios for these algorithms:

Traditional Algorithms	ORB	SIFT	AKAZE
Average Ratio	**209**	36	58

Table 4.4: Comparison of Average Ratios for Models from Traditional Algorithms.

4.2.1.2 Accuracy Test:

Moving on to the accuracy test, we evaluate the retrieval accuracy of the different algorithms. We employ two popular benchmark datasets, Oxford5k and Paris6k, to measure the performance.

We begin by assessing the accuracy of the compact deep learning models on the Oxford5k and Paris6k datasets. The following table presents the mean Average Precision (mAP) values for each model:

	MobileNetV2	VGG16	ResNet50	InceptionV3
Oxford5k	38.4	30.3	36.1	**41.8**
Paris6k	**50.1**	37.5	48.0	47.6

Table 4.5: Performance Comparison of Different Models on Oxford5k and Paris6k Datasets.

We also evaluate the accuracy of traditional algorithms, including ORB, SIFT, and AKAZE, using the Oxford5k and Paris6k datasets. The following table displays the mAP values for each algorithm:

Traditional Algorithms	ORB	SIFT	AKAZE
Oxford5k	17.8	**21.1**	14
Paris6k	14.5	**16**	14.2

Table 4.6: Performance of Traditional Algorithms on Oxford5k and Paris6k.

Through these experiments, we aim to identify algorithms that strike a balance between speed and accuracy, providing insights into the best choices for our CBIR system's filtering algorithm.

The speed test results are crucial in determining the computational efficiency of different algorithms, especially when processing historical videos of varying lengths. Looking at Table 4.3 we observe significant differences in the average ratios across the compact deep learning models. Notably, MobileNetV2 stands out with an impressive average ratio of **39**, indicating its ability to process video content more efficiently compared to other models. This result suggests that MobileNetV2 is well-suited for real-time or large-scale video processing scenarios.

In addition, the traditional algorithms in Table 4.4 exhibit higher average ratios, with ORB leading at an average ratio of **209**. This implies that traditional algorithms may require fewer computational resources, making them possible choices for applications demanding swift video retrieval.

Accuracy is a fundamental aspect of any filtering algorithm. The performance comparison of different models and algorithms on benchmark datasets provides insights into their ability to accurately filter out relevant frames.

From Table 4.5 we observe that InceptionV3 achieves the second highest for the Oxford5k dataset and the highest mAP values for the Paris6k dataset, with values of **41.8** and **47.6**, respectively. The MobileNetV2 architecture achieves the highest mAP for the Oxford5k database and the second highest mAP values for the Paris6k dataset, with values of **38.4** and **50.1**, respectively. This indicates that InceptionV3 and MobileNetV2 architectures excel in capturing the relevancy of frames within historical videos, showcasing their potential for accurate retrieval of historical content.

Comparing traditional algorithms in Table 4.6 SIFT and ORB stand out as competitive performers with mAP values of **21.1** and **16**, respectively, on the Oxford5k dataset. These traditional algorithms demonstrate their ability to accurately filter historical frames based on visual cues.

Overall, the speed and accuracy tests highlight the trade-offs between various filtering algorithms. While compact deep learning models like MobileNetV2 offer superior speed, models such as InceptionV3 exhibit exceptional accuracy. On the other hand, traditional algorithms like SIFT and ORB have exceptional computational efficiency with low accuracy. For the filtering stage of our proposed system, we decided to employ MobileNetV2 features due to high efficiency with a slight drop of mAP compared to InceptionV3.

4.2.2 Test 2: Computation Time

In this section, the system is tested in terms of efficiency. The efficiency test consists of two metrics: the ratio between video duration and computation time of the entire CBIR process using the Historical Videos database, and the duration of feature extraction per frame using various databases. The first test involves a comparison between the state-of-the-art ResNet101 + SOLAR CBIR system and our proposed system. The second test focuses on comparing different deep-learning models in terms of their feature extraction duration per frame.

4.2.2.1 Computation Time Comparison

The first aspect of efficiency testing involves assessing the computation time of the CBIR system. We compare our proposed system, which employs MobileNetV2 for filtering and ResNet101 along with SOLAR for re-ranking, against the ResNet101 + SOLAR system using the historical videos dataset. As a reminder, this dataset contains 5435 keyframes from 51 videos with a total duration of 26632 seconds. The number of queries is 25 in this database. The results are presented in Table 4.7.

Method	**Computation Time (s)**	**Video Duration/ Computation Time**
MobileNetV2 + ResNet101 + SOLAR	1011 - 1400	26 - 19
MobileNetV2 + ResNet101 + SOLAR Online	16.9 - 406	1575 - 66
MobileNetV2 + ResNet101 + SOLAR Offline	994	27
ResNet101 + SOLAR	3510	7.59

Table 4.7: Comparison of Computation Time and Video Duration.

The table provides a comparison between the computation times of the two systems, highlighting the advantages of our proposed MobileNetV2 + ResNet101 + SOLAR system in terms of efficiency. Our proposed configuration shines with a remarkable efficiency advantage, as evidenced by a significantly higher Video Duration/Computation Time ratio with at least **19**, demonstrating its superiority for real-time retrieval scenarios. To be more specific the computation time for each stage of our proposed system is presented for the online phase in table 4.9 and for the offline phase in table 4.8.

For the **offline** phase:

Process	**Total**	**Per Frame**
Keyframe Extraction	926.2	0.17
MobileNetV2 Feature Extraction	67	0.01
Building ANNOY indices with MobileNetV2 features	0.7	-

Table 4.8: The Computation Time for Each Stage of Offline Phase.

For the **online** phase:

Process	**Total**	**Per Frame**
MobileNetV2 Feature Extraction	0.24	0.01
ResNet101 + SOLAR Feature Extraction (Query)	5	0.2
ResNet101 + SOLAR Feature Extraction (Keyframes)	0 - 375	0.2
Searching with pre-built ANNOY indices MobileNetV2	0.7	0.03
Linear Search with ResNet101 + SOLAR features	0.2	0.008

Table 4.9: The Computation Time for Each Stage of Online Phase.

In the online phase, the computation time varies significantly based on the availability of ResNet101 + SOLAR features for shortlisted keyframes. This variation is due to the loading of features for shortlisted keyframes when available, which significantly accelerates the online phase. We account for both scenarios: feature extraction

(maximum computation time) and feature loading (minimum computation time) of all shortlisted keyframes in our experiment, providing a comprehensive view of the system's performance. Per query, the computation time of the online phase varies between **7.33** and **43** seconds.

The computation time of the process of ResNet101 + SOLAR Feature Extraction of keyframes is highly dependent on the threshold of shortlisting and availability of pre-stored ResNet101 + SOLAR features. Usage of a lower threshold results in a faster process and using a higher threshold results in a slower process due to more keyframes being shortlisted. In addition, the availability of pre-stored results in a faster process.

The test is done by selecting a shortlist of the top 100 keyframes similar to the input query image using MobileNetV2 features. The results provide that the feature extraction for ResNet101 + SOLAR stage is the bottleneck of our system in terms of efficiency. Progressively saving features from earlier runs will speed up the online phase of the proposed system. Lastly, our proposed system overall provides a fast system and is suitable for video image retrieval tasks with the ratio of video duration and computation time at least **19** and at most **26**.

4.2.2.2 Feature Extraction Duration Comparison

Since we discovered that the bottleneck for the CBIR system is feature extraction, the second aspect of efficiency testing focuses on the duration of feature extraction per frame for various deep-learning models. The results are presented in Table 4.10.

CNN Backbone	**Duration of Feature Extraction per Frame (seconds)**
MobileNetV2	**0.011**
AlexNet	0.042
VGGNet	0.053
ResNet50	0.05
ResNet101	0.054
ResNet150	0.063
ResNet101-GeM	0.114
ResNet101-GeM + SAHA	0.9
ResNet101-GeM + LoFTR	2.28
ResNet101-GeM + QGE	0.115
ResNet101-GeM + SIFT	10.3
ResNet101-GeM + k-reciprocal	1.23
ResNet101 + SOLAR	0.215

Table 4.10: Comparison of Duration of Feature Extraction per Frame for Different CNN Backbones.

The table unfolds a detailed examination of the time required by various CNN backbones for extracting features per frame. Strikingly, MobileNetV2 emerges as a standout choice due to its exceptional efficiency, with **0.011** seconds per frame. The particular selection of MobileNetV2 for the filtering stage is verified by this impressive efficiency, ensuring rapid processing and real-time retrieval capability. The contrast between MobileNetV2 and the state-of-the-art ResNet101 + SOLAR, with a feature extraction time of 0.215 seconds, underscores the efficiency gains brought by our design.

4.2.3 Test 3: Accuracy

4.2.3.1 Experiments on Oxford5k and Paris6k

The accuracy of different models is evaluated on the Oxford5k and Paris6k datasets, and the results are presented in Table 4.11.

Model	**Oxford5k mAP (%)**	**Paris6k mAP (%)**
MobileNetV2 + ResNet101 + SOLAR	86.92	92.58
ResNet101 + SOLAR	88.27	95.04
ResNet101 + GeM	88.74	94.52
ResNet150	88.8	89.2
AlexNet	61.7	71.5
VGGNet	80.3	83.8
ResNet101	85.2	88.8
ResNet101-GeM + QGE (Top-3)	91.02	97.21
ResNet101-GeM + QGE (Top-100)	84.84	97.17

Table 4.11: Comparison of mAP (%) for Different Models on Oxford5k and Paris6k Databases.

4.2.3.2 Experiments on ROxford5k and RParis6k

The accuracy experiments are extended to the more challenging ROxford5k and RParis6k datasets, and the results are shown in Table 4.12.

	Easy		Medium		Hard	
	R-Oxford	R-Paris	R-Oxford	R-Paris	R-Oxford	R-Paris
MobileNetV2 + ResNet101 + SOLAR	82.6	86.74	69.24	75.35	45.71	59.77
ResNet101 + SOLAR	85.88	92.95	69.9	81.57	47.91	64.45
ResNet101-GeM	82.24	92.29	55.8	69.7	28.1	47
ResNet150	–	–	57.1	70.1	29.5	48.9
AlexNet	–	–	43.3	58	17.1	29.7
VGGNet	–	–	56.2	69.2	28.9	46.9
ResNet101	–	–	55.8	69.7	28.1	47
ResNet101-GeM + QGE	84.65	94.13	69.07	87.93	44.26	77.23
ResNet101-GeM + SIFT	78.23	92.26	61.5	77.73	36.63	55.76

Table 4.12: Comparison of mAP (%) for Different Models on ROxford5k and RParis6k Databases.

The accuracy experiments are extended to the much more challenging ROxford5k and RParis6k datasets with 1 million distractor images included in the gallery, and the results are shown in Table 4.13.

	Easy		Medium		Hard	
	ROxf+R1M	RPar+R1M	ROxf+R1M	RPar+R1M	ROxf+R1M	RPar+R1M
MobileNetV2 + ResNet101 + SOLAR	76.06	81.45	58.94	60.32	33.33	37.55
ResNet101 + SOLAR	79.61	85.84	61.45	64.7	35.44	41.2
ResNet101-GeM	57.64	57.6	40.07	40.01	19.11	21
ResNet101-GeM + QGE (Top-3)	74.03	86.47	54.83	66.48	30.45	43.17
ResNet101-GeM + QGE (Top-100)	62.72	84.65	43.3	69	16.66	46.67

Table 4.13: Comparison of mAP (%) for Different Models on Easy, Medium, and Hard Splits of ROxford5k and RParis6k Databases with 1M Distractors.

4.2.3.3 Experiment on Historical Databases

The accuracy of the models is also evaluated on the Historical Videos Database, as summarized in Table 4.14.

Historical Videos Database	**mAP (%)**
MobileNetV2 + ResNet101 + SOLAR	70.38
ResNet101 + SOLAR	71.71

Table 4.14: Comparison of mAP (%) for Different Models on Historical Videos Database.

4.3 Discussion

The results presented in the tables highlight the performance in terms of accuracy and efficiency, of various models on different datasets. We observe that the MobileNetV2 + ResNet101 + SOLAR model, our proposed system, consistently achieves remarkable efficiency advantages and competitive accuracy across all datasets, showcasing its robustness in various scenarios.

- While discussing the computation time comparison, we observed that our proposed system demonstrates remarkable efficiency advantages. Notably, it outperforms the baseline ResNet101 + SOLAR in terms of computation time, with a Video Duration/Computation Time ratio of at least 19, making it highly suitable for real-time retrieval scenarios. The breakdown of computation times for both online and offline phases highlights the efficiency of each stage. The usage of shortlisting the keyframes and storing/loading available ResNet101 + SOLAR features for each run gradually makes our proposed system efficient.
- In the assessment of feature extraction duration, we discovered that MobileNetV2 exhibits exceptional efficiency, requiring only 0.011 seconds per frame. This choice for the filtering stage ensures rapid processing and real-time retrieval capability, making it a standout option for efficiency-focused applications. In contrast, the feature extraction time of ResNet101 + SOLAR is 0.215 seconds, underscoring the efficiency gains achieved by our design.

The efficiency test results provide valuable context for the overall performance of our CBIR system. When considering these results alongside accuracy metrics, several key observations emerge:

- The superiority of ResNet101 + SOLAR on the Oxford5k and Paris6k datasets indicates the effectiveness of the SOLAR method for feature extraction. However, when evaluated on more challenging datasets like ROxford5k and RParis6k, the performance of this model, including MobileNetV2 + ResNet101 + SOLAR, seems to degrade with a drop of 0.95% on Oxford5k and 3.8% on Paris6k compared to ResNet101 + SOLAR, due to the increased complexity of the queries.

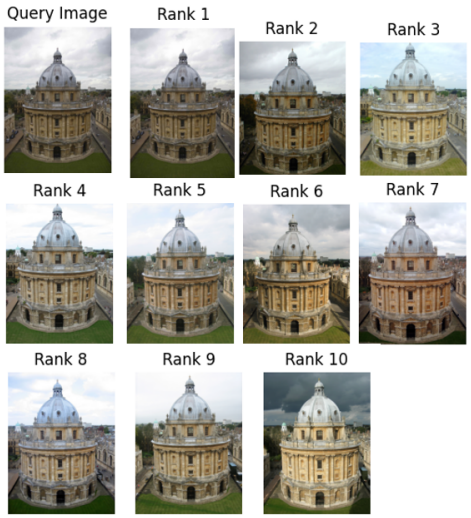
- Interestingly, the ResNet101-GeM model demonstrates competitive accuracy on the RParis6k dataset, while MobileNetV2 + ResNet101 + SOLAR remains a strong performer across the board, showcasing its potential for certain specific scenarios. The effectiveness of the ResNet101-GeM + QGE model on the harder subsets of the datasets suggests the significance of query-guided embedding, which is also present in MobileNetV2 + ResNet101 + SOLAR.

- Furthermore, the experiments on the Historical Videos Database highlight the adaptability of our proposed system of MobileNetV2 + ResNet101 + SOLAR and ResNet101 + SOLAR to historical content, underlining their potential for real-world applications beyond traditional image datasets. When assessing the Historical Videos Database, our proposed system, MobileNetV2 + ResNet101 + SOLAR achieves a competitive mAP of 70.38%. The mAP drop compared to ResNet101 + SOLAR is only 1.33%, indicating its suitability for historical content retrieval.

- These observations, including the consistent performance of MobileNetV2 + ResNet101 + SOLAR, provide a comprehensive view of the strengths and weaknesses of each model under various conditions. They also highlight the potential for further optimization and research in content-based image retrieval, particularly in addressing challenges posed by complex queries and historical content.

4.4 Visualization of the Retrieval

This section provides a visual representation of the outcomes obtained through the execution of our proposed CBVIR system. Each search conducted during the evaluation process results in the display of the top ten matched images, allowing us to gain insights into the system's performance. In addition to assessing the search results, to facilitate a seamless user experience, a user interface is developed for the search engine, available at https://github.com/dorukbarokas/Efficient-CBVIR.git. This interface empowers users to upload local videos of interest and initiate searches by inputting query images.



(a) Visual search results for an easy task.

(b) Visual search results for a challenging task.

Figure 4.8: Comparison of visual search results for easy and challenging tasks.

5 Conclusion

5.1 Summary

Our proposed video retrieval method presents a comprehensive approach to efficiently and effectively retrieve keyframes from video content while prioritizing speed. This method capitalizes on two distinct phases: the offline phase, which includes preprocessing steps such as Keyframe Extraction, feature extraction of MobileNetV2 features, and ANNOY index tree construction, and the online phase, where users input query images for retrieval.

Throughout our methodology, careful consideration has been given to optimizing the retrieval process. We employ techniques such as Singular Value Decomposition (SVD) dimensionality reduction for the keyframe extraction phase, which helps streamline feature matrices and improve efficiency. Furthermore, the adoption of MobileNetV2 as a compact Convolutional Neural Network backbone enhances computation speed, making it a standout choice for Content-Based Image Retrieval in video search. The use of approximate nearest-neighbor (ANNOY) tree-based search methods, coupled with effectively shortlisting the relevant keyframes, ensures that the retrieval process is both rapid and precise.

To address any potential trade-offs between speed and accuracy, we introduce a re-ranking module that utilizes global features and ResNet101 + SOLAR to refine the retrieval results. This module ensures that our system continues to deliver accurate results, even in scenarios where speed is paramount.

Our contribution lies in providing a versatile tool that can significantly expedite image retrieval from videos, making it particularly valuable for applications such as historical research. As our system progressively enhances its performance with each query, it offers a valuable resource for researchers and users seeking efficient, high-speed video retrieval without compromising the quality of results.

5.2 Future Work

While our current system represents a significant step forward in video image retrieval, there are areas where further research and development could enhance its capabilities:

- **Scaling for Larger Datasets:** As datasets continue to grow, optimizing our system's scalability to handle even larger video databases will be essential. This includes more efficient memory management and distributed processing. This project focused on the trade-off of efficiency against accuracy. Efficient memory management might optimize our proposed system.

- **Efficient Feature Extraction During the Online Phase:** Given the increased computational demands of feature extraction per frame with ResNet101 + SOLAR, it is imperative to explore alternative algorithms. After the implementation of our proposed system, new and innovative models have emerged. For instance, models like DINOv2: A Self-supervised Vision Transformer by Meta AI [79] or YOLO-NAS: A Next-Generation Object Detection Model from Deci's Neural Architecture Search Technology [80] offers both speed and accuracy in feature extraction, making them attractive options for our system's online phase.
- **Real-time Video Retrieval:** Extending our proposed system to support real-time video retrieval, where users can retrieve keyframes from live video streams, opens up new applications in surveillance and video monitoring.

Bibliography

[1] M. Safar, C. Shahabi, and X. Sun, "Image retrieval by shape: A comparative study," in *IEEE International Conference on Multimedia and Expo (I)*, 2000, pp. 141–144.

[2] A. Sharma, "Convolutional neural networks in python," Dec 2017. [Online]. Available: https://www.datacamp.com/tutorial/convolutional-neural-networks-python

[3] R. Thakur, "Step by step vgg16 implementation in keras for beginners," Nov 2020. [Online]. Available: https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c

[4] B. Ridha Ilyas, M. Beladgham, K. Merit, and A. taleb ahmed, "Illumination-robust face recognition based on deep convolutional neural networks architectures," vol. Vol 18, p. 1015 1027, 12 2019.

[5] L. Solbakken, "Using approximate nearest neighbor search in real world applications," Dec 2020. [Online]. Available: https://towardsdatascience.com/using-approximate-nearest-neighbor-search-in-real-world-applications-a75c351445d

[6] T. Ng, V. Balntas, Y. Tian, and K. Mikolajczyk, "SOLAR: second-order loss and attention for image retrieval," *CoRR*, vol. abs/2001.08972, 2020. [Online]. Available: https://arxiv.org/abs/2001.08972

[7] T. Ng, "Solar: Second-order loss and attention for image retrieval - long video eccv2020," Aug 2020. [Online]. Available: https://www.youtube.com/watch?v=tfIW0widG9k&ab_channel=tinyoooo

[8] E. Bernhardsson. (2023) Annoy: Approximate nearest neighbors in c++/python. Python package version 1.17.0. [Online]. Available: https://pypi.org/project/annoy/

[9] ——. (2015) Nearest neighbors and vector models - part 2 - algorithms and data structures. Accessed in March 2023. [Online]. Available: https://erikbern.com/2015/10/01/nearest-neighbors-and-vector-models-part-2-how-to-search-in-high-dimensional-spaces.html

[10] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," 2019.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[12] A. Nanetti. (2023) Engineering historical memory. [Online]. Available: https://engineeringhistoricalmemory.com/

[13] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, "Deep learning for instance retrieval: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7270–7292, 2023.

[14] A. Alzu'bi, A. Amira, and N. Ramzan, "Content-based image retrieval with compact deep convolutional features," *Neurocomputing*, vol. 249, pp. 95–105. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231217306185

[15] D. Lowe, "Object recognition from local scale-invariant features," *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, 01 2001.

[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," 11 2011, pp. 2564–2571.

[17] P. Fernández Alcantarilla, "Fast explicit diffusion for accelerated features in non-linear scale spaces," 09 2013.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[19] R. Datta, D. Joshi, J. Li, and J. Z. Wang, "Image retrieval: Ideas, influence, and trends of the new age," *ACM Computing Surveys (CSUR)*, vol. 40, no. 2, 2008.

[20] S.-F. Chang and A. Hsu, "Image information systems: Where do we go from here?" *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 5, pp. 431–442, 1992.

[21] V. Tyagi, *Content-based image retrieval: Ideas, influences, and current trends.* SPRINGER Verlag, SINGAPOR, 2018.

[22] J. Assfalg, A. D. Bimbo, and P. Pala, "Using multiple examples for content-based retrieval," in *Proceedings of International Conference Multimedia and Expo*, 2000.

[23] K. Sethi and I. Coman, "Mining association rules between low-level image features and high-level concepts," in *Proceedings of the SPIE Data Mining and Knowledge Discovery, vol. III*, 2001, pp. 279–290.

[24] A. Mojsilovic and B. Rogowitz, "Capturing image semantics with low-level descriptors," in *Proceedings of the ICIP*, 2001, pp. 18–21.

[25] R. Jain, "Visual information management systems," in *Proceedings of US NSF Workshop Visual Information Management Systems*, 1992.

[26] J. Huang, S. Kumar, and M. Mitra, "Combining supervised learning with color correlograms for content-based image retrieval," in *Proceedings 5th ACM Multimedia Conference*, 1997, pp. 325–334.

[27] K. Van de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, 2010.

[28] J. Smith and S. Chang, "Automated binary texture feature sets for image retrieval," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, 1996, pp. 2239–2242.

[29] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, 1996.

[30] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern Recognition*, vol. 37, pp. 1–19, 2004.

[31] S. Loncaric, "A survey of shape analysis techniques," *Pattern Recognition*, vol. 31, no. 8, pp. 983–1001, 1998.

[32] E. Salahat and M. Qasaimeh, "Recent advances in features extraction and description algorithms: A comprehensive survey," 2017.

[33] K. Mikolajczyk and J. Matas, "Improving descriptors for fast tree matching by optimal linear projection," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[34] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[35] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 2005, pp. 1508–1515 Vol. 2.

[36] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," vol. 6314, 09 2010, pp. 778–792.

[37] W. Huang, L.-D. Wu, H.-C. Song, and Y.-M. Wei, "Rbrief: a robust descriptor based on random binary comparisons," *Computer Vision, IET*, vol. 7, pp. 29–35, 02 2013.

[38] S. A. K. Tareen and Z. Saleem, "A comparative analysis of sift, surf, kaze, akaze, orb, and brisk," in *2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2018, pp. 1–10.

[39] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *Computer Vision – ECCV 2012*. Springer, Berlin, Heidelberg, 2012, vol. 7577.

[40] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, and M. S. Lew, "Deep learning for instance retrieval: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7270–7292, 2023.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[42] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *ACM MM*, 2014, pp. 157–166.

[43] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Applied Sciences*, vol. 12, no. 18, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/18/8972

[44] A. Ravi and A. Nandakumar, "A multimodal deep learning framework for scalable content-based visual media retrieval," *arXiv preprint arXiv:2105.08665*, 2021.

[45] L. Lebron Casas and E. Koblents, "Video summarization with lstm and deep attention models," in *International Conference on MultiMedia Modeling*. Springer, 2018, pp. 67–79.

[46] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: A decade survey of instance retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1224–1244, May 2018.

[47] G. Amato, P. Bolettieri, F. Carrara, F. Falchi, C. Gennaro, N. Messina, L. Vadicamo, and C. Vairo, "Visione at video browser showdown 2022," in *MultiMedia Modeling: 28th International Conference, MMM 2022, Phu Quoc, Vietnam, June 6–10, 2022, Proceedings, Part II*. Springer-Verlag, 2022, pp. 543–548. [Online]. Available: https://doi.org/10.1007/978-3-030-98355-0_52

[48] A. Kilgarriff and C. Fellbaum, "Wordnet: An electronic lexical database," *Language*, vol. 76, p. 706, 09 2000.

[49] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[50] V. A. Wankhede and P. S. Mohod, "Content-based image retrieval from videos using cbir and abir algorithm," in *2015 Global Conference on Communication Technologies (GCCT)*. IEEE, 2015, pp. 767–771.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] S. Li, "Deep learning-empowered content-based video image retrieval," 08 2023. [Online]. Available: http://resolver.tudelft.nl/uuid:d16300c5-6988-4172-8c20-0e2dfff8949f

[53] Y. Yao, "Image-based query search engine via deep learning," master thesis, Delft University of Technology, 07 2022. [Online]. Available: http://resolver.tudelft.nl/uuid:4a2c9c6f-b2b8-41d6-9b70-69c4f246c964

[54] H. Jégou, M. Douze, and C. Schmid, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[55] Y. Tan, J. Yuan, C. Fang, and H. Jin, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 186–201.

[56] S. Su, C. Zhang, K. Han, and Y. Tian, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, pp. 806–815.

[57] Y. A. Malkov and D. A. Yashunin, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 824–836, 2018.

[58] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, "Large-scale image retrieval with attentive deep local features," 2016.

[59] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," 2020.

[60] M. Teichmann, A. Araujo, M. Zhu, and J. Sim, "Detect-to-retrieve: Efficient regional aggregation for image search," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[61] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19 516–19 547, 2021.

[62] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," in *CVPR*, 2017.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[64] A. Araujo and T. Weyand, "Google-landmarks: A new dataset and challenge for landmark recognition," Mar 2018. [Online]. Available: https://blog.research.google/2018/03/google-landmarks-new-dataset-and.html

[65] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.

[66] B. D. Ripley, *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[67] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S-PLUS*. Springer, 1999.

[68] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," 2018.

[69] G. Tolias, R. Sicre, and H. Jegou, "Particular object retrieval with integral max-pooling of cnn activations," in *ICLR*, 2016, pp. 1, 3, 4, 6, 10, 12.

[70] A. Babenko and V. Lempitsky, "Aggregating deep convolutional features for image retrieval," in *ICCV*, 2015, pp. 1, 3, 4, 6, 10, 12.

[71] H. Jegou and O. Chum, "Negative evidences and co-occurrences in image retrieval: The benefit of pca and whitening," in *ECCV*, 2012, pp. 2, 3, 10, 11.

[72] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "Cnn features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2014, pp. 806–813.

[73] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *European conference on computer vision*. Springer, 2016, pp. 241–257.

[74] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu, and L. Shao, "Normalization techniques in training dnns: Methodology, analysis and application," 2020.

[75] D. E. Knuth, *Sorting and Searching*. Addison-Wesley, 1997.

[76] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[77] ——, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008, pp. 1–8.

[78] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, and O. Chum, "Revisiting oxford and paris: Large-scale image retrieval benchmarking," in *CVPR*, 2018.

[79] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.

[80] E. Khvedchenya and H. Sahota, "Yolo-nas by deci achieves state-of-the-art performance on object detection using neural architecture search," Aug 2023. [Online]. Available: https://deci.ai/blog/yolo-nas-object-detection-foundation-model/