



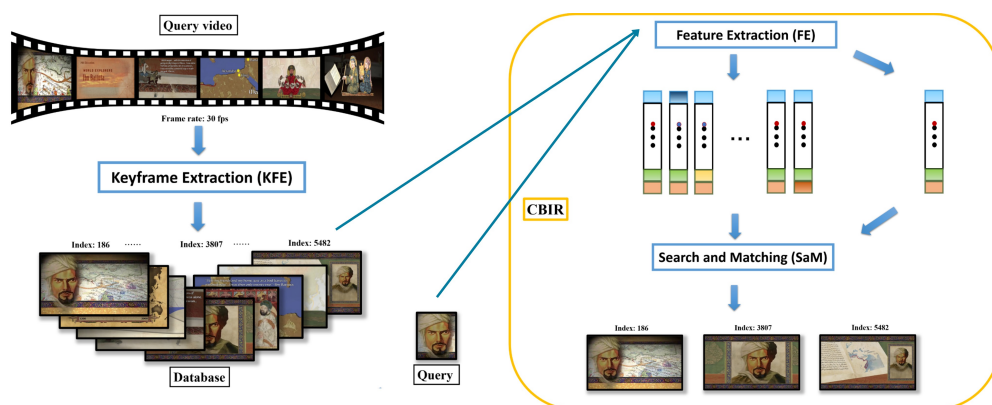
## M.Sc. Thesis

# Deep Learning-Empowered Content-Based Video Image Retrieval

Sinian Li B.Sc.

### Abstract

The advent of streaming and video has sparked a revolutionary shift in the presentation of materials across various fields, such as history and art. However, to leverage vast digital archives is time consuming and requires prolonged concentration. Content-based video image retrieval (CBVIR) is a technique designed to automatically retrieve visually similar images from a video based solely on the content of the query image. In this work, our primary objective is to develop and implement an efficient CBVIR system while ensuring minimal compromise on accuracy.





# Deep Learning-Empowered Content-Based Video Image Retrieval

---

THESIS

submitted in partial fulfillment of the  
requirements for the degree of

MASTER OF SCIENCE

in

ELECTRICAL ENGINEERING

by

Sinian Li B.Sc.  
born in Anhui, China

This work was performed in:

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



**Delft University of Technology**

Copyright © 2023 Circuits and Systems Group  
All rights reserved.

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 “**Deep Learning-Empowered Content-Based Video Image Retrieval**” by **Sinian Li B.Sc.** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: 22-08-2023

Chairman:

---

dr.ir. Justin Dauwels

Committee Members:

---

dr.ir. Merve Gürel

Advisor:

---

Yanbo Wang



# Abstract

---

The advent of streaming and video has sparked a revolutionary shift in the presentation of materials across various fields, such as history, art, and media. In this context, scholars are seeking efficient solutions to index, retrieve, and browse through digital content, searching for a specific instance. Unlike searching a specific instance in an image, searching in a video requires more than analyzing the visual features of an image and then comparing these features to a database, for it includes processing video sequences and retrieving video segments. A video can be viewed as a sequential arrangement of frames captured at a specific frame rate. The task of locating specific objects within a video can be likened to conducting retrieval within the set of frames that constitute the video. While a video usually has repetitive and redundant frames, making the processing rather inefficient.

Motivated by the urgent need and promising applications across diverse disciplines, we present a novel deep learning-empowered content-based video image retrieval (CBVIR) system with a strong emphasis on real-world applications. This system offers high efficiency and considerable accuracy, addressing the challenges associated with accessing and utilizing video materials effectively.

To address this, our initial approach revolves around the extraction of informative keyframes that effectively capture essential objects within the video. This process, known as Key Frame Extraction (KFE), enables us to distill the most crucial visual representations for further analysis. After the extraction of keyframes, the relatively smaller dataset allows for content-based image retrieval (CBIR) to be conducted, retrieving similar images from a database solely based on the content of the query image. However, capturing accurately necessitates the use of high-level representations, while processing with high efficiency requires simple or low-level interpretations of images. The existing research predominantly emphasizes accuracy and employs extensive convolutional neural networks to ensure high precision.

In this project, a wide range of methods are investigated and analyzed, including traditional representations, handcrafted feature extraction methods, and up-to-date machine learning-based image representations. Our contribution is striking a balance between high-level and low-level image representations for this task. Targeting efficiency improvement, enhanced color-based features together with dynamic clustering KFE module is proposed and implemented, achieving high efficiency ratio and satisfactory accuracy. While targeting accuracy, a traditional and deep learning-based hybrid feature is proposed, achieving valid efficiency ratio and highest accuracy. Overall, an automatic retrieving system requiring much less user engagement is provided, together with a system GUI prototype, which is available on <https://github.com/LotusCreme/CBVIR.git>, and a demo video can be found on <https://youtu.be/NiWZC823nag>.





# Acknowledgments

---

I am profoundly grateful for the invaluable support and guidance I received throughout my journey to completing this master's thesis (from November 2022 to August 2023).

First and foremost, I am deeply grateful to my supervisor, Dr. Ir. Justin Dauwels, for his unwavering guidance, expertise, and patience. His insightful feedback and weekly constructive discussions shaped the direction and quality of this thesis. And I extend my appreciation to my daily co-supervisor Yanbo Wang, who has influenced my academic journey and future career in meaningful ways. Also, I would like to express my gratitude to the SPS faculty for fostering an intellectually stimulating environment and providing valuable opportunities that inspire our academic growth.

I would like to extend my gratitude to Dr. Merve Gürel, for graciously accepting the invitation to join the thesis committee. And special thanks to Dr. Andrea Nanetti for generously providing us with a video database in historical research that greatly facilitates our exploration of the research topic.

To my beloved family, who stood by me with everlasting love, encouragement, and understanding, I owe immeasurable gratitude. Your steadfast belief in me fueled my determination to overcome challenges. Your financial and emotional support made this work possible and my journey to the Netherlands colorful and fruitful.

My sincere thanks go to my colleague Doruk Barokas Profeta and my friends Yuhang Yao, May Wu, Kangming Mao, and Yinglu Tang for their camaraderie, support, and occasional laughter that provided much-needed balance during this intense academic pursuit.

Over the past two years, I've been through a series of challenges, each of which tested my mettle. Amidst the journey of my thesis, a persistent cloud of self-doubt often loomed over me, casting shadows of uncertainty on my abilities and dedication. But luckily, God brought into my life remarkable individuals, who are kind, dynamic, and brightening. These luminous souls, with their infectious positivity, shined lights on my gloomy days. It was from their wisdom and encouragement that I drew the strength to not only survive but thrive and evolve.

Lastly, a quote from the visionary Steve Jobs, "Stay hungry, stay foolish," has been guiding me since high school. Over time, I've come to understand that the essence of this mantra lies in the profound courage to embrace the unknown, to allow myself to be vulnerable, and to cultivate a relentless hunger for knowledge. As we are stepping into our future careers, I hope we could no longer fear the notion of appearing foolish, for true foolishness lies in the inhibition of growth and learning. And we can remain hungry for the acquisition of knowledge and skills that would fuel our journey of life.

Sinian Li B.Sc.  
Delft, The Netherlands  
22-08-2023



# Contents

---

<b>Abstract</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Background . . . . .	1
1.2 Pipeline . . . . .	2
1.3 Problem Statement . . . . .	3
1.4 Outline . . . . .	4
<b>2 Related works</b>	<b>7</b>
2.1 Content-based Video Image Retrieval (CBVIR) . . . . .	7
2.2 Key Frame Extraction (KFE) . . . . .	8
2.3 Content-based Image Retrieval (CBIR) . . . . .	9
<b>3 Methods</b>	<b>11</b>
3.1 Image Representation . . . . .	12
3.1.1 Color Histogram (CH) . . . . .	12
3.1.2 Histogram of Oriented Gradients (HOG) . . . . .	17
3.1.3 Scale-Invariant Feature Transform (SIFT) . . . . .	20
3.1.4 Deep learning-Based Features . . . . .	22
3.1.5 Hybrid Feature (HF) . . . . .	28
3.2 Clustering Methods . . . . .	30
3.2.1 K-means Clustering . . . . .	30
3.2.2 Dynamic Clustering . . . . .	32
<b>4 Experiments and Results</b>	<b>35</b>
4.1 Dataset . . . . .	35
4.1.1 Properties . . . . .	35
4.1.2 Content . . . . .	35
4.1.3 Groundtruth . . . . .	36
4.2 Metrics . . . . .	37
4.2.1 Accuracy . . . . .	38
4.2.2 Redundancy . . . . .	38
4.2.3 Efficiency Ratio . . . . .	40
4.2.4 mean Average Precision (mAP) . . . . .	41
4.3 Results . . . . .	42
4.3.1 Experiments on KFE . . . . .	42
4.3.2 Image representation methods comparison on KFE . . . . .	48
4.3.3 Experiments on the overall task . . . . .	49
4.3.4 Discussion . . . . .	52
4.4 System Prototype . . . . .	52

<b>5</b>	<b>Conclusions</b>	<b>57</b>
5.1	Conclusion . . . . .	57
5.2	Future directions . . . . .	57

# List of Figures

---

1.1	Two-stage CBVIR pipeline . . . . .	3
3.1	Illustration of HSV color model . . . . .	12
3.2	Illustration of RGB color model . . . . .	13
3.3	Illustration of RGB color histograms . . . . .	13
3.4	Example patterns having overall similar color histograms, but different spatial distributions (adapted from [1]) . . . . .	14
3.5	The illustration of including spatial information . . . . .	15
3.6	The pipeline of color-based feature matrix formation . . . . .	16
3.7	Illustration of HOG results . . . . .	19
3.8	Illustration of searching for local maxima and minima of the DoG images (adapted from [2]) . . . . .	20
3.9	SIFT detected keypoints on example images . . . . .	21
3.10	SIFT keypoints matching results on example images . . . . .	22
3.11	Illustration of VGGNet structure (adapted from [3,4]) . . . . .	23
3.12	Illustration of VGG19 structure . . . . .	24
3.13	Example feature maps from different VGG16 layers . . . . .	25
3.14	Illustration of a buiding block for residual learning (adapted from [5]) . . . . .	25
3.15	Network architectures comparison for ImageNet (adapted from [5]) . . . . .	27
3.16	Example feature maps from different ResNet18 layers . . . . .	28
3.17	Illustration of hybrid feature structure . . . . .	29
3.18	A demonstration of KFE results . . . . .	30
3.19	VSUMM approach pipeline (adapted from [6]) . . . . .	31
3.20	Illustration of the process clustering upcoming frame into one cluster and the process establishing a new cluster . . . . .	33
4.1	Dataset example (Video-Queries Pair) . . . . .	36
4.2	KFE results - example one . . . . .	39
4.3	KFE results - example two . . . . .	40
4.4	Average Precision Example . . . . .	42
4.5	The pipeline of black bar removal technique . . . . .	43
4.6	Plots of ResNet variations' Accuracy, Redundancy and mean Efficiency Ratio performances in KFE module . . . . .	45
4.7	Illustration of K-means clustering . . . . .	47
4.8	Illustration of dynamic clustering . . . . .	47
4.9	Illustration of recycling scheme . . . . .	50
4.10	Screenshot of the graphical user interface in home page . . . . .	53
4.11	Screenshot of the start page that requests upload and processes operations from users . . . . .	53

4.12	Illustration of how the results are presented in application window and saved keyframe set together with the final result list in txt file in local computer (Example of the query video of <Ibn Battuta PBS World Explorers> and the query image of index 1) . . . . .	54
4.13	Screenshot of result page (Example of the query video of <Zhenghe facts and his accomplishments, the untold story> and the query image of index 411) . . . . .	54
4.14	Screenshot of result page (Example of the query video of <Zhenghe facts and his accomplishments, the untold story> and the query image of index 412) . . . . .	55

# List of Tables

---

3.1	Properties comparison of models . . . . .	26
4.1	Example groundtruth for a video . . . . .	37
4.2	Black bar removal experiment setting . . . . .	43
4.3	The impact of black bar removal on traditional methods performances .	43
4.4	Experiment setting for different ResNet variations . . . . .	44
4.5	Performance comparison among different ResNet variations in KFE module	45
4.6	Experimental setting in testing different clustering methods for KFE task	46
4.7	Performance comparison on two clustering methods: K-means clustering and dynamic clustering (ranked by redundancy) . . . . .	46
4.8	Experimental setting in testing different feature extraction methods for KFE . . . . .	48
4.9	Performance comparison on different image representation methods in KFE task . . . . .	49
4.10	Experimental setting in testing different feature extraction methods for the whole system . . . . .	50
4.11	Overall performance comparison on different combinations of feature methods . . . . .	51





This project aims to develop an efficient and reliable content-based video image retrieval system to automatically locate query images within videos. While the application potential is diverse, our primary focus lies on its utility in historical research. The system is a part of the project known as "Engineering Historical Memory" (EHM), led by Dr. Andrea Nanetti, aiming to support ongoing historical research. The objective of this project is to harness emerging digital techniques like artificial intelligence to aggregate and disseminate historical knowledge [7]. Further elaboration can be found in Section 1.1: Background.

The proposed system's applicability extends beyond a specific application. It is well-suited for scenarios requiring efficient navigation through video databases to pinpoint specific patterns. Moreover, the system could be useful for content creators, video editors, news generators, and criminal investigators. These diverse users can leverage it to select, highlight, and analyze raw video content, swiftly access visual information from lengthy recordings, and efficiently identify subjects of interest in CCTV footage.

Our approach involves two distinct schemes: the division of the task into two stages- Key Frame Extraction and Content-based Image Retrieval- and strategic utilize different image processing techniques in each stage. The second scheme is recycling image representations across both stages. Section 1.2: Pipeline delves into the rationale behind this approach and explains the functions of each module.

For clarity, a comprehensive problem statement and objective statement are provided in Section 1.3: Problem Statement. Furthermore, the structural layout of this thesis is outlined in Section 1.4: Outline.

This endeavor seeks to blend academic rigor with practical utility, addressing a range of applications while anchored in the pursuit of enhancing historical research through cutting-edge digital methodologies.

## 1.1 Background

The advent of streaming and video has sparked a revolutionary shift in the presentation of materials across various fields, such as history and art. In this context, scholars are increasingly seeking efficient solutions to index, retrieve, and browse digital historical content. These solutions are crucial for enabling researchers to leverage vast digital archives without expending excessive time and effort on filtering out irrelevant information.

Recent advancements in deep learning methodologies have demonstrated their pivotal role in enhancing the image processing and retrieving process [8]. Additionally, the combination of deep learning with traditional media and signal processing methods can yield compelling outcomes [9].

Motivated by the urgent need and promising applications across diverse disciplines, we present a novel deep learning-empowered content-based video image retrieval system. This system offers high efficiency and considerable accuracy, addressing the challenges associated with accessing and utilizing digital historical materials effectively.

## 1.2 Pipeline

To achieve this goal, we propose decomposing the task into two stages. Figure 1.1 is the pipeline illustration. The initial stage is denoted as Key Frame Extraction (KFE) and depicted by the first blue box. This stage focuses on removing consecutive and redundant frames from the input query video, while simultaneously identifies frames of importance. Subsequently, the second stage, represented by the orange box in Figure 1.1 and referred to as Content-Based Image Retrieval (CBIR), consists of Feature Extraction (FE) and Search and Match (SaM). Within this stage, the system extracts features from the query image as well as the newly formed keyframe database. Moreover, it employs a certain Approximate Nearest Neighbor (ANN) search to match these features, ultimately identifying the most suitable candidates that correspond closely to the query image.

A video can be considered a collection of frames at a specific frame rate [10]. Locating specific objects within a video is equivalent to conducting retrieval within the set of frames comprising the video.

Typically, videos have a frame rate of 30 frames per second, meaning a five-minute video encompasses 9000 frames with significant content repetition, leading to wasted efforts. This redundancy poses challenges in finding objects within such datasets. To address this, our initial approach is KFE, which involves selecting and displaying informative keyframes that capture essential objects within the video. The elimination of redundant frames thereby enhances the efficiency and accuracy of subsequent processing steps.

The fastest way to remove redundancy in a video is uniform downsampling, but uniformly extracting keyframes is impractical due to varying shot durations, which makes discerning crucial and redundant information impossible, just like random selection. Consequently, effective non-uniform extraction is crucial, necessitating the selection of an appropriate image feature representation.

Once keyframes are extracted, content-based image retrieval (CBIR) can be performed on this relatively smaller dataset. The CBIR is designed to retrieve relevant images from a database based only on the visual content of the query image, without reliance on text or keywords. However, existing research predominantly focuses on accuracy rates, often utilizing extensive convolutional neural networks to ensure high accuracy. In the context of this task, where the dataset size is no longer excessive, exploring image interpretations that yield enhanced gains becomes an essential focus of this study.

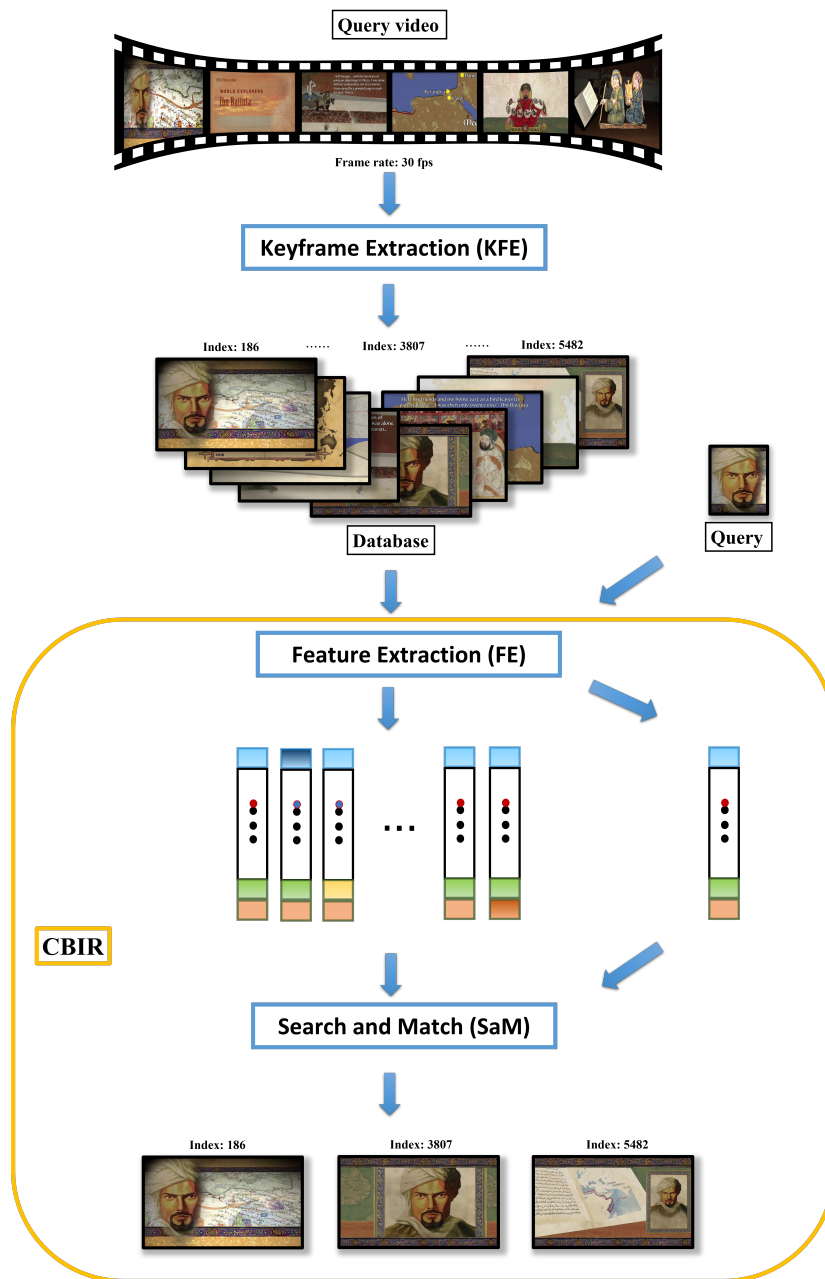


Figure 1.1: Two-stage CBVIR pipeline

### 1.3 Problem Statement

This study focuses on developing an efficient content-based video image retrieval (CBVIR) system with a strong emphasis on real-world applications. Efficiency and accuracy are the primary metrics of concern due to their significant impact on system performance. The efficient retrieval of data plays a crucial role in enabling researchers to access relevant information quickly and conveniently. Achieving accuracy in media data extraction necessitates the use of high-level representations, while high efficiency

and speed require low-level interpretations. **The core focus of this work is to strike a balance between low-level features, which enhance efficiency, and high-level features, which bolster accuracy.**

The proposed system is modular, thus flexible, allowing for the enhancement of efficiency through improvements in its two stages. But It is crucial to emphasize that the overall accuracy and efficiency of the system are intricately intertwined with both modules. For instance, if the Key Frame Extraction (KFE) module operates with commendable efficiency but unfortunately omits crucial keyframes associated with a particular query image, the Content-Based Image Retrieval (CBIR) module becomes ineffective in aiding the retrieval process, resulting in a decrease in overall accuracy. Conversely, if the KFE module selects an excessive number of redundant frames, it negatively affects the subsequent feature extraction and search processes within the CBIR module. Therefore, we have established three specific quantitative goals for the KFE module, CBIR module, and the overall system.

1. **The Key Frame Extraction (KFE)** module should demonstrate a significant increase in efficiency, aiming for a processing speed that is at least 20 times faster than the original duration of the video. Simultaneously, it should maintain a high accuracy level of 0.95.
2. **The Content-Based Image Retrieval (CBIR)** module should exhibit notable efficiency improvements, targeting a processing speed that is at least 20 times faster than the original duration of the video. Additionally, it should achieve a mean average precision of 0.9, ensuring reliable and precise image retrieval results.
3. **The overall system** should strive for an efficiency ratio of 10 times faster than the original duration of the video. It should successfully combine the enhanced efficiency of both the KFE and CBIR modules. Moreover, the system should achieve a mean average precision of 0.9.

More details of those metrics can be found in Chapter 4.

## 1.4 Outline

The outline of this thesis is as follows:

- Chapter 2 delves into significant research and methodologies related to video summarization and content-based image retrieval.
- Chapter 3 presents the details of image representation and clustering methods that are implemented or tested in this project. Also, this chapter thoroughly examines their respective advantages and drawbacks.
- Chapter 4 first introduces the dataset we utilized for testing our methodologies, and subsequently outlines the metrics used to evaluate the methods throughout the experiments. Then, the results are presented in an order of sub-topics under Key Frame Extraction (KFE) module, the overall KFE module, the connected system's performance and the GUI prototype.

- Chapter 5 serves as the last chapter encapsulates the outcomes of the research and proposes potential directions for future exploration.



In this chapter, related works in Content-based Video Image Retrieval (CBVIR) will be discussed. And particularly, image feature extraction methods along with their respective application scenarios will also be included.

## 2.1 Content-based Video Image Retrieval (CBVIR)

Generally, this topic is a relatively new topic compared to Content-based Image Retrieval (CBIR) from large databases for most of the recent advancements in artificial intelligence have been focusing on enhancing real-time CBIR or accurate intelligent image search capabilities [11].

VISIONE is a tool for large-scale video search, developed for Video Browser Showdown 2019 challenge [12]. This paper mentioned that three types of queries are supported in this system. Keyword-based search allows users to search for specific video segments by textual keywords using the Caffe framework [13]. The system annotates images using WorkNet [14]. It extracts scene attributes and utilizes an automatic annotation system for untagged images. It also includes object location search, which can sketch simple bounding boxes to indicate spatial locations of objects in the scene, by integrating the real-time object detection algorithm YOLOv3 [15]. Object information is indexed using a text-based representation. And for visual search, it extracts visual features using the Regional Maximum Activations of Convolutions (R-MAC) [16] and transforms visual features into textual representations for text search engines. This proposed system highly relies on text-based indexing which may lose visual nuances and potentially struggle with complex content understanding and indexing. Because it is not open-sourced, all the functions have not been tested and validated yet.

Two reviews of CBVIR [17,18] pointed out that dealing with excessive video frames requires shot boundary detection and key frame selection. As for image retrieval, there are two categories: query-by-text (QbT) and query-by-example (QbE) [17]. QbT corresponds to Annotation-based Image Retrieval (ABIR), which is sophisticated and even extended to real-time re-ranking to reduce ambiguity and noisy results [19]. QbE means the search is based on image content, which is also known as Content-based Image Retrieval (CBIR). The subsequent and main discussions are around the progress in image retrieval instead of video content retrieval.

As for the adaptable framework for content-based visual media retrieval, a framework leverages Convolutional Neural Networks (2D CNN), 3D Convolutions (3D CNN), and Long Short Term Memory networks (LSTMs) [20] to process images and videos for content-based retrieval [21]. Its recurrent convolutional architecture includes LSTM processing to generate a final feature representation for the video. This framework works effectively for both images and videos for retrieval. Their 3D model is good

at retrieving seen data while 2D model is better at unseen data. It enhances video comprehension and turning points identification by using LSTM. This network includes sequence information and relationship of scenes of the video [22], but the whole pipeline aims to process video into a single feature encapsulating the essence of the video. This causes the lack of temporal information and it is at the expense of efficiency.

Generally, video image retrieval topic exhibits a scarcity of pertinent literature within the academic domain. In the next section, video summarization and image feature extraction methods will be discussed.

## 2.2 Key Frame Extraction (KFE)

As discussed in Chapter 1, our proposed implementation of this application involves two distinct stages: Key Frame Extraction (KFE) and Content-Based Image Retrieval (CBIR). Irrespective of the specific stage, the primary focus for enhancing efficiency and ensuring accuracy lies in employing an appropriate image feature description method.

The existing literature primarily focuses on KFE methods within the realm of video summarization. Video summarization aims to generate concise synopses of lengthy videos while preserving crucial characters and main storylines. Traditional approaches encompass techniques such as determining optical flow based on brightness patterns [23], object-based inter-frame change detection [24], and utilizing color histogram features [25]. Audiovisual features are used to describe the characteristics of the shots and utilize Support Vector Machines (SVM) to select relevant shots [26]. While these methods effectively extract keyframes, their redundancy performance is compromised.

A technique named VSUMM is designed to generate condensed video summaries. This approach involves extracting HSV color features from frames and employing the k-means clustering algorithm [6]. In a similar vein, researchers exploit the variance curve of dynamic color histograms to identify gradual shot transitions, leading to the extraction of a set of keyframes. They also utilize a rapid wavelet histogram technique via optimized k-means to create another set of keyframes. These two sets are then combined using mutual information to produce the final selection of keyframes [27]. Conversely, instead of dealing with visual features, a framework first trains the network to link the visual information to textual inputs and then takes a textual query as input and generates a corresponding keyframe set [28]. Works leveraging deep learning models like Graph Convolutional Networks (GCNs) [29], Bi-directional Long Short-Term Memory (BiLSTM) fuse multi-modality to sort the scene relations and select the candidate shots for the essence creation [9, 30]. These pipelines offers valuable insights for conducting our project but they are not perfectly aligned with our task requirements and structures. Instead, decomposing the video into independent frames as VSUMM [6] pipeline is more economical in this task.



## 2.3 Content-based Image Retrieval (CBIR)

The historical researchers' pain point in searching visual materials is that the database is not thoroughly annotated with textual descriptions, and even if it is labeled or fully annotated in a certain language, cross-cultural and cross-language research still face text-based retrieval limitations. Moreover, researchers often lack textual clues, such as keywords or detailed descriptions. Different from query-by-text, CBIR requires no textual information, treating all visual materials as feature vectors, and by comparing features' similarity, it can provide the most similar candidates.

Earlier progress in image feature analysis techniques like handcrafted features Scale-Invariant Feature Transform (SIFT) [2], Speeded Up Robust Features (SURF) [31], Oriented fast and Rotated BRIEF (ORB) [32], and Hashing value [33] replaced traditional methods and were prevalent in image representation until the advent revolutionizing deep convolutional neural networks (DCNN) [11], including AlexNet [34] and ensuing networks [4, 5, 35, 36].

VGGNet and ResNet are outstanding CNNs regarding accuracy and efficiency. More details and discussions can be found in Chapter 3. Most studies tend to favor the approach of individually mapping vectors from a convolutional layer as local features [37] and subsequently combining them into a global feature [16, 38].

R-MAC generates a comprehensive image description by combining CNN activation characteristics from various spatial regions through a specific aggregation process. [16], Cross-dimensional Weighting (CroW) descriptor is an efficient non-parametric weighting and aggregation scheme, aiming to combine and summarize the information contained within convolutional features to create a more condensed and representative global feature [39]. Sum-Pooled Convolutional (SPoC) descriptor (based on simple sum-pooling aggregation) leverages preprocessed Gaussian centers to aggregate convolutional features, deriving concise global descriptors. It yields a significant performance enhancement compared to previous global image descriptors commonly employed in image retrieval tasks [38]. Additionally, deep learning methods are revolutionizing this field with much more advanced features that could adjust the attention and highlight the most relevant regions within the extracted features [40, 41].

Obviously, researchers have increasingly turned to these more advanced features for image retrieval tasks. However, it's worth noting that though these techniques are more effective compared to low-level feature extraction approaches, they also require higher computational resources and are less efficient.

Both KFE and CBIR need the implementation of the same processing scheme: image representation where each frame or image is represented by a vector using a particular extraction method, and representation comparison, involving comparing the similarity of images with a certain comparison measure. However, they exhibit distinct levels of sensitivity and accuracy requirements. For KFE, the focus is on consecutive frames, which often exhibit similar characteristics, whether in terms of background or foreground objects. Regardless of a hard or soft transition, intermediary frames experience a phase of blurriness or a sudden and substantial change in optical features, both of which require only a low feature "resolution". On the other hand, CBIR's similarity comparison centers around a given query image, and all comparison objects

receive equal consideration. The combination of adjacent frame features does not yield any attentional advantage. Therefore, it needs high-level features instead of traditional methods.

The aforementioned works are groundbreaking researches that are worth further investigation and experiment. In the next chapter, different levels of image representation methods will be thoroughly analyzed.

In Chapter 2, we thoroughly examined and analyzed various image representation methods, along with their respective application scenarios. However, it is worth noting that there is a scarcity of research or discussion specifically addressing efficient video frame retrieval tasks. Consequently, we were unable to reach a definitive conclusion regarding the optimal method or scheme for such tasks. In light of this, rather than exclusively focusing on cutting-edge methods, we broadened our scope to include less current categories.

In this chapter, we will delve into the state-of-the-art image representation methods from different categories, both traditional handcrafted features and up-to-date deep learning-based methods. Our objective is to explore their potential, drawbacks, and suitability within both the Key Frame Extraction (KFE) and Content-Based Image Retrieval (CBIR) modules. In KFE module, extracted features later serve as cluster references for the system to take key frames from the video. In CBIR module, extracted features will be compared using a certain measure to retrieve all the relevant candidates from the keyframes. By undertaking this comprehensive examination, we aim to analyze the strengths and weaknesses of these methods and their applicability to our specific research goals.

Among the methods for image representation to be examined are the following:

- Color-histogram
- Histogram of gradients
- Scale-Invariant Feature Transform (SIFT)
- VGGNet-based
- ResNet-based
- Hybrid feature

The methods for clustering features to be discussed are the following:

- K-means clustering
- Dynamic clustering

## 3.1 Image Representation

### 3.1.1 Color Histogram (CH)

A color histogram is a color-based image visual representation that effectively captures and illustrates the distribution of colors within an image. Particularly for digital images, the color histogram quantifies the number of pixels associated with specific colors across the entire color space of the image. Statistically, the color histogram encapsulates the joint probability of intensities for each of the three color channels, thereby presenting a comprehensive portrayal of the global color distribution in the image [42].

While a color histogram can be constructed for any color space, it is commonly employed in the context of three-dimensional color spaces such as RGB (Red-Green-Blue color model) or HSV (Hue-Saturation-Value color model). In this work, for key frame extraction part, we mainly use RGB color model, for HSV separating color information from brightness information also makes computation less efficient, while the gain is not outstanding. For comparison, we include the VSUMM scheme proposed in [6]. Next, more details of this scheme and RGB color histogram calculation will be discussed.

#### 3.1.1.1 Video SUMMarization (VSUMM)

Video SUMMarization (VSUMM) approach aims to automatically create a summary for a given video. This approach is based on HSV color histogram feature extraction from video frames and K-means clustering algorithm [6].

In the image representation step, the processing of the color space is HSV (Hue, Saturation, Value) model as shown in Figure 3.1.

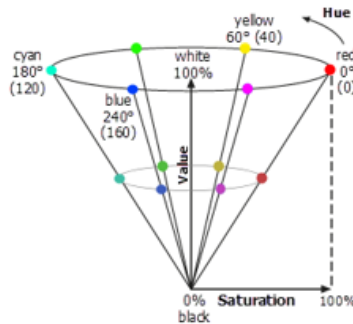


Figure 3.1: Illustration of HSV color model

It computes the dominant spectral component color and quantizes the color histogram to 16 bins. Then it calculates the histogram of each bin unit, connects the histograms of all bin units in the same image block to form the histogram feature of the image block, and normalizes it. Finally, the feature descriptions of all image blocks in the image are connected to obtain the histogram feature of the entire image.

Its clustering method will be discussed in Section 3.2.

### 3.1.1.2 Color-based feature vector formation

Assuming the RGB color space, as shown in Figure 3.2 contains  $L$  color bins.

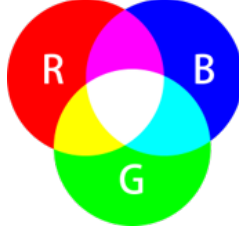


Figure 3.2: Illustration of RGB color model

The proportion of pixels in the  $k^{th}$  color bin of an image with  $N$  pixels can be presented as Equation 3.1:

$$h(k) = \frac{\eta(k)}{N}, \quad k \in \{1, 2, \dots, L\}, \quad (3.1)$$

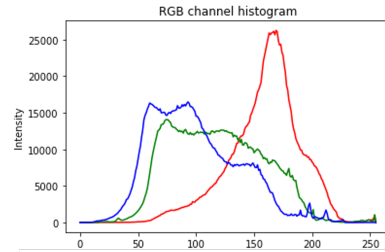
where  $\eta(k)$  is the total number of pixels in the  $k^{th}$  color bin. The full color histogram of the entire image  $I$  can be expressed as:

$$\mathbf{H}(I) = [h(1) h(2) \dots h(L)]. \quad (3.2)$$

Figure 3.3 is the illustration of two example images with their color histogram plots.



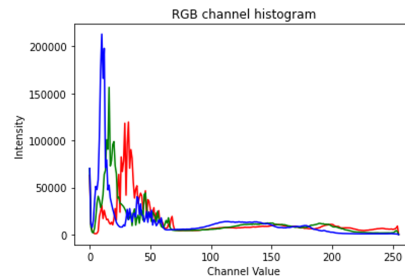
(a) Example image - mural



(b) Full color histogram illustration - 1



(c) Example image - religious figure



(d) Full color histogram illustration - 2

Figure 3.3: Illustration of RGB color histograms

The calculation of color histograms is simple and fast, and the color information it keeps can be applied to simple image retrieval and similarity check tasks, but it has

two major limitations. Firstly, it is inherently vulnerable to slight brightness changes or quantization errors. This sensitivity to bin allocation can lead to misrepresentations and inaccurate comparisons [43]. Specifically, when colors are visually similar but actually distinct, there is a possibility that they will be perceived as identical if they happen to be assigned to the same histogram bin. Conversely, colors that belong to different bins, even if they share a significant visual similarity, will be regarded as completely different. And the other limitation is that color histogram is proven to be less efficacious on large databases [44]. Because a color histogram primarily emphasizes the distribution of different color types within an image, disregarding their spatial arrangement. Consequently, images that appear distinct can exhibit similar color histograms. Figure 3.4 This issue becomes particularly pronounced when dealing with extensive image databases, where a considerable number of images may share similar color histograms which exacerbates the problem.

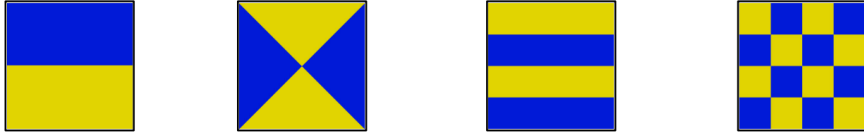


Figure 3.4: Example patterns having overall similar color histograms, but different spatial distributions (adapted from [1])

Recognizing these constraints, we can first increase the number of bins to mitigate the impact of quantization error. Additionally, we can still incorporate spatial information alongside color distributions by dividing the original image into blocks and flattening the features after embedding each block respectively.

Given that the color span is divided into  $L$  color bins, the color histogram of a block located at  $p^{th}$  row and  $q^{th}$  column with  $N_{(p,q)}$  pixels can be presented as Equation 3.3:

$$h_{(p,q)}(k) = \frac{\eta_{(p,q)}(k)}{N_{(p,q)}}, \quad k \in \{1, 2, \dots, L\}, \quad (3.3)$$

where  $\eta_{(p,q)}(k)$  is the  $(p, q)^{th}$ ,  $p \in \{1, 2, \dots, P\}$ ,  $q \in \{1, 2, \dots, Q\}$  block total number of pixels in the  $k^{th}$  color bin.

Therefore, the image feature matrix  $H(I)$  could be expressed accordingly:

$$H(I) = \begin{bmatrix} h_{(1,1)}(k) & h_{(1,2)}(k) & \cdots & h_{(1,Q)}(k) \\ h_{(2,1)}(k) & h_{(2,2)}(k) & \cdots & h_{(2,Q)}(k) \\ \vdots & \vdots & \ddots & \vdots \\ h_{(P,1)}(k) & h_{(P,2)}(k) & \cdots & h_{(P,Q)}(k) \end{bmatrix}. \quad (3.4)$$

For a more compact expression that can be easily used in later feature comparison, we can vectorize  $H(I)$  into a feature vector  $\mathbf{h}(I)$ , as shown in Equation 3.5:

$$\mathbf{h}(I) = \text{vec}(\mathbf{H}(I)), \quad (3.5)$$

where  $\text{vec}$  represents the vectorization function [45]:

$$\text{vec}(\mathbf{A}) = [a_{1,1}, \dots, a_{m,1}, a_{1,2}, \dots, a_{m,2}, \dots, a_{1,n}, \dots, a_{m,n}]^T,$$

which stacks the columns of a  $m \times n$  matrix  $\mathbf{A}$ .

Figure 3.5 shows the pipeline of how the enhancements are implemented. The first image is a frame read from a historical video, and it is partitioned into a grid of  $3 \times 3$ , resulting in a total of 9 blocks. After embedding each block using color histogram, this feature matrix is vectorized into a feature vector that can represent the whole image without losing all spatial information.

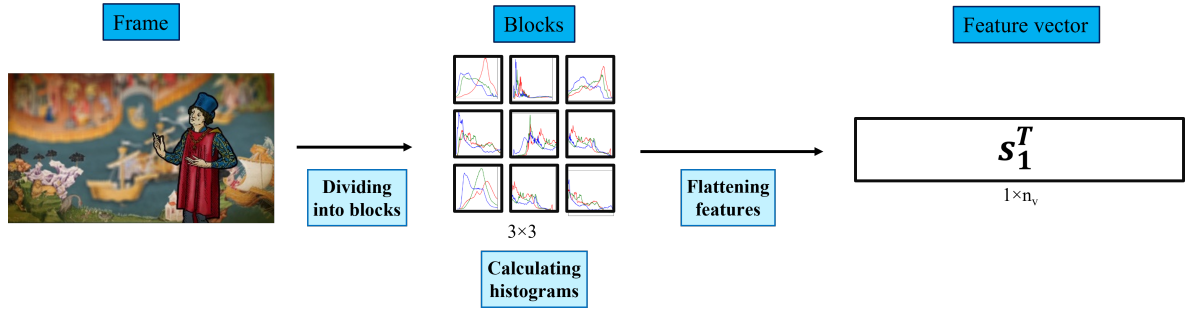


Figure 3.5: The illustration of including spatial information

### 3.1.1.3 Color-based feature matrix formation

Moreover, a video consists of many frames, which can be denoted as  $\mathbf{I}$ . Therefore, we can express the color-based feature matrix as  $\mathbf{H}(\mathbf{I})$ , and

$$\mathbf{H}(\mathbf{I}) = [\mathbf{h}(\mathbf{I}_1) \quad \mathbf{h}(\mathbf{I}_2) \quad \cdots \quad \mathbf{h}(\mathbf{I}_m)], \quad m \in \{1, 2, \dots, M\}, \quad (3.6)$$

where  $M$  is the number of frames contained in a certain video.

### 3.1.1.4 Singular Value Decomposition (SVD)

We have introduced the color-based image representation of a single frame and a single video in the previous section. The raw color histogram representation is a redundant form with a large dimension. Reducing the dimensionality of these features alleviates the computational burden in subsequent steps. In this section, we will introduce a Principle Component Analysis(PCA) method Singular Value Decomposition(SVD) that can reduce the dimension of our raw color-histogram features. This technique has been used in motion video summarization tasks with the user-specified length [46].

The fundamental concept behind Principal Component Analysis (PCA) is to transform a set of features from a higher-dimensional space ( $n$ -dimensional) into a lower-dimensional space ( $k$ -dimensional). This new space, consisting of  $k$  orthogonal features, is referred to as the principal components [47]. These principal components are essentially a compressed representation of the original  $n$ -dimensional features [47]. In practical applications, the matrices we decompose are often non-square, and the same

holds true for our application. Singular Value Decomposition (SVD) [48] is a versatile decomposition method that can be applied to any matrix. Given an  $n \times m (n \geq m)$  matrix  $A$ , there always exists a singular value decomposition:

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (3.7)$$

where  $\mathbf{U}$  is an  $n \times n$  orthogonal matrix, and the orthogonal vectors in  $\mathbf{U}$  are called left singular vectors.  $\mathbf{\Sigma}$  is a diagonal matrix of  $n \times m$ , the elements outside the diagonal are all 0, and the elements on the diagonal are called singular values, and the singular values are generally arranged from large to small.  $\mathbf{V}^T$  is an  $m \times m$  orthogonal matrix, which is the transpose matrix of  $\mathbf{V}$ , and the orthogonal vector in it is called the right singular value vector [49].

After performing SVD on the feature matrix  $\mathbf{H}(\mathbf{I})$ , we truncate the right singular vectors to take only the top 60 of them. Singular Value Decomposition (SVD) holds an important implication: the larger the singular value, the more information it represents, while smaller singular values can be considered negligible. In practice, retaining the first  $k$  singular values aims to find a low-rank approximation of an arbitrary matrix  $A_{n \times n}$ . This approximation seeks to closely resemble the original matrix by capturing its essential characteristics. Figure 3.6 is an illustration of color-based image features formation pipeline.

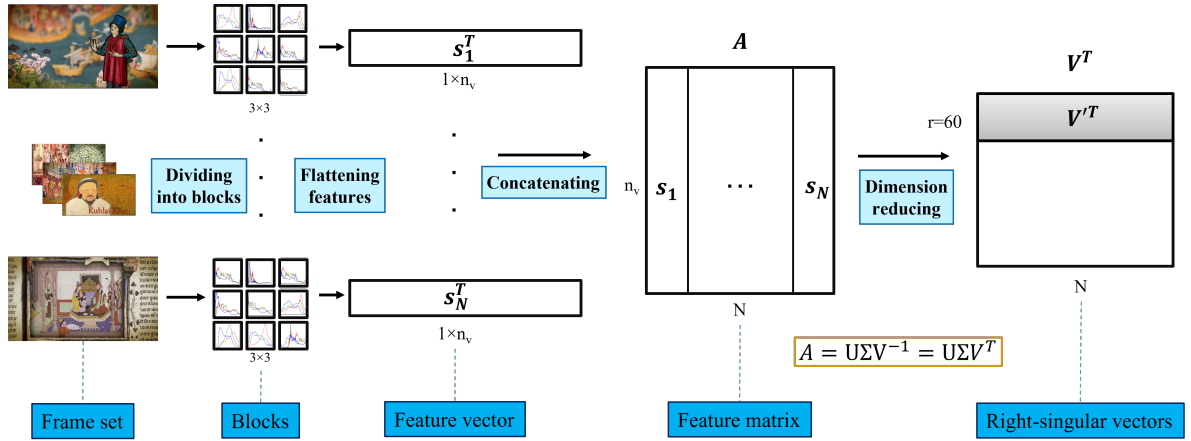


Figure 3.6: The pipeline of color-based feature matrix formation

There are various reasons to utilize a low-rank matrix for approximation. For our case, in an image, only the first few dozen singular values along with their corresponding orthonormal basis vectors are retained. This can dramatically reduce the storage space required and reduces future comparison burdens between feature vectors. Additionally, for many other image features, low-rank approximations can be used to remove noise. Small singular values, often associated with noise introduced during image sampling, can be discarded to enhance image quality [48].

Another improvement falls in the matrix formation before SVD. The feature matrix we are working with before SVD remains large. As the pipeline in Figure 3.6 shows, the number of columns, denoted as  $N$ , represents the number of frames we are handling. For instance, a 4-minute video contains approximately 7000 frames. SVD



algorithm based on Lanczos method has the computation complexity proportional to  $\mathcal{O}(\min\{m^2n, mn^2\})$  [50]. Therefore, further improvement is required in this aspect. The most intuitive way is to segment the feature matrix before SVD. According to the SubMatrix Selection Singular Value Decomposition (SMSSVD) [51], no tuning is required when applying SMSSVD to a dataset and different dimensions can be interpreted separately from each other, which means the orthogonality holds when we implement submatrix SVD.

### 3.1.1.5 Color histogram analysis

By improving in these aspects, the spatial color information is partially preserved with the order of blocks. Introducing block-wise color histograms incurs minimal costs in relation to the gain it can offer. The dimension of the feature matrix is reduced by SMSSVD, which will alleviate the burden of later similarity check and processing. These enhancements contribute to more efficiency without losing comprehensiveness and accuracy in image comparisons.

In essence, after a few improvements, the color histogram keeps valuable insights that facilitate subsequent assessments of image similarity. This attribute renders the color histogram a practical tool for diverse applications in image processing and analysis.

## 3.1.2 Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients (HOG) is a shape-based feature descriptor. Widely used in image processing tasks like human detection, HOG also performs well in object detection and recognition tasks [52, 53]. It captures the distribution of gradient orientations in an image, both the magnitudes and orientations of the changes in pixel intensities. The information it can provide is significant, for it depicts the local structure and outlines the edges of an image.

Though the idea of HOG is not complicated, the implementation is a bit complex and computationally demanding. It first converts the input color image into grayscale to simplify the future calculation for color intensities that are less relevant in gradient analysis except more dimensions are included. Next, the gradient magnitude and orientation for each pixel in the image are calculated, usually in the  $x$  and  $y$  directions.

### 3.1.2.1 HOG Feature Calculation

Say we are given an grayscale image  $\mathbf{I} \in [0 : 255]^{128 \times 64}$ , ( $\mathbf{I}(r, c)$  denotes the  $(r, c)^{th}$  entry), one can follow these steps to obtain the HOG feature of the image.

**Step 1):** Calculate the gradient matrices

$$\mathbf{G}_x(r, c) = \mathbf{I}(r, c + 1) - \mathbf{I}(r, c - 1), \quad (3.8)$$

$$\mathbf{G}_y(r, c) = \mathbf{I}(r - 1, c) - \mathbf{I}(r + 1, c). \quad (3.9)$$

**Step 2):** Calculate the magnitude matrix  $\mathbf{M}$  and the angle matrix  $\mathbf{A}$  as,

$$\mathbf{M}(r, c) = \sqrt{\mathbf{G}_x(r, c)^2 + \mathbf{G}_y(r, c)^2}, \quad (3.10)$$

$$\mathbf{A}(r, c) = |\tan^{-1}(\mathbf{G}_y(r, c)/\mathbf{G}_x(r, c))|. \quad (3.11)$$

**Step 3):** After obtaining  $\mathbf{M}$  and  $\mathbf{A}$ , we compute the histogram of the gradients in the following way. First, divide  $\mathbf{M}$  and  $\mathbf{A}$  into  $8 \times 8$  cells to form blocks (the blocks can have other sizes, here we use  $8 \times 8$  as an example. Denote the corresponding submatrix for the  $i^{th}$  cell as  $\mathbf{M}_i$  and  $\mathbf{A}_i$  respectively. Say we have  $16 \times 8 = 128$  cells. Now consider the  $i^{th}$  block and its corresponding submatrices  $\mathbf{M}_i$  and  $\mathbf{A}_i$ . For each block, a 9-point histogram is calculated. Specifically, let  $V \in \mathbb{R}^9$  record the value for the  $j^{th}$  bin,  $j \in [1 : 9]$ . For each cell (entry) in the  $i^{th}$  block, let  $j = \lfloor \mathbf{A}(r, c)/20^\circ \rfloor$  [54].

Then increase  $V(j)$  by  $\mathbf{M}(r, c) \cdot (\mathbf{A}(r, c)/20^\circ - 1/2)$ , and increase  $V(j + 1)$  by  $\mathbf{M}(r, c) \cdot (\mathbf{A}(r, c) - C(j))/20^\circ$ , where  $C(j) = 20^\circ(j + 1/2)$ .

**Step 4):** Once histogram computation is over for all blocks, 4 blocks from the 9 point histogram matrix are clubbed together to form a new block  $2 \times 2$ . This clubbing is done in an overlapping manner with a stride of 8 pixels. For all 4 cells in a block, we concatenate all the 9 point histograms for each constituent cell to form a feature vector with dimension 36. Denote the feature vector for the  $i^{th}$  (new) block as  $f_i$ .

**Step 5):** Normalize  $f_i$  for each block so that  $f_i$  has unit norm.

**Step 6):** Concatenate the feature vectors for all blocks and obtain the final feature vector for the image.

### 3.1.2.2 HOG analysis

Figure 3.7 are two example illustrations of oriented gradients, the arrows in the right-hand side images are indicating both the magnitude and directions of each block. The pointer of the arrow shows the direction of spatial blocks intensity change, and the color indicates the magnitude of change, of which lighter color means a larger magnitude.

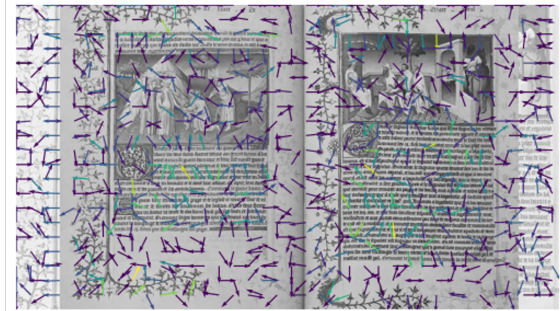
As can be seen, for content like a book, the feature vectors of cell responses scatter around the text and box edges. In images with figures, the algorithm can draw the edge of the character by the clue of contours.

Compared to color-based image representations, shape-based methods are robust to changes in brightness, contrast, or minor movement of the interest object and even handle deformations and slight occlusions well [54].

Another reason why this study also includes investigating this method is this descriptor can find and depict the interesting part of the image, and less computationally demanding when only CPU is provided. Much progress was made in improving both the accuracy and speed in the real-time application of HOG because of its effectiveness and robustness in outdoor human activity detection [55], where changes in environmental



(a) Example image - book



(b) HOG illustration - 1



(c) Example image - Polo



(d) HOG illustration - 2

Figure 3.7: Illustration of HOG results

lighting or background are frequent. This is also the challenge machine learning-based schemes face [56].

However, as elaborated above, its robustness is at a cost of efficiency, for it segments the image into small overlapping cells, conducts gradient calculations for each pixel, accumulates the histogram of gradient orientations, and concatenates the normalized block histograms to form the final feature vector. And for comparison or retrieval tasks, it is limited in providing contextual information [57], as a result of which it struggles to represent multiple interest points within the image (especially in historical materials).

### 3.1.3 Scale-Invariant Feature Transform (SIFT)

The Scale-Invariant Feature Transform (SIFT) algorithm is a handcrafted image feature extraction method. It excels in capturing unique and scale-invariant features from images that can robustly represent the main objects or even background scenes. Its prominent properties, including but not limited to scale-invariant, rotation invariant, and resistance to lighting change make it outstands in image processing and image retrieval in relatively large databases [2].

The SIFT algorithm utilizes scale-space filtering by applying the Difference of Gaussians (DoG) approximation to detect keypoints at various scales. The DoG images are then searched for local extrema over scale and space. By comparing each pixel with its neighbors in the current scale, as well as neighboring pixels in adjacent scales, potential keypoints are identified as local extrema, as the illustration in Figure 3.8.

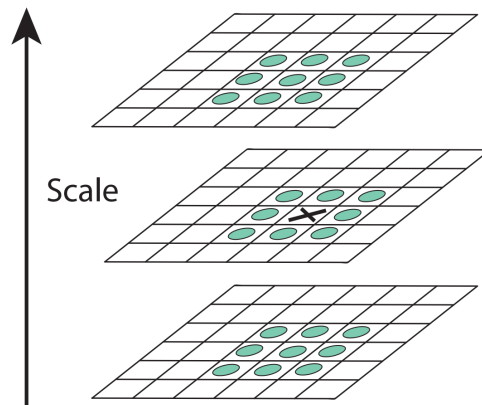


Figure 3.8: Illustration of searching for local maxima and minima of the DoG images (adapted from [2])

These keypoints represent the best scale for capturing distinctive features. This process forms the foundation for subsequent steps: assigning orientations and computing keypoint descriptors.

#### 3.1.3.1 SIFT descriptor calculation

In this subsection, more details of how to conduct the calculation and get the final descriptors will be discussed.

- Scale-space extrema detection: as mentioned above, it constructs a scale-space representation of an image by applying Gaussian blurring at multiple scales. It then identifies potential interest points, or keypoints, by locating local extrema in the Difference-of-Gaussian function across the scale space.
- Keypoint localization: after getting the candidate locations, the algorithm analyzes each candidate keypoint and eliminates unstable ones based on criteria such as low contrast, poorly defined edges, or unstable location.

- Orientation assignment: it computes the dominant orientation for each keypoint by considering the gradient magnitudes and orientations in its local neighborhood. This step helps achieve rotation invariance.
- Keypoint descriptor generation: a descriptor is computed for each keypoint to capture its local appearance. This descriptor is based on the gradients and orientations of the image patches surrounding the keypoint. It is designed to be invariant to changes in illumination, scale, and rotation.

### 3.1.3.2 SIFT feature analysis

Figure 3.9 are the example results of SIFT extracted keypoints. The images presented here are extracted from two historical videos featuring the renowned travelers Marco Polo and Ibn Battuta. This figure highlights the keypoints identified using the SIFT algorithm. Distinct colors have been assigned to different ranges or intensities, spanning from blue to red via green and yellow. The progression of colors, from blue to yellow, represents the keypoint score, indicating the significance of each keypoint. As observed in the provided figures, the SIFT algorithm successfully produces highly distinctive keypoints which are also robust against partial object appearances, providing key information from multiple objects simultaneously.

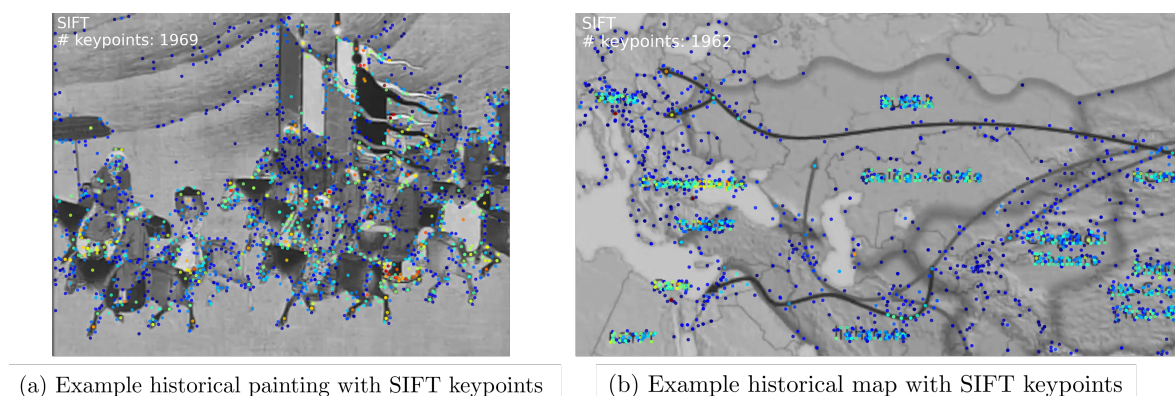


Figure 3.9: SIFT detected keypoints on example images

Furthermore, Figure 3.10 shows the matching results of another two frames extracted from a historical video. In this example, after using SIFT to depict certain keypoints in the images, we also employ an approximate nearest neighbor search algorithm on these keypoints. The search is to establish a match between corresponding keypoints from two frames. It offers a comparison of their similarity and showcases the uniqueness of each keypoints.

Overall, the performance of SIFT is sufficient to deal with the relatively large image dataset (around thousands of images) comparison task, but its efficiency is not high enough. The computational load in SIFT primarily stems from the utilization of the Difference-of-Gaussian approximation in scale-space extrema detection. Speeded Up Robust Features (SURF) is a faster alternative feature detection and description algorithm. It employs a technique called the Hessian matrix together with a more efficient

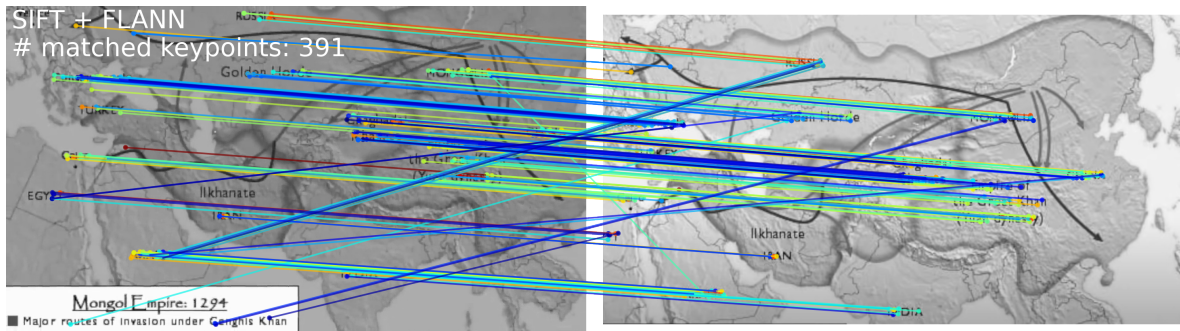


Figure 3.10: SIFT keypoints matching results on example images

approximation technique to detect interest points in an image. These modifications result in a threefold increase in speed for detection and description implementations than SIFT, which could greatly contribute to the efficiency of overall image feature extraction and comparison [31]. However, it is important to note that due to patent restrictions, the implementation of SURF is disabled in several open-source libraries. Consequently, this application-oriented study will focus on and evaluate SIFT instead of SURF.

### 3.1.4 Deep learning-Based Features

Convolutional Neural Networks (CNNs) have achieved remarkable success in the field of and keep drawing attention from various research areas, including image retrieval, object detection, activity detection, facial recognition and more [58]. The availability of extensive datasets and repositories, such as ImageNet [59], has been instrumental in the training of multilayered neural networks on massive amounts of data. By using a backpropagation algorithm, these models can iteratively fine-tune their internal parameters, enabling the computation of progressively abstract representations. The large-scale datasets have greatly enhanced the learning capabilities of neural networks and the integration of high-performance computing systems has made intensive computations possible [60]. Therefore, deep convolutional networks have been revolutionizing image and video processing, leading to significant breakthroughs in the aforementioned domains. They enable the creation of complex computational models with multiple layers, allowing them to learn hierarchical representations of data [61].

In detail, CNNs use convolutional layers that apply filters to local regions of the input image. This allows them to capture local patterns and features, which are essential for understanding the spatial structure of images. The localized receptive fields help in extracting meaningful features from different parts of the image. Generally, lower layers learn simple features like edges and textures, while higher layers learn more complex features and representations. This hierarchical feature learning allows CNNs to capture both low-level and high-level features, enabling them to understand the hierarchical structure of images [3].

It is not surprising that video summarization aims to leverage this technology to extract crucial information from videos more accurately. In fact, there is a growing interest in not only identifying the key elements but also uncovering the underlying

storyline, mirroring human-like comprehension [22, 30, 62, 63]. But these works require more than just image understanding. They also incorporate the textual information and grow graphs for understanding the scene relations, which would be overly sophisticated for our specific task, as our focus primarily revolves around efficient image understanding. In this sense, we intend to employ up-to-date models to effectively represent the images, aligning with our objectives.

In this section, two deep learning architectures that are commonly used as backbones for image representation in image interpretation tasks will be discussed. One is VGGNet (developed by an Oxford group named Visual Geometry Group), famous for its simplicity and effectiveness [4]. Another one is ResNet (Residual Neural Network), which significantly improved training deep neural networks by tackling the problem of vanishing gradients [5].

### 3.1.4.1 VGGNet

The VGG (Visual Geometry Group) architecture was proposed as an easy and efficient design principle for Convolutional Neural Networks (CNNs) in the field of image recognition. It has a deeper architecture compared to its predecessors.

The left illustration of Figure 3.11 shows the general structure of VGG incorporated with the idea that filters with small sizes could enhance CNN performance by using a stack of  $3 \times 3$  filters instead of larger  $5 \times 5$  and  $11 \times 11$  filters used in its predecessor ZefNet [64]. The parallel assignment of these small-size filters proved to be as effective as larger filters in terms of receptive field efficiency and produced similar results [3]. Additionally, using small-size filters reduced the number of parameters and computational complexity.

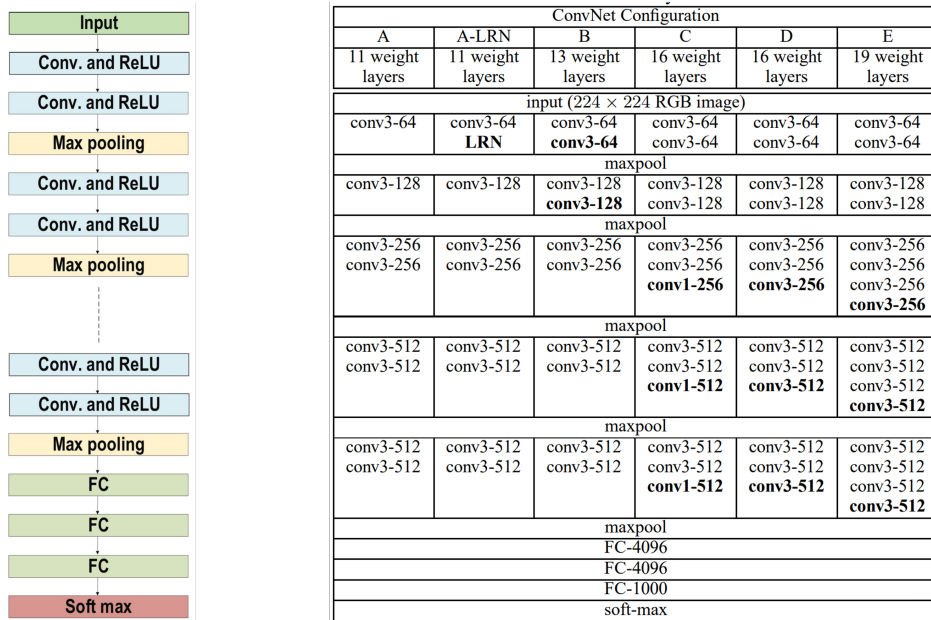


Figure 3.11: Illustration of VGGNet structure (adapted from [3, 4])

To regulate network complexity, VGG introduced  $1 \times 1$  convolutions in the middle

of the convolutional layers as shown in the right-hand side of Figure 3.11, increasing the nonlinearity of the decision function. After the convolutional layers, the structure includes a max pooling layer and applies padding to maintain spatial resolution. VGG achieved significant results in image classification task [4] for its enlarged depth, homogenous topology, and simplicity compared to the predecessor networks.

VGG16 and VGG19 are two variations of the VGG architecture. Figure 3.12 illustrates the construction of their detailed structure (VGG16 is indicated in this figure by removing 3 convolutional layers).

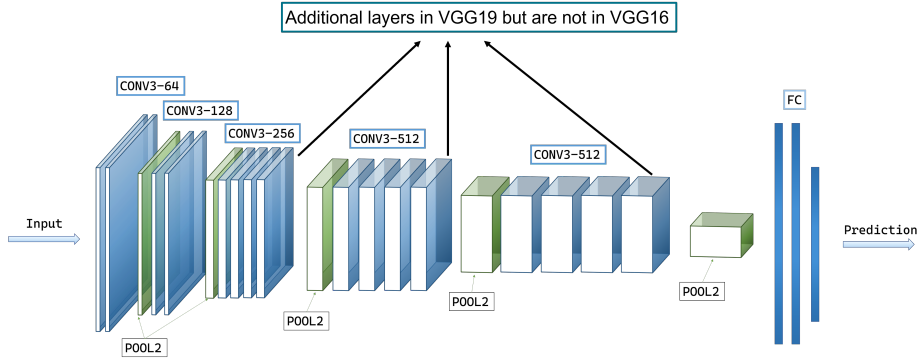


Figure 3.12: Illustration of VGG19 structure

VGG16 consists of 16 layers, including 13 convolutional layers and 3 fully connected layers. It starts with a series of convolutional layers with  $3 \times 3$  filters, followed by max-pooling layers [4]. The network gradually increases the number of filters as the spatial resolution decreases. VGG19 has additional convolutional layers which provide increased representational capacity but also make the network more computationally expensive. It is generally used when a more complex model is required, or when the task at hand demands a higher level of feature extraction and discrimination. In our study, we will implement and analyze the performance of VGG16 in KFE module. Figure 3.13 shows an example image together with a few visualizations of its feature maps.

### 3.1.4.2 ResNet

ResNet, short for Residual Neural Network, is a deep learning architecture that revolutionized image classification tasks. It tackles the challenge of training very deep neural networks by introducing shortcut connections, allowing the network to effectively propagate gradients during training and preventing the vanishing gradient problem. This innovation enables it to achieve remarkable accuracy and has influenced subsequent advancements in the field of deep learning.

Figure 3.15 shows the VGG19 network, plain network, and Residual network structures. The middle model is called a plain network inspired by VGG nets (the left one of Figure 3.15). It consists of convolutional layers with  $3 \times 3$  filters and follows two design rules: the same number of filters for the same output feature map size, and doubling the number of filters when the feature map size is halved to maintain time complexity per layer. Downsampling is performed using convolutional layers with a



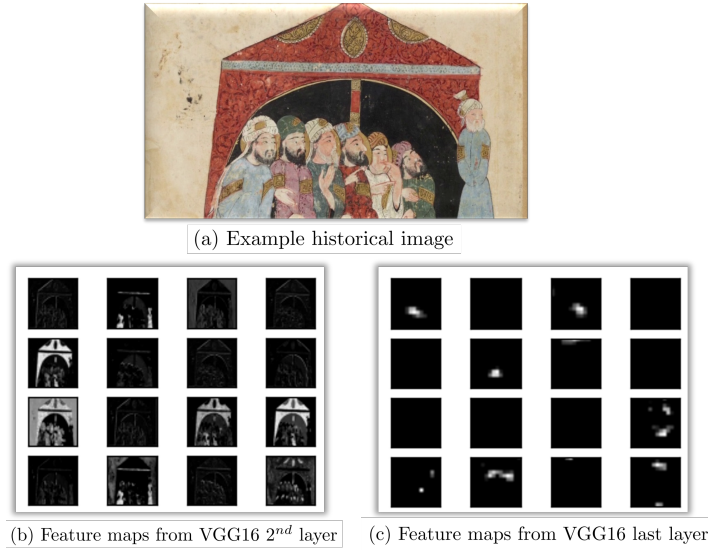


Figure 3.13: Example feature maps from different VGG16 layers

stride of 2. The network concludes with a global average pooling layer and a 1000-way fully-connected layer with softmax. This inspired baseline model has a considerable reduction in the computational complexity of VGG nets for the plain network FLOPs is only 18% of VGG19 model.

The right model in Figure 3.15 is the proposed Residual Network-34, an extension of the plain network. Shortcut connections are inserted into the network, transforming it into its residual version. Figure 3.14

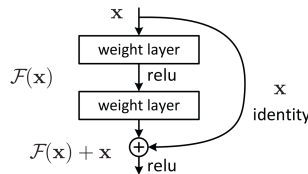


Figure 3.14: Illustration of a building block for residual learning (adapted from [5])

The definition of a building block is:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}, \quad (3.12)$$

meaning the output  $\mathbf{y}$  is a combination of input  $\mathbf{x}$  and a residual mapping to be learned  $F$ , defined as  $\mathcal{F}(\mathbf{x}, \{W_i\})$  are the input and output vectors of the layers considered.

Identity shortcuts are used when the input and output have the same dimensions. The input vector is directly added to the output of a layer without any transformation or alteration, increasing dimensions. It also allows the network to learn residual mappings by comparing the original input to the transformed output. To match the dimensions, two options are considered: one is to use identity mapping with zero padding, introducing no extra parameters and the other one is to employ  $1 \times 1$  convolutions to match dimensions. When the shortcuts span feature maps of different sizes, they are performed with a stride of 2.

### 3.1.4.3 Models analysis

To have a clearer idea of the parameters amount and calculation complexity of different models, which can greatly influence the efficiency of extracting image features, we present the properties of these models in Table 3.1.

Table 3.1: Properties comparison of models

Dataset	Model	Number of conv. Layers	Number of parameters	FLOPs
ImageNet	VGG-16	13	138M	15.5G
ImageNet	ResNet-18	18	11.5M	1.8G
ImageNet	ResNet-34	34	21.8M	3.6G
ImageNet	ResNet-50	50	25.6M	3.8G
ImageNet	ResNet-101	101	44.5M	7.6G
ImageNet	ResNet-152	152	60.2M	11.3G

As can be seen, VGG nets have a significant number of parameters and FLOPs, resulted from the inclusion of pooling layers for each block and a stack of convolutional layers with small 3 filters. On the other hand, ResNet introduces residual connections, which enable the network to learn residual mappings rather than directly learning the entire mapping from scratch. These connections allow for the flow of gradients directly from later layers to earlier layers in training, bypassing several intermediate layers. By leveraging these shortcuts, ResNet can effectively address the degradation problem that arises with very deep networks.

The use of residual connections reduces the number of parameters in ResNet compared to VGG because it reduces the necessary complexity to learn each layer. Instead of trying to learn all the details of a particular layer, ResNet focuses on learning the residual information, which requires fewer parameters. This parameter efficiency allows ResNet to achieve similar or even better performance than VGG while utilizing fewer parameters.

This thesis will explore the efficiency and accuracy of these models in interpreting a large number of historical materials. In Figure 3.16, an example image with a few visualizations of its feature maps extracted from ResNet-18 can be found. Together with Figure 3.13, it is evident that feature maps capture local patterns and structures in the input images. The filters in the network focus on a specific region of the input image and learn to detect local patterns such as corners, textures, and edges. And the hierarchical structure allows the model to extract relevant features at multiple levels of abstraction.

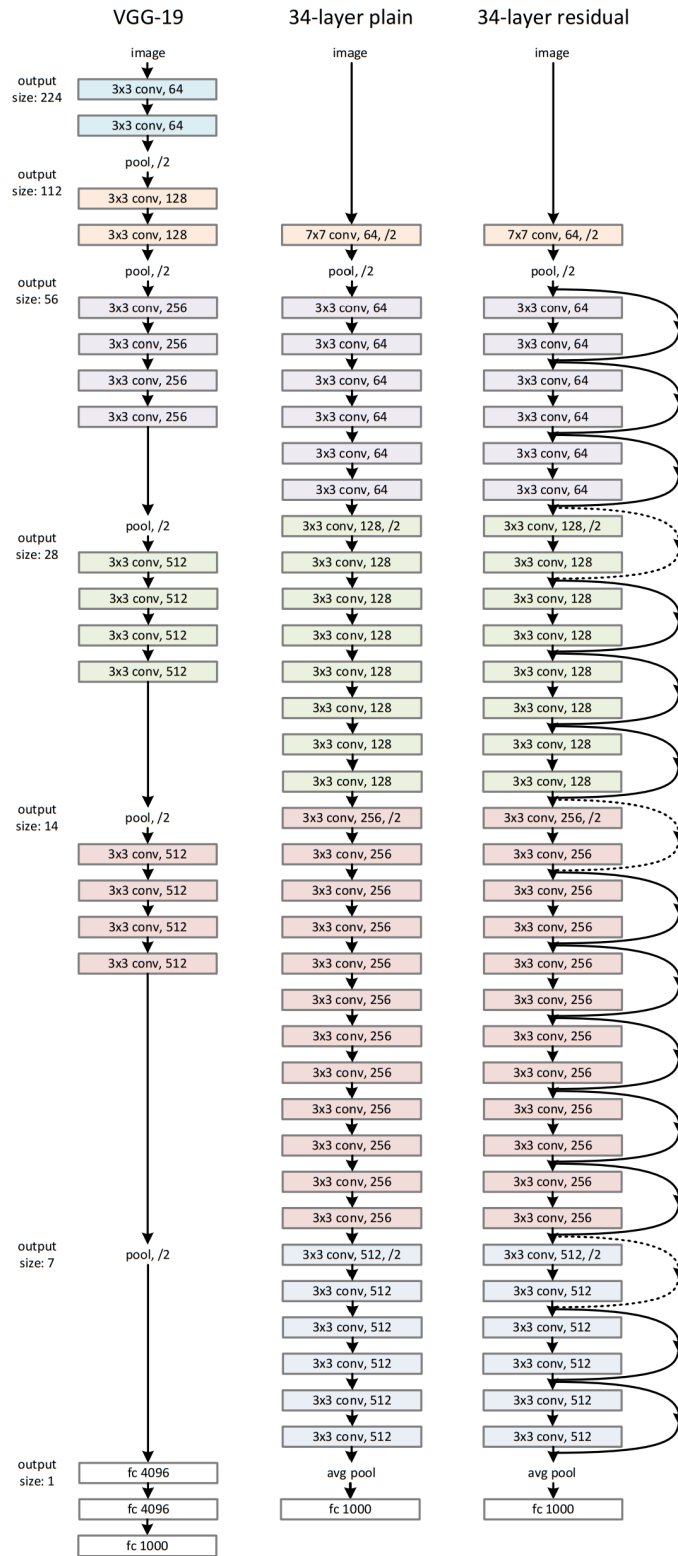


Figure 3.15: Network architectures comparison for ImageNet (adapted from [5])

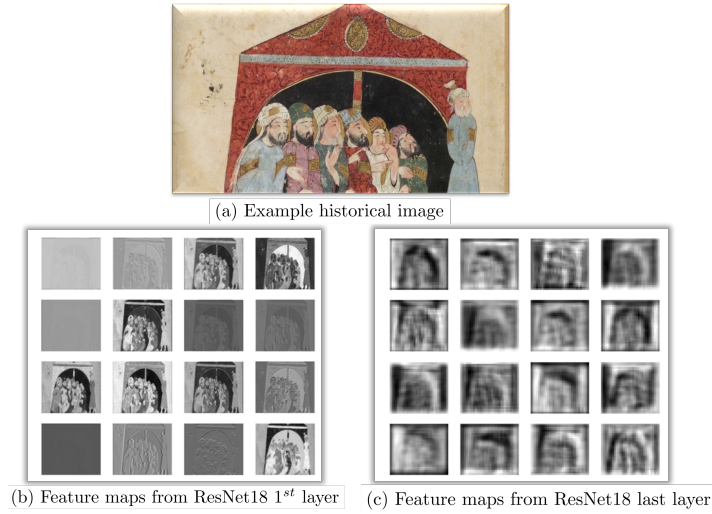


Figure 3.16: Example feature maps from different ResNet18 layers

### 3.1.5 Hybrid Feature (HF)

#### 3.1.5.1 Problem statement

After conducting preliminary tests on the methods mentioned in the KFE module, I have obtained several results. Upon analyzing these results, I have identified several inherent problems in these methods, along with a few noteworthy points. To name the most prominent ones, traditional methods, and deep learning-based methods have the following limitations:

- Color histogram-based features are greatly influenced by the presence of black bars within the frame. As explained in Section 1, our improved color histogram features incorporate spatial information by dividing the frame into blocks and extracting the features accordingly. However, videos sourced from certain platforms targeting mobile viewership and propagation often contain large side black bars. In such cases, the color histogram is greatly influenced by these bars, leading to potential inaccuracies in the extracted features.
- Deep learning-based features, on the other hand, are not influenced by the black bars. The inclusion of filters in deep learning models enables them to detect patterns of interest without considering the surrounding less relevant areas. Consequently, the feature maps generated by deep learning approaches do not give attention to these black bars and remain unaffected by their presence.
- Deep learning-based features can be negatively impacted by similar interest patterns that appear in consecutive frames. Due to the nature of deep learning algorithms, which aim to capture temporal dependencies, the presence of similar patterns across successive transition frames can introduce redundancy and lead to diminished performance.
- Color histogram-based features are not influenced by similar interest patterns in

consecutive frames. Since color histogram features rely on the distribution of colors in each frame individually, the presence of similar patterns in consecutive frames does not impact the extraction process.

Therefore, considering the potential benefits derived from the respective strengths of both approaches, we have thought about the possibility of combining them. As a result, we propose a hybrid feature that integrates key components from both methods. In the next section, we will discuss the hybrid feature structure.

### 3.1.5.2 Hybrid feature construction

The hybrid feature is a combination of a de-dimensionalized color-based feature and a Resnet18 feature vector. The structure is shown in Figure 3.17. Both components are normalized according to the weight assigned to them. This feature can leverage the sensitivity of high-level features in detecting instances within an image, without being influenced by a large area of non-informative content. It can also mitigate the influence of high-level feature fixed interest points when dealing frames with minor changes.

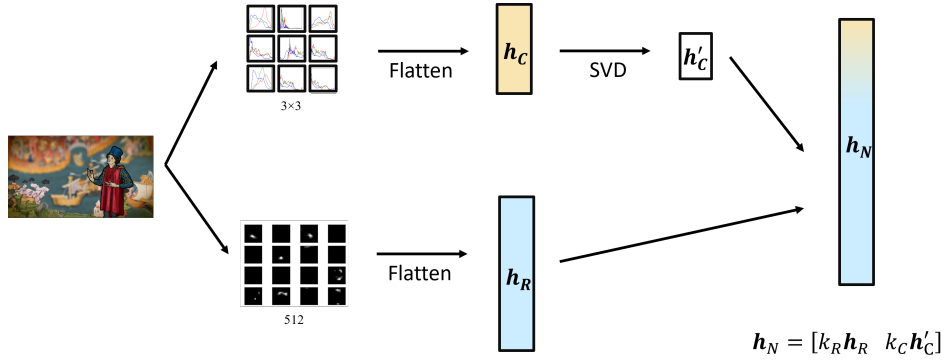


Figure 3.17: Illustration of hybrid feature structure

The structure of this hybrid feature is as follows: given an image  $I$ , denote  $\mathbf{h}_R$  as the feature vector of  $I$  extracted using ResNet, and  $\mathbf{h}_C$  as the feature of  $I$  extracted using color histogram. We construct a new feature vector,

$$\mathbf{h}_N = [k_R \mathbf{h}_R \quad k_C \mathbf{h}_C], \quad (3.13)$$

where  $k_R, k_C$  are non-negative weights. Note that if  $\mathbf{h}_R$  and  $\mathbf{h}_C$  are normalized such that  $\|\mathbf{h}_R\|_p = \|\mathbf{h}_C\|_p = 1$  for any  $p$  norm  $\|\cdot\|_p$ , then the new feature is automatically normalized if we choose  $k_R^p + k_C^p = 1$  because

$$\|\mathbf{h}_N\|_p = (k_R^p + k_C^p)^{1/p} = 1. \quad (3.14)$$

By tuning  $k_R$  and  $k_C$ , we are able to find good features derived from both deep learning-based and traditional color histogram, thereby obtaining an ideal hybrid feature. For this particular task, I have assigned equal weights of half to each component.

## 3.2 Clustering Methods

As mentioned in Chapter 1, keyframe extraction module is to remove consecutive and redundant frames from the input query video. Clustering is a useful way to group similar image features after which a frame from each group could be selected to serve as the key frame and represent the shot. Figure 3.18 is a demonstration of what result we expect the KFE module to produce. In this section, two clustering algorithms: K-means clustering and dynamic clustering that are widely used to group similar objects or features will be discussed.

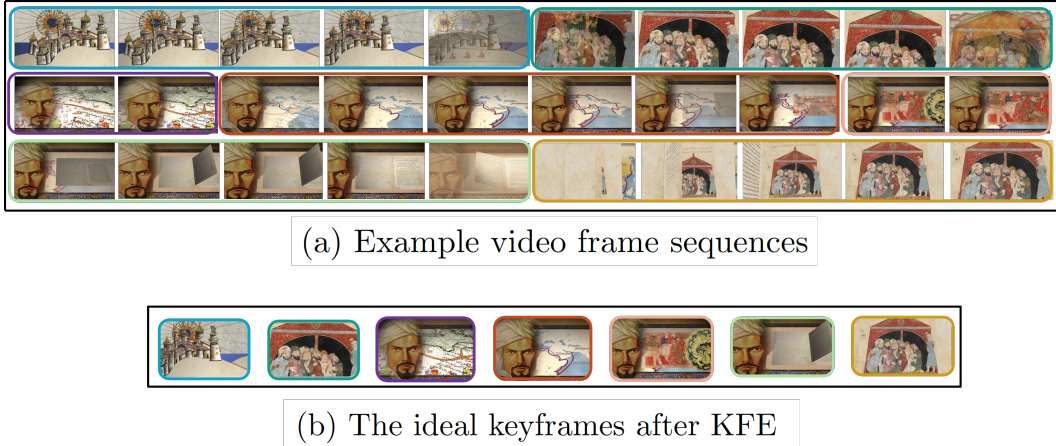


Figure 3.18: A demonstration of KFE results

### 3.2.1 K-means Clustering

K-means clustering is a centroid-based clustering algorithm that aims to divide a dataset into  $K$  clusters ( $K$  is a predefined hyperparameter) and provide the center point corresponding to each cluster, which is the key frame in our case.

VSUMM model is utilizing the K-means clustering algorithm for its simplicity and effectiveness. [6]. Their pipeline is shown in Figure 3.19

K-means clustering algorithm operates iteratively and follows these main steps:

1. Initialization: Randomly select  $K$  points from the dataset as initial cluster centroids, each is denoted as  $\mu_1^{(0)}, \mu_2^{(0)}, \dots, \mu_k^{(0)}$ .
2. Assign features: Assign each feature (data point) to the nearest centroid based on a distance metric, using Euclidean distance. The loss function is defined as:

$$J(c, \mu) = \min \sum_{i=1}^M \|\mathbf{f}_i - \mu_{c_i}\|^2, \quad (3.15)$$

where  $c_i$  is the  $i^{th}$  feature center point vector,  $M$  is the number of the features. After the iteration, each frame color histogram vector  $\mathbf{f}_i$  is assigned to its nearest center cluster vector.

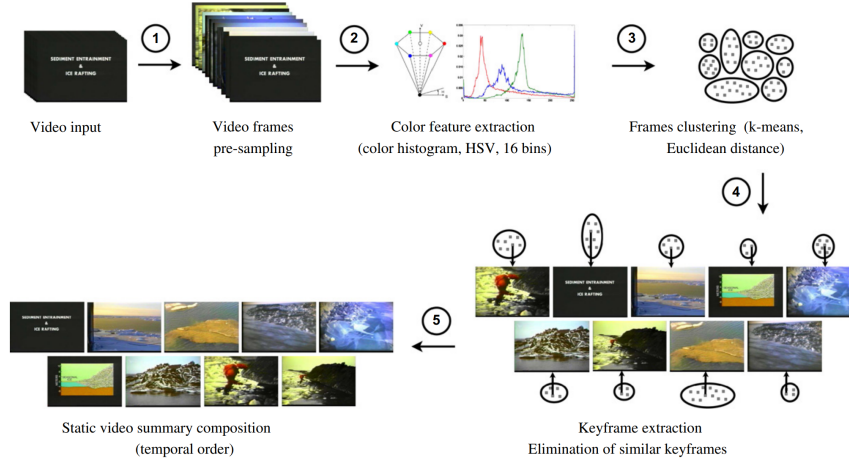


Figure 3.19: VSUMM approach pipeline (adapted from [6])

3. Update: Recalculate the centroids of each cluster by taking the mean of all data points assigned to that cluster.

In  $t^{th}$  iteration, the center point and center cluster are updated according to:

$$\mu_k^{(t+1)} < - \operatorname{argmin}_{\mu} \sum_{i:c_i^t=k}^b \|\mathbf{f}_i - \mu\|^2, \quad (3.16)$$

$$c_i^t < - \operatorname{argmin}_k \|\mathbf{f}_i - \mu_k^t\|^2. \quad (3.17)$$

The algorithm iterates the assignment and updates steps until the centroids no longer change significantly or a maximum number of iterations is reached [65]. The goal of K-means clustering is to minimize the within-cluster sum of squares, also known as inertia. It seeks to find the best partition of the data into K clusters, where the data points within each cluster are as close to each other as possible.

However, K-means clustering has some limitations. First, the algorithm is sensitive to the initial centroid selection, potentially leading to convergence on suboptimal solutions. Secondly, in our scenario, where the same object may appear across multiple shots dispersed throughout the video, K-means clustering treats these frames collectively, disregarding the temporal attribute inherent in individual frames. Thirdly, K-means clustering requires a predefined number of clusters, but videos have different durations and scene transition frequencies. If  $K$  is assigned too small, videos with frequent scene transitions cannot be processed well, leading to lower accuracy. Conversely, when  $K$  is set too large, videos with fewer transitions may be excessively fragmented, resulting in redundancy keyframes. Finally, the computation of distances between every feature and every center vector can be relatively time-consuming, causing slower processing speeds.

### 3.2.2 Dynamic Clustering

Another centroid-based clustering algorithm is dynamic clustering, which offers several advantages. Dynamic clustering is an unsupervised and scalable technique with linear complexity [66]. Unlike other traditional clustering algorithms that work on static datasets, dynamic clustering is designed to handle data that changes over time or where new data points are continuously added. This temporal adaptability makes dynamic clustering particularly well-suited for video data, given its inherent temporal characteristics. Algorithm 1 showcases the implementation of dynamic clustering in grouping frames and detecting keyframes.

---

**Algorithm 1:** Dynamic clustering in KFE

---

```

Data: Feature set:  $\mathbf{F}$ , Num. of features:  $N$ , Threshold:  $thr$ 
Result: Final centroids:  $\mathbf{C}$ , Keyframe indices:  $\mathbf{I}$ 
Two initial features:  $\mathbf{F}_1, \mathbf{F}_2$ ;
One initial centroid:  $C_0 = \frac{\mathbf{F}_1 + \mathbf{F}_2}{2}$ ;
 $i \leftarrow 2$ ;
 $n \leftarrow 2$ ;
while  $i \leq N$  do
     $\mathbf{C}_k^{i+1} \leftarrow \frac{n \cdot \mathbf{C}_k^i + \mathbf{F}_{i+1}}{n+1}$ 
    if  $\cos(\mathbf{C}_k^i, \mathbf{C}_k^{i+1}) \leq thr$  then
         $\mathbf{C}_k \leftarrow \mathbf{C}_k^{i+1}$ ; /* Update current centroid */
         $n \leftarrow n + 1$ ;
         $i \leftarrow i + 1$ ;
    else
        if  $\cos(\mathbf{C}_k^i, \mathbf{C}_k^{i+1}) \geq thr$  then
             $\mathbf{C}_k \leftarrow \mathbf{C}_k^i$ ; /* Store the centroid before update */
             $I_k \leftarrow i$ ; /* Store the keyframe index */
             $n \leftarrow 2$ ;
             $i \leftarrow i + 2$ ;
             $k \leftarrow k + 1$ ;
            ; /* Update the next centroid index */
             $\mathbf{C}_k^i \leftarrow \frac{\mathbf{F}_{i-2} + \mathbf{F}_{i-1}}{2}$ ; /* Initialize the next centroid */
        end
    end
end

```

---

Figure 3.20 is an illustration of the process for these two cases shown in the algorithm.

And the threshold defined in the "Data" is a decision boundary of how similar two centroid vectors should be. And we use Cosine Similarity defined in Equation 3.18 as the measure of similarity check.

$$\cos(\theta) = \frac{\mathbf{c}_1 \cdot \mathbf{c}_2}{\|\mathbf{c}_1\| \|\mathbf{c}_2\|} = \frac{\sum_{i=1}^n c_1^i c_2^i}{\sqrt{\sum_{i=1}^n (c_1^i)^2} \sqrt{\sum_{i=1}^n (c_2^i)^2}} \quad (3.18)$$



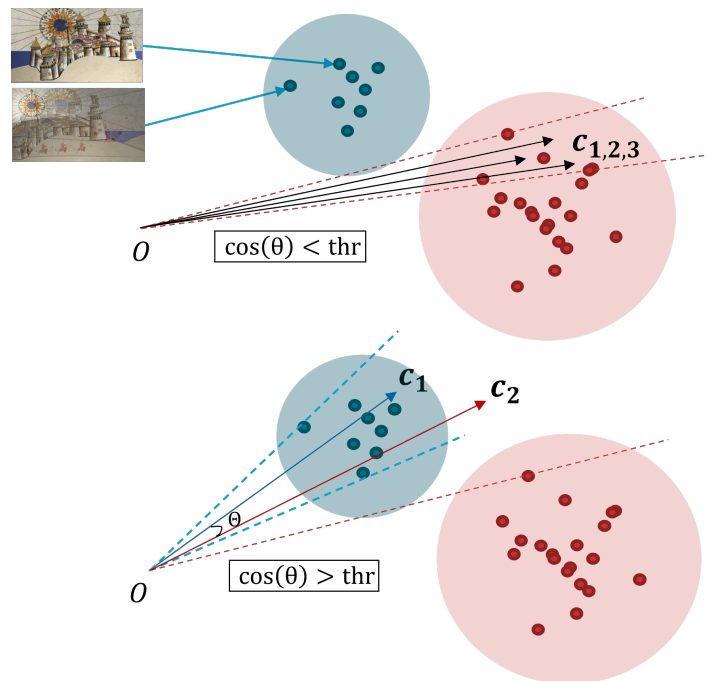


Figure 3.20: Illustration of the process clustering upcoming frame into one cluster and the process establishing a new cluster

Moreover, it does not require predefined hyperparameters. It seamlessly adjusts to changes in the data distribution and can automatically adjust the cluster assignments as a new feature becomes available. This means the aforementioned dilemma of setting a suitable predefined number of clusters can be resolved.



# Experiments and Results

---

## 4.1 Dataset

The dataset comprises a collection of videos that focus on the exploration journeys undertaken by three prominent explorers: Ibn Battuta, Zheng He, and Marco Polo. These videos have been meticulously selected by a historical scholar Andrea Natetti from School of Art, Design and Media, the Nanyang Technological University (NTU). These videos are also included in his featured project: Engineering Historical Memory Project (EHM, <https://engineeringhistoricalmemory.com/>) and ensured to be relevant and accurate. EHM offers a broad spectrum of perspectives and scholarly contributions, ensuring a well-rounded and comprehensive collection of historical resources.

### 4.1.1 Properties

#### 4.1.1.1 Resolution

The videos encompass different resolutions, including  $1280 \times 710$ ,  $480 \times 360$ ,  $576 \times 1024$ , and  $540 \times 960$ . This diverse range of resolutions caters to different display capabilities and enables us to test different methods and optimize our algorithms on various video qualities.

#### 4.1.1.2 Duration

Regarding duration, the videos are categorized into three distinct segments: 0~3 minutes, 3~10 minutes, and 10+ minutes. A wide range of durations enable us to evaluate the efficiency and stability of the methods. And the videos have different frame rates, including 25, 30, and 60 frames per second (fps). By considering videos of different durations and frames per second, we can gain valuable insights into the robustness and scalability of different algorithms and techniques.

### 4.1.2 Content

The videos cover a wide range of content, including depictions of historical figures, iconic landmarks, historical maps, and even animated representations. This rich and diverse content enables us to evaluate the accuracy and robustness of different methods in video processing and image retrieval. It imposes requirements for the methods to have a comprehensive interpretation of frames. The inclusion of such diverse content imposes rigorous requirements on methods to possess a comprehensive interpretation of frames.

In addition to the videos, the dataset provides a supplementary resource of approximately 10 query images for each video. These images serve as queries for conducting specific retrieval test. Figure 4.1 is a video-queries pair example. It consists of one historical video and ten image queries.

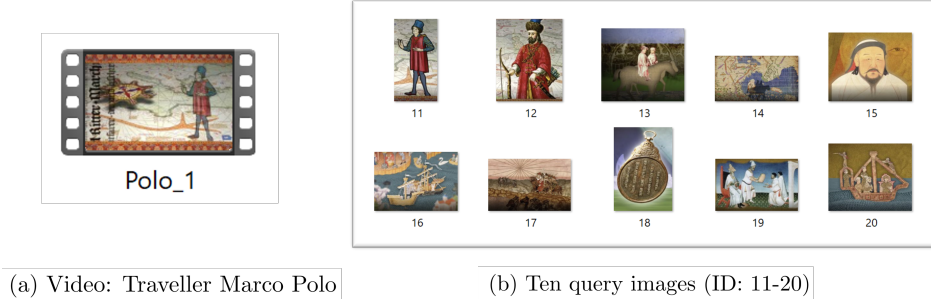


Figure 4.1: Dataset example (Video-Queries Pair)

### 4.1.3 Groundtruth

Besides the video-query pairs, we also annotated all occurrences of every query image as the groundtruth. Table 4.1 is an example groundtruth sheet for a single video.

1. Query: This column represents the query identifier. In the given example, the queries are represented by numbers (431, 432, etc.). Each unique query has its own row in the sheet.
2. Start Stamp: This column indicates the starting timestamp of a query occurrence. The format used is minutes:seconds (mm:ss). For instance, in the first row, the start stamp is 0:01, meaning the query starts at 0 minutes and 1 second.
3. End Stamp: This column specifies the ending timestamp of a query occurrence. Similar to the start stamp, it is also in the format mm:ss. In the first row, the end stamp is 0:43, indicating that the query ends at 0 minutes and 43 seconds.
4. Query ID: This column also consists of the identifier or code associated with a specific query.
5. Number of Occurrences: This column indicates the count or frequency of a specific query occurrences. It represents how many times a particular query appears within the specified time range. In the first row, the number of occurrences is 1, suggesting that query 431 appears once within the given video.

Overall, this meticulously curated dataset serves as an academic resource, offering a rich assortment of videos and query images with their occurrences information. It can be used in evaluating both accuracy and redundancy of the keyframe extraction module. Additionally, it can be leveraged to assess the mean Average Precision (mAP) in the content-based image retrieval (CBIR) module. In next section, the metrics of evaluating the proposed methods will be discussed.

Table 4.1: Example grountruth for a video

Query	Start stamp[s]	End stamp[s]	Query ID	Number of occurrences
431	0:01	0:43	431	1
432	0:43	0:48	432	13
432	2:03	2:26	433	7
432	2:38	2:52	434	5
432	3:27	3:42	435	1
432	4:08	4:13	436	2
432	4:20	4:31	437	1
432	5:07	5:09	438	1
432	5:26	5:47	439	1
432	6:13	6:19	440	1
432	6:40	6:43		
432	7:05	7:10		
432	7:14	7:20		
432	7:45	7:53		
433	1:13	1:38		
433	1:45	1:54		
433	1:54	2:07		
433	2:16	2:26		
433	2:38	2:52		
433	5:05	5:09		
433	7:57	8:00		
434	2:26	2:44		
434	2:54	3:08		
434	3:55	4:03		
434	6:43	6:53		
434	7:36	7:45		
435	3:08	3:11		
436	4:08	4:13		
436	4:20	4:24		
437	5:30	5:34		
438	6:21	6:33		
439	6:33	6:43		
440	7:20	7:31		

## 4.2 Metrics

In this section, the metrics to evaluate both modules are presented. For Key Frame Extraction (KFE) module, accuracy, redundancy and efficiency ratio are considered. Content-Based Image Retrieval (CBIR) module is evaluated using mean Average Precision (mAP) and efficiency ratio. These metrics provide a comprehensive framework for evaluating the performance and efficiency of each module.

### 4.2.1 Accuracy

The accuracy metric quantifies the proportion of correctly identified intervals, where keyframes are correctly retained or discarded within those intervals. This evaluation criterion examines whether an interval is labeled as positive or negative based on the presence or absence of at least one retained frame index falling within the duration of that interval. The interval is defined by its start and end timestamps, and the accuracy metric compares the actual state of the interval (positive or negative) to the predicted state based on the presence of retained keyframes. The accuracy is defined as Equation 4.1.

$$\text{Accuracy} := \frac{\text{Number of positive intervals}}{\text{Number of all intervals}}. \quad (4.1)$$

The accuracy metric serves as a reliable indicator of the correctness of the keyframe extraction results. High accuracy signifies a reliable and precise performance of the method. Accuracy is a widely used evaluation metric that provides a straightforward and intuitive understanding of the method performance. It allows for easy comparison between different methods and enables quick assessment of the overall effectiveness of the KFE module.

To illustrate the significance of accuracy, we present two examples. In the first example, the keyframe extraction accuracy is relatively low, indicating inaccuracies or missed keyframes. Conversely, in the second example, the keyframe extraction process achieves a high accuracy level, accurately identifying relevant keyframes. These examples highlight the impact of accuracy on the quality and reliability of the keyframe extraction results. In the first example shown as Figure 4.2, the total number of intervals in this example is 8, and the number of positive intervals is 7. Therefore, its accuracy is  $\text{Accuracy}_1 = 0.875$

While in the second example shown as Figure 4.3, the total number of intervals in this example is also 8, and the number of positive intervals is 8. Therefore, its accuracy is  $\text{Accuracy}_2 = 1$

These are two simple examples showing how the accuracy in KFE is defined for a single video, but for more than one video, we take the mean of all accuracies from all video results as in Equation 4.2.

$$\text{mean Accuracy} := \frac{\sum_{k=1}^{N_v} \text{Accuracy}_k}{N_v}, \quad (4.2)$$

where  $N_v$  denotes the number of videos considered.

### 4.2.2 Redundancy

Redundancy metric quantifies the level of redundancy in the keyframe extraction. It represents how much redundancy exists within a specific interval of keyframes. A lower redundancy value indicates a higher degree of diversity and uniqueness among the keyframes within that interval. On the contrary, a higher redundancy value suggests that the keyframes within the interval are more similar or redundant. It is defined as Equation 4.3.








Interval index	start stamp/s	end stamp/s	Results
11-1	0:02	0:13	
11-2	1:11	1:24	
11-3	2:05	2:10	
11-4	3:29	3:41	
11-5	4:39	4:46	
12-1	0:13	0:21	
12-2	4:17	4:22	
12-3	4:46	5:10	

Figure 4.2: KFE results - example one

$$\text{Redundancy} := 1 - \frac{1}{\text{Number of keyframes within the given interval}}. \quad (4.3)$$

This metric enables an assessment of the variation and representativeness of the extracted keyframes. It allows for the evaluation of how well the keyframe extraction method captures diverse visual content, providing insights into the richness and distinctiveness of the selected keyframes within a given interval. Redundancy serves as an indicator of the computational burden or stress that the Content-Based Image Retrieval (CBIR) module will encounter. A lower redundancy value suggests a reduced level of redundancy among the keyframes, implying that fewer calculations or computations will be required by the CBIR module. This positively impacts the efficiency ratio, as lower redundancy translates to a more streamlined and efficient retrieval process. By minimizing redundancy, the CBIR module can operate more swiftly and effectively, enhancing the overall efficiency of the system.

The two examples can also be compared in this metric as well. In the first example, the keyframe extraction redundancy is relatively high, indicating there still are redundant keyframes retained. In the second example, the keyframe extraction process achieves a low redundancy level, accurately discarding unuseful frames. In the first example shown as Figure 4.2, its redundancy is  $\text{Redundancy}_1 = 0.476$ , while in the second example shown as Figure 4.3, its redundancy is  $\text{Redundancy}_2 = 0.146$ .

Also, for more than one video, we take the mean of all redundancies from all video results as in Equation 4.4.







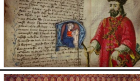

Interval index	start stamp/s	end stamp/s	Results
11-1	0:02	0:13	
11-2	1:11	1:24	
11-3	2:05	2:10	
11-4	3:29	3:41	
11-5	4:39	4:46	
12-1	0:13	0:21	
12-2	4:17	4:22	
12-3	4:46	5:10	

Figure 4.3: KFE results - example two

$$\text{mean Redundancy} := \frac{\sum_{k=1}^{N_v} \text{Redundancy}_k}{N_v}, \quad (4.4)$$

where  $N_v$  denotes the number of videos considered.

### 4.2.3 Efficiency Ratio

**Efficiency Ratio:** The efficiency ratio measures the ratio of video duration and certain module's processing time. It quantifies the speed improvement achieved by the system, indicating how much faster the method processes the video compared to the minimum human processing time, which is the video duration. Equation 4.5 is its definition.

$$\text{Efficiency Ratio} := \frac{\text{Video duration}}{\text{Computation time}}. \quad (4.5)$$

And for all the videos tested, we should calculation the mean Efficiency Ratio (mER) which is defined as Equation 4.6:

$$\text{mean Efficiency Ratio} := \frac{\sum_{k=1}^{N_v} \text{ER}_k}{N_v}, \quad (4.6)$$

The efficiency ratio provides a straightforward measure of computational efficiency and speed enhancements. It allows for comparisons between different methods to determine the efficiency of optimization efforts.



#### 4.2.4 mean Average Precision (mAP)

Mean Average Precision (mAP) is a metric commonly used to evaluate the performance of information retrieval systems. It combines the average precision values of multiple queries to calculate the mean, representing the overall effectiveness of the retrieval methods. The calculation of mAP includes precision and recall calculation.

Precision is calculated as the intersection of relevant and retrieved images divided by the total number of retrieved images (Equation 4.7). Recall is the intersection of relevant and retrieved images divided by the total number of relevant images (Equation 4.8).

$$\text{Precision} = \frac{|\{ \text{relevant images} \} \cap \{ \text{retrieved images} \}|}{|\{ \text{retrieved images} \}|}, \quad (4.7)$$

$$\text{Recall} = \frac{|\{ \text{relevant images} \} \cap \{ \text{retrieved images} \}|}{|\{ \text{relevant images} \}|}, \quad (4.8)$$

For the retrieval task, the retrieved candidates are many, ranked by the relevancy between them and the given query image, so we expect the most relevant items to appear at the top of the results. However, precision and recall do not consider the ranking. Different rankings can yield the same precision and recall values as long as the number of retrieved relevant images remains constant. mean Average Precision is a metric that takes ranking into account, giving a more comprehensive understanding of the systems performance.

Considering the ranks of the retrieved results, we need to calculate a new metric called precision-at-k denoted as  $P(k)$ . By taking the first  $K$  ranked images, calculating the precision-at-k and calculating recall-at-k, denoted as  $recall(k)$  similarly, we can get a precision-recall curve based on them and obtain the comprehensive performance. And the Average Precision (AP) is the area under the precision-recall curve. It can be approximated by summing the product of  $P(k)$  and  $rel(k)$ , where  $k \in \{1, 2, \dots, K\}$  divided by the total number of relevant intervals ( $N$ ). Note that  $rel(k)$  is an indicator of whether  $k^{th}$  retrieved image index falls in the groundtruth intervals, and it equals 1 if the  $k^{th}$  retrieved image is positive and 0 otherwise. Therefore, we can define the AP as follows:

$$\text{Average Precision (AP)} := \frac{\sum_{k=1}^n P(k) \cdot rel(k)}{N}. \quad (4.9)$$

And Equation 4.10 is defined accordingly:

$$\text{mean Average Precision(mAP)} := \frac{\sum_{m=1}^M \sum_{k=1}^N P_m(k) \cdot rel_m(k)}{MN}. \quad (4.10)$$

It takes the mean over the average precisions of all the queries.

And Figure 4.4 is an example of the calculation process of AP:

In this example, the AP is

$$\frac{\sum_{k=1}^n P(k) \cdot rel(k)}{N} = \frac{1 + 0.67 + 0.5 + 0.44 + 0.5}{5} = 0.$$

Retrieved results										
rel(k)	1	0	1	0	0	1	0	0	1	1
P(k)	1	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5
Recall(k)	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.6	0.8	1
P(k) · rel(k)	1	0	0.67	0	0	0.5	0	0	0.44	0.5

Figure 4.4: Average Precision Example

Note that, for a certain query image, the number of retrieved image is usually larger than the groundtruth interval number because of inevitable redundancy in KFE module. Therefore, the goal in CBIR part is not to retrieve all frames with the query image, but to at least one frame in the positive interval. The  $K$  in Equation 4.10 is the number of positive intervals for a certain query.

In short, mAP provides a comprehensive assessment of the retrieval algorithm's performance by considering precision at various rank positions for multiple queries. It highlights methods that not only retrieve relevant results but also rank the positive results higher, offering a more comprehensive evaluation of the retrieval quality.

Overall, by utilizing these metrics, we can gain insights into the accuracy, redundancy, efficiency, and retrieval performance of the keyframe extraction and content-based image retrieval algorithms. Their application allows for a more comprehensive evaluation and comparison of different approaches, aiding in the refinement and optimization of the methods.

## 4.3 Results

### 4.3.1 Experiments on KFE

In this part, different experiments revolving around keyframe extraction module are conducted and presented. Experiment settings are shown in the setting table followed by results presented in tabular form as well.

#### 4.3.1.1 Black bar removal

Firstly, as mentioned in the problem analysis of Chapter 3 hybrid features section, it was observed that color-based features are susceptible to being influenced by the presence of black bars. Before implementing the hybrid features, we introduced the black bar removal technique into the color-based feature extraction in order to mitigate this issue.

The black bar removal pipeline is shown as Figure 4.5.

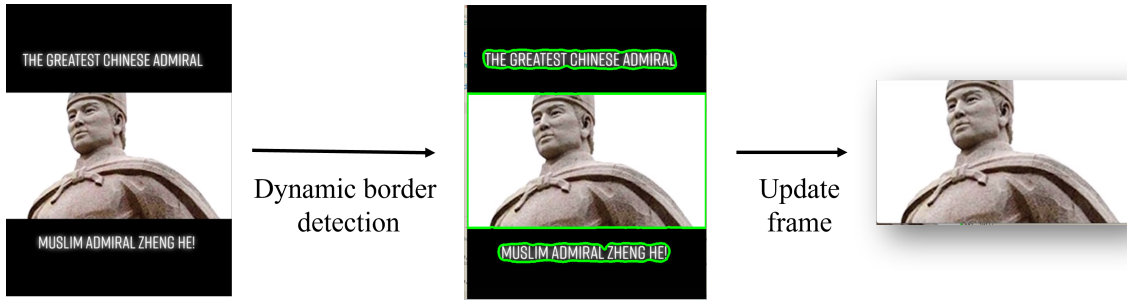


Figure 4.5: The pipeline of black bar removal technique

It conducts dynamic border detection on each frame, selecting the largest contours among the detected informative areas. By cropping around the largest detected contour, the low-level feature extraction is able to perform accurately on those regular frames. This experiment aims to assess the extent to which black bars influence the performance of low-level image representation.

The experiment sets up as Table 4.2:

Table 4.2: Black bar removal experiment setting

Experimental parameters	Setting
Black bar removal	True/False
Test set	24 videos
Feature method	RGB color histogram/CH+SMSSVD
Clustering method	Dynamic cluster
	Accuracy
Metrics	Redundancy
	mER

Table 4.3 shows the keyframe extraction results of the system with and without the black bar removal technique. In Table 4.3, the first feature tested is the original color histogram image representation, and the enhanced color-based image feature is also tested, denoted as 'CH+SMSSVD'.

Table 4.3: The impact of black bar removal on traditional methods performances

Feature type	Metrics	Without	With bar removal
Raw color histogram	Accuracy	0.809	0.964
	Redundancy	0.305	0.371
	mER	7.519	7.377
CH+SMSSVD	Accuracy	0.898	<b>0.969</b>
	Redundancy	0.296	<b>0.349</b>
	mER	20.473	<b>19.35</b>

Table 4.3 showcases the influence of a black bar removal technique on the performance of keyframe extraction. When comparing the two settings, it is evident that

employing black bar removal leads to notable improvements in various metrics.

In terms of the raw color histogram, applying black bar removal significantly enhances accuracy, increasing it from 0.809 to 0.964. This improvement suggests that the technique aids in more precise image representation. During the test, setting a higher similarity threshold can also help to improve the accuracy, because it gives a finer resolution when comparing frames, but it will cause very high redundancy. Applying black bar removal causes a reasonable increase of redundancy from 0.305 to 0.371, because when the accuracy is low, many keyframes are not detected, and those intervals' redundancy is not defined, leaving only one frame retained and zero redundancy, so the average redundancy is very low. For the CH+SMSSVD method, the implementation of image representation is different from the original CH-based features. Thus, the overall performance is slightly better than the original CH-based features. Accuracy increases from 0.898 to 0.969 and redundancy degrades to 0.349. So for the latter implementation, the impact of black bar removal is similarly positive. The mean Efficiency Ratios (mER) of them are both obtained on my local device. They both have a minimal increase because of the calculation to detect contours.

Overall, these results demonstrate that the black bar removal technique improves accuracy in both the raw color histogram and CH+SMSSVD approaches, but makes the system less efficient.

#### 4.3.1.2 Variations of ResNet

In Chapter 3, high-level features were introduced for image interpretation, with ResNet being particularly notable for its residual connections, fewer parameters, and fewer computational operations (FLOPs). Various ResNet variations were considered, including ResNet18, ResNet34, ResNet50, ResNet101, and ResNet152, each possessing distinct properties based on their structural differences. In this investigation, we aim to assess the performance of these ResNet variations in keyframe extraction (KFE) tasks and provide support for the selection of ResNet18 for hybrid features, as opposed to the other variations.

The experiment setting is shown in Table 4.4.

Table 4.4: Experiment setting for different ResNet variations

Experimental parameters	Setting
Feature methods	ResNet18/34/50/101/152
Test set	24 videos
Clustering method	Dynamic cluster
	Accuracy
Metric	Redundancy
	mER

Table 4.5 lists the feature dimensions and shows the performance of different ResNet variations in KFE module.

From the results shown in Table 4.5 or the plots in Figure 4.6, the most surprising result we get is that more layers of the network do not equal better performance.

Table 4.5: Performance comparison among different ResNet variations in KFE module

Variation	Feature Dimension	Accuracy	Redundancy	mER
ResNet18	512	0.949	<b>0.258</b>	<b>33.82</b>
ResNet34	512	<b>0.955</b>	0.267	28.18
ResNet50	2048	0.927	0.337	22.15
ResNet101	2048	0.911	0.352	17.60
ResNet152	2048	0.920	0.353	14.15

When considering deep learning algorithms for intricate image representation, strong local interest point searching ability is a desirable attribute. This ability signifies fixed attention on similar images, which is advantageous for content-based image retrieval (CBIR) tasks as they require the exclusion of disturbing objects. However, in the context of KFE, it may lead to the omission of certain frames that do not necessarily contain only one interest object.

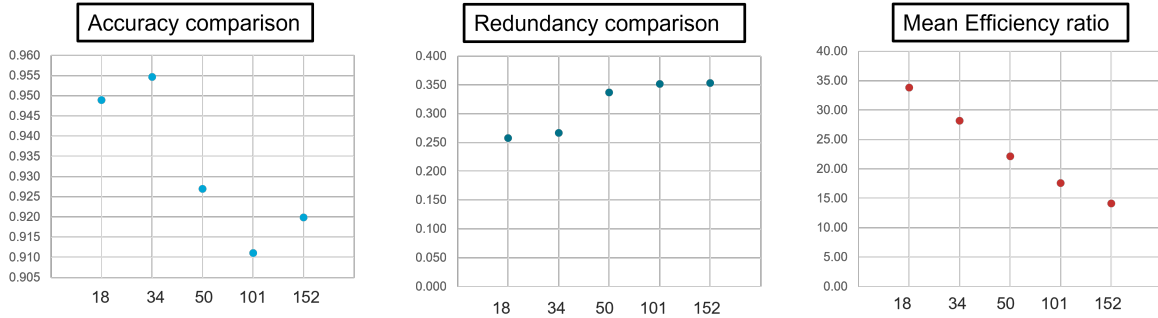


Figure 4.6: Plots of ResNet variations' Accuracy, Redundancy and mean Efficiency Ratio performances in KFE module

Examining the feature dimensions across the ResNet variations, ResNet18 and ResNet34 have a feature dimension of 512, while ResNet50, ResNet101, and ResNet152 have a higher feature dimension of 2048. Despite the higher dimensionality, ResNet18 and ResNet34 achieve comparable or better accuracy compared to the other variations. It's worth noting that ResNet34 exhibits slightly higher accuracy and redundancy compared to ResNet18 but at the cost of lower efficiency. On the other hand, ResNet50, ResNet101, and ResNet152 showcase higher redundancy and lower efficiency, resulting in a trade-off between accuracy and computational performance. With only 18 convolutional layers, ResNet18 demonstrates the lowest redundancy, highest efficiency, and second-highest accuracy among the tested variations. These results support the selection of ResNet18 for hybrid features, indicating its favorable performance in KFE tasks.

Overall, the results suggest that ResNet18's balanced performance in terms of accuracy, redundancy, and efficiency makes it a suitable choice for the hybrid features in the context of KFE. It strikes a good compromise between accurately representing frames and maintaining computational efficiency.

### 4.3.1.3 Different clustering methods

In Section 3.2, we discussed two centroid-based clustering methods that are particularly suitable for grouping similar frames and creating a keyframe dataset: K-means clustering and dynamic clustering. These methods exhibit prominent and influential differences. Specifically, K-means clustering requires a predefined number of clusters, while dynamic clustering does not. Additionally, K-means clustering overlooks the temporal characteristics of videos, whereas dynamic clustering leverages the temporal information.

To obtain quantitative results and gain insights into the performance of these methods in the task at hand, we designed an experiment as outlined in Table 4.6.

Table 4.6: Experimental setting in testing different clustering methods for KFE task

Experimental parameters	Setting
Clustering method	K-means/Dynamic
Test set	24 videos
Feature method	CH+SMSSVD
Metric	Accuracy
	Redundancy
	Average efficiency [s]

The results, ranked by redundancy, are presented in the Table 4.7.

Table 4.7: Performance comparison on two clustering methods: K-means clustering and dynamic clustering (ranked by redundancy)

K per 60 seconds	Average efficiency[s]	Accuracy	Redundancy
5	0.678	0.665	0.062
10	0.752	0.910	0.223
<b>Dynamic(0.9)</b>	<b>0.217</b>	<b>0.941</b>	<b>0.227</b>
12	0.806	0.933	0.297
<b>Dynamic(0.92)</b>	<b>0.27</b>	<b>0.969</b>	<b>0.349</b>
15	0.815	0.958	0.361
20	0.883	0.966	0.455

Starting with K-means clustering, as the number of clusters per 60 seconds increases from 5 to 20, the average efficiency grows from 0.678 seconds to 0.883 seconds, the accuracy moves from 0.665 to 0.966 accompanied by the redundancy degradation from as low as 0.062 to 0.455.

Moving on to dynamic clustering, denoted as "Dynamic" in the table, it does not require a predefined number of clusters but instead adjusts a similarity threshold to group similar frames. When the redundancy is around 0.22, its average efficiency is 0.217 seconds, and the accuracy is 0.941, both surpassing that of K-means clustering at 10 clusters. As the redundancy rises to 0.349, the accuracy of dynamic clustering has already exceeded the K-means clustering at 20 clusters per 60 seconds.

The results clearly demonstrate the superiority of dynamic clustering in this task. With both higher accuracy at a fixed level of redundancy and efficiency compared to K-means clustering, dynamic clustering proves its capability to effectively extract keyframes of the video content.

Figure 4.7 and Figure 4.8 are two illustrations of an example case demonstrating why temporal characteristic is important in this task. These two figures showcase the algorithm difference of whether including temporal adaptability.

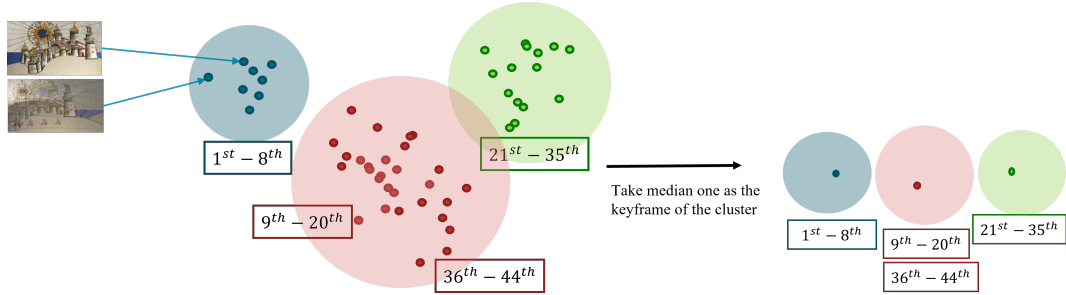


Figure 4.7: Illustration of K-means clustering

K-means clustering (Figure 4.7) treats dispersed similar frames collectively, neglecting the inherent temporal attribute of individual frames. This limitation makes it less suitable for capturing the sequential nature of videos. Moreover, K-means clustering requires a predefined number of clusters, which poses challenges when videos have varying durations and scene transition frequencies.

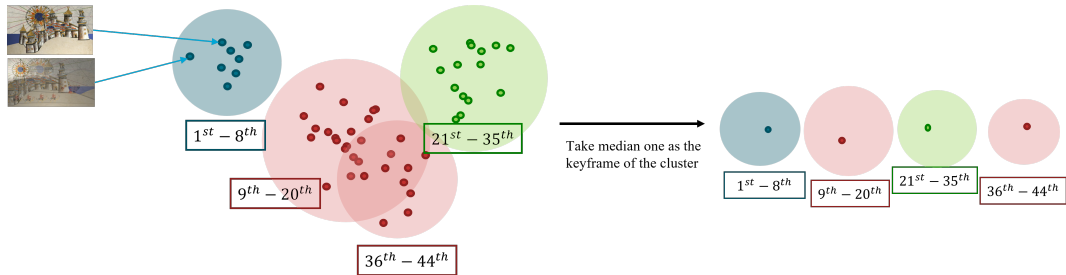


Figure 4.8: Illustration of dynamic clustering

In contrast, dynamic clustering (Figure 4.8) addresses these shortcomings by leveraging the temporal information present in the frames. It does not rely on a fixed number of clusters, allowing for a more adaptive and flexible grouping process. Instead, dynamic clustering utilizes an adjustable similarity threshold to identify and group similar frames dynamically.

Overall, these findings support the decision to favor dynamic clustering over K-means clustering for keyframe extraction. Dynamic clustering's ability to adaptively group similar frames based on capturing temporal information makes it a more suitable choice in scenarios where video durations and scene transition frequencies vary.

### 4.3.2 Image representation methods comparison on KFE

In Chapter 3, we comprehensively explored various image representation methods, including traditional color-based features, edge-based features - histogram of gradients, handcrafted features like SIFT, and deep learning-based features such as VGG16, ResNet18, and hybrid features. Given the large volume of frames involved in the keyframe extraction module, it becomes imperative to represent the images efficiently while preserving adequate information for establishing similarity among similar frames and discerning distinct content.

Before proceeding to test the promising image representation methods for KFE, we conducted a few preliminary tests, some of which were discussed in previous experiment sections. After identifying the most potential approaches among the proposed methods, we proceeded to implement and test them using our comprehensive database. In this section, we will delve into the detailed results obtained from these experiments. The experimental setup details are presented in Table 4.8.

Table 4.8: Experimental setting in testing different feature extraction methods for KFE

Experimental parameters	Setting
Feature method	Edge-based/ VSUMM/ VGG16/ ResNet18/ color-based/ Hybrid feature
Clustering method	Dynamic clustering
Test set	24 videos
	Accuracy
Metric	Redundancy
	Mean efficiency ratio

Table 4.9 shows the results of the mean efficiency ratio, accuracy, and redundancy of those tested methods. The edge-based method demonstrates a very low-efficiency ratio of 0.3. The method involves extensive computations for gradient calculations, leading to higher demands on computational resources. Consequently, it was not tested on the entire dataset for mean accuracy and redundancy due to resource constraints. The efficiency limitation is a significant drawback in practical scenarios.

VSUMM presents a noteworthy efficiency ratio of 34.22, indicating its ability to efficiently identify keyframes while maintaining a decent level of mean accuracy of 0.879, but the mean redundancy of 0.42 is too high, as a result of which it will bring much burden in the following process. And VSUMM does not meet the quantitative goal we set in Section 1.3 Problem Statement.

On the other hand, the improved color-based method exhibits the highest mean efficiency ratio, reaching an impressive value of 49.5. This highlights the method’s ability to efficiently process and extract meaningful information from the image for KFE task. It also achieves a high mean accuracy of 0.969 and a moderate level of redundancy of 0.349, making it a strong candidate for keyframe extraction tasks where efficiency and accuracy are crucial.

Note that mean efficiency ratio values marked with \* were obtained using NVIDIA RTX A6000. Values without the \* were obtained using the CPU. VGG16 and ResNet18,



Table 4.9: Performance comparison on different image representation methods in KFE task

Methods	Mean efficiency ratio	Mean accuracy	Mean redundancy
Edge-based	0.3	-	-
VSUMM	34.22	0.879	0.42
VGG16	1.96/18.06*	0.962	0.195
ResNet18	3.45/33.82*	0.949	0.258
Color-based	<b>49.5</b>	0.969	0.349
Hybrid feature	20.4*	<b>0.974</b>	<b>0.176</b>

both deep learning-based methods, showcase competitive efficiency ratios of 18.06 and 33.82 respectively. These models achieve high mean accuracy (VGG16: 0.962, ResNet18: 0.949) and moderate redundancy (VGG16: 0.195, ResNet18: 0.258), making them valuable contenders for keyframe extraction tasks where accuracy is a prioritized metric, and there are available GPU resources.

The Hybrid Feature method, combining elements from different approaches, yields a mean efficiency ratio of 20.4. The significant reduction in efficiency is attributed to its incorporation of two image interpretations. However, this approach yields remarkable mean accuracy of 0.974 and low mean redundancy of 0.176, making it a highly attractive option for tasks that require high accuracy.

In conclusion, the results and analysis reveal that each method has distinct strengths and weaknesses in terms of efficiency, accuracy, and redundancy. Researchers can consider the specific requirements of their keyframe extraction tasks to determine the most suitable method for their application, taking into account the trade-offs.

Moreover, for the CBIR (Content-Based Image Retrieval) module, we can exclude the methods that are less competitive based on our quantitative goals. This leaves us with four candidates: color-based, VGG16, ResNet18, and the hybrid feature. These selected methods offer promising potential for effective video image retrieval, aligning well with our desired objectives.

### 4.3.3 Experiments on the overall task

The subsequent phase of our investigation involves the integration of the Key Frame Extraction (KFE) module with the Content-Based Image Retrieval (CBIR) module to accomplish the overall task. As previously discussed, the CBIR module requires a more sophisticated feature of image representation to ensure retrieving accuracy. As shown in Tabel 4.10, in this experimental setup, we investigate the integrated KFE module with previously filtered-out methods and high-level features in CBIR module’s performance. The KFE module is evaluated using the hybrid feature, color-based, and ResNet18-based features, and meanwhile, dynamic clustering is employed. For the CBIR module, we explore SIFT, ResNet18, and ResNet101 feature methods. The experiment evaluation relies on the mean Efficiency Ratio and mean Average Precision metrics to gauge the balance of retrieval accuracy and computational efficiency. This comprehensive experiment aims to provide insights into the collective impact of both modules on the overall system’s effectiveness.

Table 4.10: Experimental setting in testing different feature extraction methods for the whole system

Experimental parameters	Setting
KFE feature method	Hybrid feature/ Color-based/ ResNet18
Clustering method	Dynamic clustering
CBIR feature methods	SIFT/ResNet18/ResNet101
Test set	24 videos
Metric	Mean efficiency ratio Mean average precision

Furthermore, within this section, we introduce and evaluate a recycling scheme, aiming to enhance the efficiency of the overall realization. Figure 4.9 shows the pipeline of this scheme.

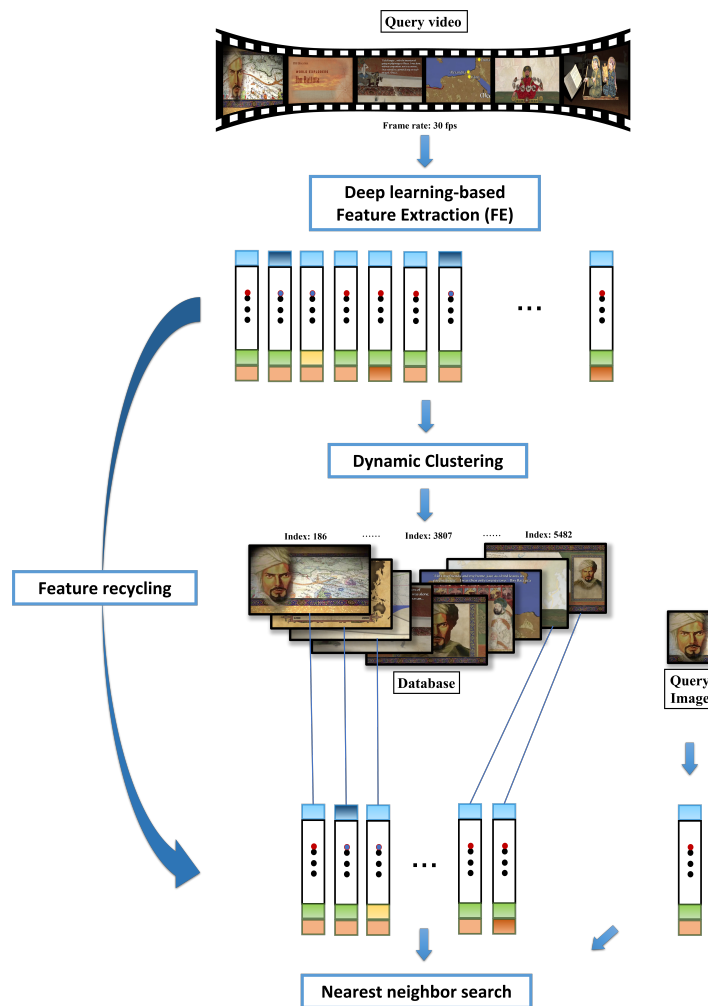


Figure 4.9: Illustration of recycling scheme

The recycling scheme optimizes the utilization of frame features, because we only extract frame features once, which is in the KFE phase. During this initial phase, frame features are used in the second phase, the chosen keyframe features are intelligently repurposed for comparison with the query image feature. This innovative strategy avoids the need for redundant feature extraction in the latter stage. In terms of feature selection, deep learning-based techniques are favored. ResNet18 emerges as the prime choice for it is the most efficient high-level feature approach in the KFE phase. Due to the high volume of frames in the first stage, using this method is the most "economical" in the recycling scheme.

Table 4.11 shows the results. The combination of Hybrid + SIFT exhibits a mean efficiency ratio of 17.91 and mAP of 0.936, indicating a balance between computational efficiency and retrieval accuracy. While the combination of Hybrid + ResNet101 demonstrates a lower mean efficiency ratio of 15.27 but a higher mAP of 0.945. This emphasizes the advantageous role of ResNet101 in boosting retrieval accuracy. But generally, the hybrid feature slightly improved retrieval accuracy at the cost of losing our primary focus: efficiency.

Table 4.11: Overall performance comparison on different combinations of feature methods

Methods	Mean efficiency ratio	mAP
Hybrid + SIFT	17.91	0.936
Hybrid + ResNet101	15.27	<b>0.945</b>
Color-based + ResNet101	<b>30.22</b>	0.902
ResNet18 + ResNet101	21.59	0.933
Recycling Scheme	<b>30.43</b>	0.781

Color-based + ResNet101 approach excels in computational efficiency, boasting an Efficiency Ratio of 30.22, yet it has low retrieval accuracy with an mAP of 0.902. The trade-off between efficiency and accuracy is evident here. Although the color-based feature is efficacious in extracting keyframes, it has relatively high redundancy. This redundancy contributes to an expanded candidate retrieval dataset, consequently impacting the effectiveness of the retrieval process. Intervening in this process. This divergence is a key factor behind the relatively diminished performance of ResNet101 in the retrieval task, in contrast to its previously observed excellence.

In terms of efficiency, the only competitive scheme for color-based + ResNet101 is the recycling scheme. Notably, the recycling scheme showcases high-efficiency gains, yielding a mean efficiency Ratio of 30.43, the highest among the tested approaches. However, the mAP sees a sharp drop to 0.781, indicating that ResNet18 is incapable of retrieving relevant candidates. After some specific validations, we found that it is highly susceptible to highly focus blurred or close-up scenes lacking discernible patterns. This demonstrates that though the recycling scheme exhibits a high potential in augmenting efficiency, it is insufficient in balancing both efficiency and accuracy in this task, using the current model.

Lastly, the ResNet18 + ResNet101 combination strikes a balance with a mean efficiency ratio of 21.59 and an mAP of 0.933. The efficiency gain due to ResNet18's

feature extraction in KFE is complemented by the accuracy boost from ResNet101 in accurate searching and matching.

#### 4.3.4 Discussion

In summary, except for the recycling scheme, the other schemes satisfy the quantitative objectives raised in Section 1.3. As can be seen, the result analysis reveals a dynamic interplay between efficiency and accuracy across the methods. The trade-offs are evident as certain combinations emphasize computational efficiency, while others prioritize retrieval accuracy. These results underscore the significance of thoughtful method selection based on the specific objectives and computational resources available. The recycling scheme's efficiency gains highlight its potential for resource optimization, but the decline in mAP suggests the need for further refinement to strike a more favorable balance between efficiency and accuracy. For tasks with volumes of videos to process, the combination of color-based and ResNet101 is optimal. Hybrid feature should be considered when the task prioritizes retrieval accuracy. Notably, the combination of two variants of ResNet offers an unparalleled equilibrium between accuracy and efficiency, catering to both metrics.

### 4.4 System Prototype

Given that this is a practical application-driven problem, we aim to complete the whole project with a functional prototype. This prototype needs to have a user-friendly interface, designed for simple operations and only uncomplicated uploads. The interface's usability should be intuitive, minimally demanding of user interaction, and seamlessly adaptable across devices with various configurations. In terms of outcome presentation, clarity is of great importance. By the final result display, users can directly obtain a complete and reliable target image temporal location.

Therefore, I also developed a simple application written in Python. This application has a simplified interface, with the initial access page illustrated as the home page in Figure 4.10.

By clicking the "Start" button, we can go to the upload page shown as Figure 4.11.

On this page, users are required to upload both a query video and a query image. This can be done simply by clicking the corresponding buttons, navigating to their desired files within their folders, and selecting the target image and video. To initiate the processing, users should click the "Run" button after successfully uploading the relevant items, as exemplified in Figure 4.11. As the processing commences, informative labels will appear to indicate the progress of both the Keyframe Extraction (KFE) and Content-Based Image Retrieval (CBIR) procedures, with each step being marked as in progress or completed. Subsequently, users will be able to access the results by clicking the "Click to See Results" button. The following figures: Figure 4.12, Figure 4.13, and Figure 4.14 are examples of this application in action across various retrieval tasks.

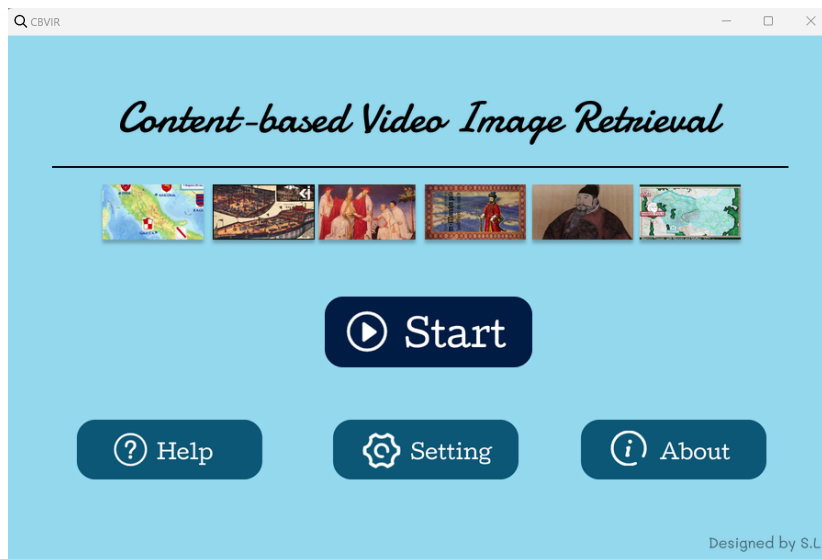


Figure 4.10: Screenshot of the graphical user interface in home page

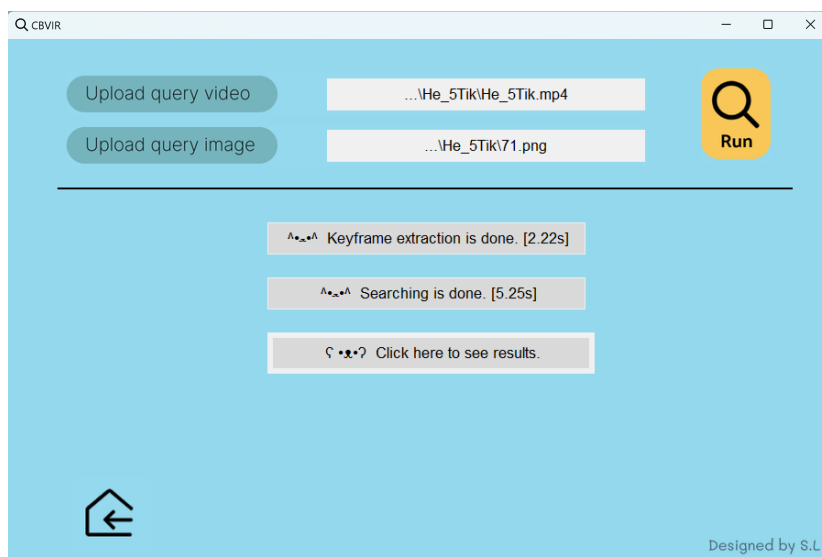


Figure 4.11: Screenshot of the start page that requests upload and processes operations from users

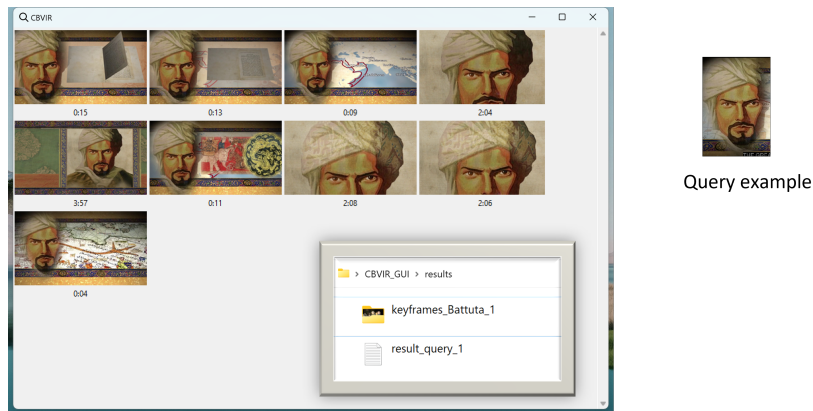


Figure 4.12: Illustration of how the results are presented in application window and saved keyframe set together with the final result list in txt file in local computer (Example of the query video of <Ibn Battuta PBS World Explorers> and the query image of index 1)

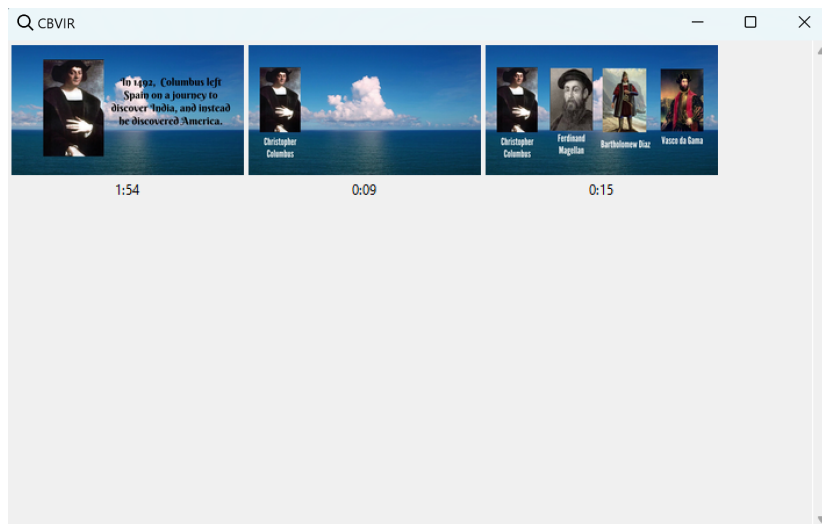
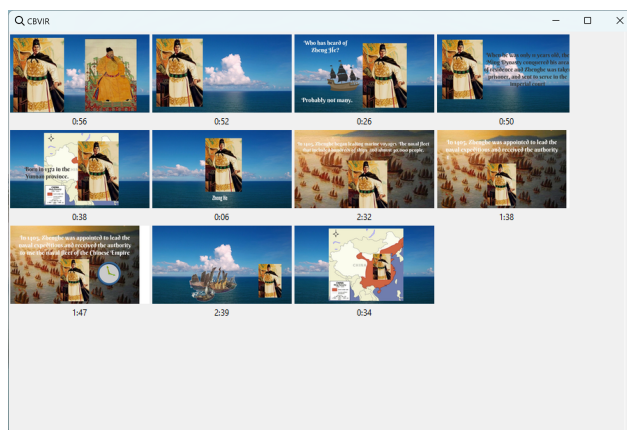


Figure 4.13: Screenshot of result page (Example of the query video of <Zhenghe facts and his accomplishments, the untold story> and the query image of index 411)



Query example

Figure 4.14: Screenshot of result page (Example of the query video of <Zhenghe facts and his accomplishments, the untold story> and the query image of index 412)





# Conclusions

---

## 5.1 Conclusion

In conclusion, this work has successfully addressed the challenges of efficient content-based video image retrieval (CBVIR) by decomposing the task and developing a modular system that strikes a delicate balance between low-level features for enhanced efficiency and high-level features for stable accuracy. Through the Key Frame Extraction (KFE) and Content-Based Image Retrieval (CBIR) modules, significant advancements have been achieved in both efficiency and accuracy, bringing us closer to a comprehensive solution for rapid and precise video content retrieval.

Starting from traditional image representation techniques, this work focused on the most promising color-based method and improved the algorithm by constructing sub-matrices within the feature matrix and reducing the dimension by SVD to accelerate the comparing process. The investigation extended to handcrafted features, proving their reliability as an alternative to high-level feature methods, particularly under constrained computing resources. Deep learning-based features, the focus of the overall realization, have shown great performance in accuracy. Building upon these foundations, a hybrid feature is proposed which reached significantly high accuracy. Moreover, integral to the overall system's success were endeavors such as integrating the black-bar removal technique and introducing dynamic clustering in drawing keyframes.

Finally, the quantitative objectives set for each module were met, demonstrating substantial efficiency improvements while maintaining a high level of accuracy. The dynamic interplay between efficiency and accuracy across various methods underscores the need for thoughtful method selection based on specific objectives and available computational resources.

## 5.2 Future directions

Looking ahead, there are several promising directions for future research in the field of content-based video image retrieval.

- First, further refinement of the recycling scheme could lead to a more optimal balance between efficiency and accuracy. The exploration could be around fine-tuning the model and proposing novel techniques for more robust image interpretation. Additionally, we still need more investigation on hybrid approaches that more dynamically combine the advantageous characteristics of different methods, catering to a wide range of applications.
- Second, the proliferation of large-scale video datasets and advances in deep learning offer opportunities for leveraging techniques such as transfer learning and

self-supervised learning to further improve both efficiency and accuracy.

- Third, as the digital landscape evolves, the demand for efficient and accurate content-based video image retrieval will only intensify. Video instance retrieval goes beyond the limitations of searching for specific instances in images, which is not sufficient for certain scenarios like selecting, highlighting, and analyzing raw video content and efficiently identifying target objects in surveillance footage. More advanced models like 3D-CNNs models designed to capture both spatial and temporal information are crucial. Or by cooperating models with the ability to interpret objects, motions, and storylines to textual information, researchers can achieve much higher real-time retrieval efficiency.

# Bibliography

---

- [1] R. Schettini, G. Ciocca, S. Zuffi *et al.*, “A survey of methods for colour image indexing and retrieval in image databases,” *Color imaging science: exploiting digital media*, pp. 183–211, 2001.
- [2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *International journal of computer vision*, vol. 60, pp. 91–110, 2004.
- [3] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions,” *Journal of big Data*, vol. 8, pp. 1–74, 2021.
- [4] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [6] S. E. F. De Avila, A. P. B. Lopes, A. da Luz Jr, and A. de Albuquerque Araújo, “Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method,” *Pattern recognition letters*, vol. 32, no. 1, pp. 56–68, 2011.
- [7] A. Nanetti, “Engineering historical memory,” [engineeringhistoricalmemory.com](https://engineeringhistoricalmemory.com/). [Online]. Available: <https://engineeringhistoricalmemory.com/>
- [8] 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.
- [9] E. Apostolidis, E. Adamantidou, A. I. Metsai, V. Mezaris, and I. Patras, “Video summarization using deep neural networks: A survey,” *Proceedings of the IEEE*, vol. 109, no. 11, pp. 1838–1863, 2021.
- [10] N. Sigger, N. Al-Jawed, and T. Nguyen, “Spatial-temporal autoencoder with attention network for video compression,” in *Image Analysis and Processing – ICIAP 2022*, S. Sclaroff, C. Distanto, M. Leo, G. M. Farinella, and F. Tombari, Eds. Cham: Springer International Publishing, 2022, pp. 290–300.
- [11] 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, 05 2018.
- [12] 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*. Berlin,

- Heidelberg: Springer-Verlag, 2022, pp. 543–548. [Online]. Available: [https://doi.org/10.1007/978-3-030-98355-0\\_52](https://doi.org/10.1007/978-3-030-98355-0_52)
- [13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” in *Proceedings of the 22nd ACM international conference on Multimedia*, 2014, pp. 675–678.
  - [14] A. Kilgarriff and C. Fellbaum, “Wordnet: An electronic lexical database,” *Language*, vol. 76, p. 706, 09 2000.
  - [15] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
  - [16] G. Toliás, R. Sivic, and H. Jégou, “Particular object retrieval with integral max-pooling of cnn activations,” *arXiv preprint arXiv:1511.05879*, 2015.
  - [17] 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.
  - [18] P. N. Chatur and R. Ranjit.M.Shende, “A simple review on content based video images retrieval,” *International journal of engineering research and technology*, vol. 2, 03 2013.
  - [19] J. Cui, F. Wen, and X. Tang, “Real time google and live image search re-ranking,” in *Proceedings of the 16th ACM international conference on Multimedia*, 2008, pp. 729–732.
  - [20] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
  - [21] A. Ravi and A. Nandakumar, “A multimodal deep learning framework for scalable content based visual media retrieval,” *arXiv preprint arXiv:2105.08665*, 2021.
  - [22] 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.
  - [23] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
  - [24] C. Kim and J.-N. Hwang, “An integrated scheme for object-based video abstraction,” in *Proceedings of the eighth ACM international conference on Multimedia*, 2000, pp. 303–311.
  - [25] H. J. Zhang, J. Wu, D. Zhong, and S. W. Smoliar, “An integrated system for content-based video retrieval and browsing,” *Pattern recognition*, vol. 30, no. 4, pp. 643–658, 1997.

- [26] A. F. Smeaton, B. Lehane, N. E. O'Connor, C. Brady, and G. Craig, "Automatically selecting shots for action movie trailers," in *Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, ser. MIR '06. New York, NY, USA: Association for Computing Machinery, 2006, pp. 231–238. [Online]. Available: <https://doi.org/10.1145/1178677.1178709>
- [27] Z. Zong and Q. Gong, "Key frame extraction based on dynamic color histogram and fast wavelet histogram," in *2017 IEEE International Conference on Information and Automation (ICIA)*, 2017, pp. 183–188.
- [28] J.-H. Huang and M. Worring, "Query-controllable video summarization," in *Proceedings of the 2020 International Conference on Multimedia Retrieval*, ser. ICMR '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 242–250. [Online]. Available: <https://doi.org/10.1145/3372278.3390695>
- [29] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying graph convolutional networks," in *International conference on machine learning*. PMLR, 2019, pp. 6861–6871.
- [30] P. Papalampidi, F. Keller, and M. Lapata, "Movie summarization via sparse graph construction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 15, 2021, pp. 13 631–13 639.
- [31] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture notes in computer science*, vol. 3951, pp. 404–417, 2006.
- [32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International Conference on Computer Vision*, 2011, pp. 2564–2571.
- [33] B. Li, D. Ming, W. Yan, X. Sun, T. Tian, and J. Tian, "Image matching based on two-column histogram hashing and improved ransac," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 8, pp. 1433–1437, 2014.
- [34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [36] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [37] J. Yue-Hei Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015, pp. 53–61.

- [38] A. Babenko and V. Lempitsky, “Aggregating local deep features for image retrieval,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1269–1277.
- [39] Y. Kalantidis, C. Mellina, and S. Osindero, “Cross-dimensional weighting for aggregated deep convolutional features,” in *Computer Vision–ECCV 2016 Workshops: Amsterdam, The Netherlands, October 8–10 and 15–16, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 685–701.
- [40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.
- [41] H. Noh, A. Araujo, J. Sim, T. Weyand, and B. Han, “Large-scale image retrieval with attentive deep local features,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3456–3465.
- [42] J. R. Smith, *Integrated spatial and feature image systems: Retrieval, analysis and compression*. Columbia University, 1997.
- [43] X.-Y. Wang, J.-F. Wu, and H.-Y. Yang, “Robust image retrieval based on color histogram of local feature regions,” *Multimedia Tools and Applications*, vol. 49, no. 2, pp. 323–345, 2010.
- [44] G. Pass and R. Zabih, “Histogram refinement for content-based image retrieval,” in *Proceedings Third IEEE Workshop on Applications of Computer Vision. WACV’96*. IEEE, 1996, pp. 96–102.
- [45] J. R. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics*. John Wiley & Sons, 2019.
- [46] Y. Gong and X. Liu, “Video summarization and retrieval using singular value decomposition,” *Multimedia Systems*, vol. 9, no. 2, pp. 157–168, 2003.
- [47] H. Abdi and L. J. Williams, “Principal component analysis,” *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.
- [48] G. Strang, *Introduction to linear algebra*. SIAM, 2022.
- [49] V. Klema and A. Laub, “The singular value decomposition: Its computation and some applications,” *IEEE Transactions on Automatic Control*, vol. 25, no. 2, pp. 164–176, 1980.
- [50] V. Vasudevan and M. Ramakrishna, “A hierarchical singular value decomposition algorithm for low rank matrices,” *arXiv preprint arXiv:1710.02812*, 2017.
- [51] R. Henningsson and M. Fontes, “Smssvd: submatrix selection singular value decomposition,” *Bioinformatics*, vol. 35, no. 3, pp. 478–486, 2019.

- [52] K. Mizuno, Y. Terachi, K. Takagi, S. Izumi, H. Kawaguchi, and M. Yoshimoto, “Architectural study of hog feature extraction processor for real-time object detection,” in *2012 IEEE Workshop on Signal Processing Systems*, 2012, pp. 197–202.
- [53] H. S. Dadi and G. M. Pillutla, “Improved face recognition rate using hog features and svm classifier,” *IOSR Journal of Electronics and Communication Engineering*, vol. 11, no. 4, pp. 34–44, 2016.
- [54] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR’05)*, vol. 1. Ieee, 2005, pp. 886–893.
- [55] C. Q. Lai and S. S. Teoh, “A review on pedestrian detection techniques based on histogram of oriented gradient feature,” in *2014 IEEE Student Conference on Research and Development*, 2014, pp. 1–6.
- [56] D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superpoint: Self-supervised interest point detection and description,” *CoRR*, vol. abs/1712.07629, 2017. [Online]. Available: <http://arxiv.org/abs/1712.07629>
- [57] R. Hu and J. Collomosse, “A performance evaluation of gradient field hog descriptor for sketch based image retrieval,” *Computer Vision and Image Understanding*, vol. 117, no. 7, pp. 790–806, 2013.
- [58] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [59] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [60] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang *et al.*, “Large scale distributed deep networks,” *Advances in neural information processing systems*, vol. 25, 2012.
- [61] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [62] J. Fajtl, H. S. Sokeh, V. Argyriou, D. Monekosso, and P. Remagnino, “Summarizing videos with attention,” in *Computer Vision–ACCV 2018 Workshops: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers 14*. Springer, 2019, pp. 39–54.
- [63] T.-J. Fu, S.-H. Tai, and H.-T. Chen, “Attentive and adversarial learning for video summarization,” in *2019 IEEE Winter Conference on applications of computer vision (WACV)*. IEEE, 2019, pp. 1579–1587.

- [64] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.
- [65] X. Jin and J. Han, *K-Means Clustering*. Boston, MA: Springer US, 2010, pp. 563–564. [Online]. Available: [https://doi.org/10.1007/978-0-387-30164-8\\_425](https://doi.org/10.1007/978-0-387-30164-8_425)
- [66] M. G. Omran, A. Salman, and A. P. Engelbrecht, “Dynamic clustering using particle swarm optimization with application in image segmentation,” *Pattern Analysis and Applications*, vol. 8, pp. 332–344, 2006.